Propuesta de una arquitectura multiagente para la gestión de tráfico

M. Robledo, R. Fuentetaja, A. García Serrano, J. Z. Hernández

Departamento de Inteligencia Artificial
Universidad Politécnica de Madrid
Campus de Montegancedo s/n
28660 Boadilla del Monte (Madrid), España
{phernan, mrobledo, rfuente, agarcia}@isys.dia.fi.upm.es

Resumen. Se presenta el diseño de una arquitectura multiagente para la gestión de tráfico rodado. Tras una discusión sobre las arquitecturas de agentes en la bibliografía se realiza una propuesta global para un sistema multiagente en un entorno real. Finalmente se incluyen algunas conclusiones respecto al diseño del sistema y las líneas futuras a tener en cuenta.

Palabras clave: Sistemas Multiagente, Gestión de trafico, Ingeniería del Conocimiento

Tópicos del congreso: Sistemas Multiagente e IA Distribuida
Sistemas de Ayuda a la Decisión

Persona de contacto: Mónica Robledo
Departamento de Inteligencia Artificial
Universidad Politécnica de Madrid
Campus de Montegancedo s/n
28660 Boadilla del Monte (Madrid), España
email: mrobledo@isys.dia.fi.upm.es
Tlf.; 91 336 7390
Fax.: 91 352 4819

Sección: Open Discussion Track

Propuesta de una arquitectura multiagente para la gestión de tráfico [1]

M. Robledo, R. Fuentetaja, A. García Serrano, J. Z. Hernández

Departamento de Inteligencia Artificial
Universidad Politécnica de Madrid
Campus de Montegancedo s/n
28660 Boadilla del Monte (Madrid), España
{phernan, mrobledo, rfuente, agarcia}@isys.dia.fi.upm.es

Resumen. Se presenta el diseño de una arquitectura multiagente para la gestión de tráfico rodado. Tras una discusión sobre las arquitecturas de agentes en la bibliografía se realiza una propuesta global para un sistema multiagente en un entorno real. Finalmente se incluyen algunas conclusiones respecto al diseño del sistema y las líneas futuras a tener en cuenta.

1 Introducción

El auge de los agentes software en el desarrollo de aplicaciones en los últimos años se debe en gran medida a la necesidad de afrontar nuevos desarrollos para problemas complejos en entornos distribuidos, como en el caso de la gestión de tráfico rodado [13]. Esta tecnología es especialmente potente cuando en un mismo entorno conviven distintos agentes software que deben coordinar sus actividades.

La dificultad que conlleva definir la coordinación se observa al menos por los múltiples intentos de definición en las diferentes disciplinas que lo abordan e incluso por el hecho de que en diferentes estudios se confunde el término con el de cooperación (cuando hay un objetivo común) o colaboración (para connotar grupos con un objetivo compartido de carácter muy genérico) e incluso a veces con competición (cuando de la coordinación se llega a componentes perdedores y ganadores). Además en sociología es bien conocida la problemática sobre el nexo micro-macro que expresa la relación entre los componentes del grupo de agentes (los sistemas micro) y su estructura social (o sistema macro). En esta disciplina se plantea que la resolución de la problemática referida al nexo macro-micro permite contestar a la pregunta: ¿Cómo y por qué puede un sistema macro regular a sus micro sistemas autónomos y movidos por sus intereses propios de forma que el conjunto exhiba un

[1] Este trabajo está financiado por los proyectos DAMMAD (Diseño y Aplicación de Modelos Multiagente para Ayuda a la Decisión), TIC-2000-1370-C04 y Agentcities.NET, IST-2000-28-384 Traffic Agentcity for knowledge-based Recommendation (TRACK-R).

comportamiento acorde con el servicio que ofrecen individualmente y los que reclaman?

En el dominio de la gestión de tráfico, es necesario plantear el nexo micro-macro debido a que en ocasiones es imposible prefijar todas las alternativas generadas por las necesidades de los diferentes agentes que demandan el servicio de los restantes.

Una solución es la utilización de formulaciones basadas en el conocimiento que permitirán incorporar las reglas del juego para las tareas realizadas cooperativamente en un sistema de múltiples agentes, reglas que pueden variar en el tiempo de acuerdo con las circunstancias cambiantes. En este tipo de soluciones es posible que: los protocolos de cooperación se basen en la composición adecuada y no determinista de tareas típicas durante la actuación del grupo de agentes y en la introducción progresiva en el sistema de diferentes tipos y niveles de servicios. Este es el planteamiento que dirige el trabajo cuya primera etapa de revisión bibliográfica y propuesta de diseño se presenta en este artículo.

En la sección 2 se describen diferentes arquitecturas de agente y se plantea la propuesta que se incluye en la siguiente sección y que se desarrollará en el marco de los proyectos DAMMAD y Agentcities.NET. Finalmente, en la sección 4, se presentan algunas conclusiones y las líneas futuras a tener en cuenta.

2 Arquitectura de agentes

Esta sección incluye una discusión sobre las diferentes arquitecturas de agente que se han planteado hasta el momento, sin ánimo de ser exhaustivos.

La definición de agente ha resultado ser tan controvertida como la definición de Inteligencia Artificial ya que en la literatura se han propuesto muchas definiciones para este concepto sin que ninguna de ellas haya sido aceptada plenamente por la comunidad científica. Quizás por este motivo se ha optado por la definición mediante un conjunto de propiedades que caracterizan a los agentes, aunque un agente no tenga por qué poseer todas ellas. Así, seguramente la definición más citada sea la propuesta por Wooldridge y Jennings [9], según la cual *un agente es un sistema informático que está situado en un entorno y es capaz de actuar de forma autónoma y flexible.*

El hecho de que un agente sea una entidad situada en un entorno implica que recibe información de entrada de sensores situados en el entorno y a partir de esta información realiza acciones que de alguna forma implican cambios en el entorno. La autonomía se refiere a que el agente debe ser capaz de actuar de una forma racional sin intervención de humanos o de otros agentes. Que un agente sea capaz de actuar de forma flexible para conseguir sus objetivos implica que el agente debe ser:

- Reactivo: capaz de responder en un tiempo adecuado a cambios en el entorno en el que se encuentra situado.

- Proactivo: capaz de exhibir un comportamiento oportunístico, dirigido a obtener sus metas y tomar la iniciativa cuando sea apropiado.
- Social: capaz de interactuar con otros agentes (humanos o no) a través de un lenguaje de comunicación entre agentes.

Otras propiedades que se pueden atribuir en mayor o menor medida a los agentes son:

- Continuidad temporal, se considera un agente como un proceso sin fin, ejecutándose continuamente y desarrollando su función.
- Adaptabilidad, el agente aprende y puede cambiar su comportamiento basándose en este aprendizaje.
- Movilidad, capacidad de un agente de transladarse a través de una red telemática.
- Veracidad, asunción de que un agente no comunica información falsa a propósito.
- Benevolencia, asunción de que un agente está dispuesto a ayudar a otros agentes si esto no entra en conflicto con sus propios objetivos.

No existe un consenso sobre el grado de importancia de cada una de estas propiedades para un agente, pero de hecho son muy utilizadas en la bibliografía.

Los agentes conviven con otros agentes y esto lleva al planteamiento de dos nuevos paradigmas de diseño: el de Sistema Basado en Agentes y el de Sistema Multiagente. Entendemos por un S*istema Basado en Agentes* es un sistema que contiene uno o varios agentes. En principio el término *basado en agentes* significa que el sistema puede ser descrito conceptualmente en términos de agentes, aunque se implemente sin estructuras software que correspondan a agentes. Cuando el sistema es diseñado e implementado mediante varios agentes (heterogéneos o no) que interactúan mediante una asignación de tareas cooperativa, una negociación o por coordinación emergente, por ejemplo, entonces se denomina S*istema Multiagente*.

Los Sistemas Multiagente y los Sistemas Basados en Agentes son especialmente adecuados para problemas complejos en los que son necesarios múltiples métodos de resolución de problemas y/o existen muchas entidades para la resolución de problemas. No solo se facilita con este diseño la reutilización de software y su mantenimiento, además comparte las ventajas de los sistemas concurrentes y permiten utilizar patrones de interacción complejos.

Respecto a la arquitectura de los agentes, esta a su vez estará descrita por diferentes módulos y por la forma de interactuar para que el agente pueda exhibir una determinada conducta. Frente a otros paradigmas para desarrollo de software cuyos componentes son fijos como la de objetos (objetos, atributos y métodos) o la de sistemas basados en el conocimiento (motor de inferencia, base de hechos, base de conocimiento y otros elementos opcionales), en los agentes es posible encontrar una grado elevado de no determinismo y flexibilidad. Desde un punto de vista general un agente puede contar con los siguientes módulos:

- *Módulos de interacción*: Proporcionan una interfaz de percepción y actuación con el entorno[2], que puede estar formados por usuarios humanos o cualquier otra fuente de información. Normalmente, para cada uno de los componentes del entorno, el agente tiene un módulo de interacción especializado.
- *Bases de conocimiento*: en la que el agente mantiene una *representación interna de su entorno*, en caso de que ésta sea necesaria.
- *Módulo/s de integración de información*: que deberán estar conectados con los módulos de interacción y cuya función principal es integrar de forma consistente la información que reciben en la *base de conocimiento*.
- *Módulo/s de interpretación de información*: cuya función es interpretar la nueva información, para que el módulo de planificación determine las acciones a realizar.
- *Módulo/s de planificación*: cuya función es construir un plan, es decir, una secuencia de acciones a realizar para conseguir los objetivos del agente.
- *Módulo/s de acción*: que deberán estar conectados con los módulos de planificación y cuya función es ejecutar y monitorizar las acciones pertinentes.
- *Módulo/s de coordinación*: que se encargan de la comunicación y cooperación con otros agentes.

Las arquitecturas de agente se pueden clasificar atendiendo a diferentes criterios. En concreto, según el comportamiento que exhiben los agentes se pueden clasificar en deliberativas, reactivas e híbridas:

- Los agentes con una *arquitectura deliberativa* se caracterizan porque contienen explícitamente un *modelo simbólico del entorno*, y las decisiones (por ejemplo sobre que acciones realizar) son tomadas vía razonamiento lógico (o al menos pseudo-lógico) basado en equiparación de patrones y manipulación simbólica [3].

- Los *agentes reactivos* se caracterizan porque pueden operar rápida y efectivamente sin la necesidad de procesar una representación simbólica del entorno, ya que que no poseen modelos simbólicos externos del entorno en el que se encuentran [6]. Los agentes reactivos toman decisiones basadas totalmente en el presente, sin hacer uso de lo que ha ocurrido en el pasado, porque no conocen su historia [15]. Este tipo de agentes actúan siguiendo un esquema *estímulo-respuesta* según el estado actual del entorno en el que están embebidos. La característica más importante de los agentes reactivos es el hecho de que los agentes son relativamente simples e interaccionan con otros agentes de manera sencilla. Sin embargo, si se observa de una forma global el conjunto de agentes, las interacciones pueden dan lugar a patrones de comportamiento muy complejos.

- Las *arquitecturas híbridas* combinan componentes de tipo reactivo con componentes de tipo deliberativo. La parte reactiva interacciona con el entorno y reacciona rápidamente a los eventos que en él se producen sin invertir tiempo en realizar razonamiento, mientras que la parte deliberativa planifica y se encarga de la parte de toma de decisiones, es decir, realiza tareas a un nivel de abstracción superior [3].

La selección del tipo de arquitectura de agente debe realizarse de acuerdo con los requisitos de la aplicación a desarrollar. En el caso de la gestión de tráfico se identifican como necesarias tanto las arquitecturas deliberativas como las reactivas. El uso de arquitecturas deliberativas es para que la gestión de trafico sea un proceso de inferencia de las Bases de Conocimiento con la información recibida en tiempo real. Sin embargo la adquisición de datos del entorno, en general, exige una transformación cuantitativa o cualitativa de carácter simple, por ello una arquitectura reactiva para este tipo de agentes es suficiente.

3 Propuesta de una arquitectura multiagente aplicada a un problema de gestión de tráfico

Las características de los problemas para las cuales los sistemas multiagente son una solución acertada aparecen en el dominio de la gestión de tráfico: (1) los problemas de gestión de tráfico son geográficamente y funcionalmente distribuidos, (2) los subsistemas suelen tener un alto grado de autonomía, (3) la reutilización y mantenimiento de los subsistemas locales es una tarea crítica y (4) normalmente las aplicaciones de gestión de tráfico han de ser dinámicas.

A partir de la experiencia previa en el dominio de tráfico [4,5,8,11,12] se están realizando dos proyectos complementarios que tienen como principal objetivo el uso de tecnología multiagente para el desarrollo de problemas en el dominio de tráfico. La aplicación comenzará a desarrollarse para la ciudad de Bilbao y tiene como objetivo ofrecer los siguientes servicios:

- Ayudar a un agente (humano o software) cuya misión es supervisar una carretera. Para realizar esta tarea se debe conocer en tiempo real el estado de la carretera y cómo resolver los problemas que en ella se produzcan. Por lo tanto, el servicio que proporcionará el sistema deberá incluir por una parte la presentación de la información relativa a los problemas que se están produciendo en la carretera y por otra parte la recomendación de las acciones de control a llevar a cabo para resolverlos. El desarrollo de este primer servicio se lleva a cabo en el proyecto DAMMAD.

- Asistir a otro agente (humano o software) que quiere conocer las distintas rutas alternativas para realizar un viaje. Para ello, el sistema proporcionara los caminos alternativos, ordenados según el estado de las carreteras. El desarrollo de este servicio se lleva a cabo dentro del proyecto europeo Agentcities.NET.

Dado que estos servicios presentan puntos comunes (por ejemplo para poder llevar a cabo el servicio, en ambos casos es necesario conocer el estado de la carretera) y dado que las características de ambos encajan en la filosofía de los sistemas multiagente, se plantea el diseño de una única arquitectura multiagente que de soporte a los dos servicios.

Para diseñar una arquitectura multiagente, en una primera etapa hay que determinar los distintos tipos de agentes que la integran. En una primera aproximación y en función de las necesidades de los dos servicios antes mencionados se pueden distinguir cuatro tipos de agentes:

- *Agentes de Carretera*, que según la distribución geográfica de la carretera se encargarán de monitorizar el estado de un determinado área de la misma, y por lo tanto serán responsables de la detección de los problemas que se produzcan en ese área. El número de agentes de carretera dependerá, por consiguiente, del número de áreas en que ésta se descomponga. Estos agentes están implicados en los dos servicios ya que es imprescindible en ambos casos conocer el estado de la carretera en cada momento.

- *Agentes Problema*, que son capaces de detectado un problema, recomendar las acciones de control a llevar a cabo para poder solucionarlo, utilizando la información que reciban de los agentes de carretera y coordinándose con los distintos Agentes Problemas si tienen que resolver problemas cuyas soluciones pueden estar relacionadas. Cada agente problema estará especializado en un tipo concreto de problema, por lo tanto existirán tantos como tipos de problema sean identificados. Estos agentes estarán implicados únicamente en el servicio que da soporte a la supervisión de la carretera.

- *Agentes de Gestión de Tráfico*, que en tiempo real deberán informar al resto de agentes (humanos o software) encargados de llevar a cabo las acciones de control, sobre la existencia de problemas y las acciones a llevar a cabo. Para ello tendrán que coordinarse con los distintos Agentes de gestión de Trafico si existen problemas comunes a ambos para llegar a una solución conjunta. Existirá un agente de este tipo por cada centro de control. Los agentes de gestión de tráfico, al igual que los de problema están implicados en el servicio que da soporte a la supervisión de la carretera.

- *Agentes de recomendación de rutas alternativas (TRACK-R)*, que serán capaces de recomendar las distintas rutas alternativas para viajar de un lugar a otro, en base del estado de las carreteras que unen ambos lugares. Las rutas deberán ofrecerse ordenadas dependiendo del estado de las carreteras, luego estos agentes deberán comunicarse con los agentes de carretera para obtener la información necesaria. Cada agente de recomendación será responsable de un área geográfica, por lo que sí un único viaje implica atravesar varias áreas deberá comunicarse y coordinarse con los agentes de recomendación de rutas encargados de cada una de ellas. Esto agentes únicamente están implicados en el servicio de recomendación de rutas.

Una vez identificados los distintos tipos de agentes, pasamos a establecer cómo estos agentes están organizados, planteando una visión de lo que sería la arquitectura multiagente del sistema, para después identificar con qué módulos debe contar la arquitectura de cada agente.

La arquitectura multiagente propuesta para el desarrollo de ambos servicios se puede observar en la Figura 1

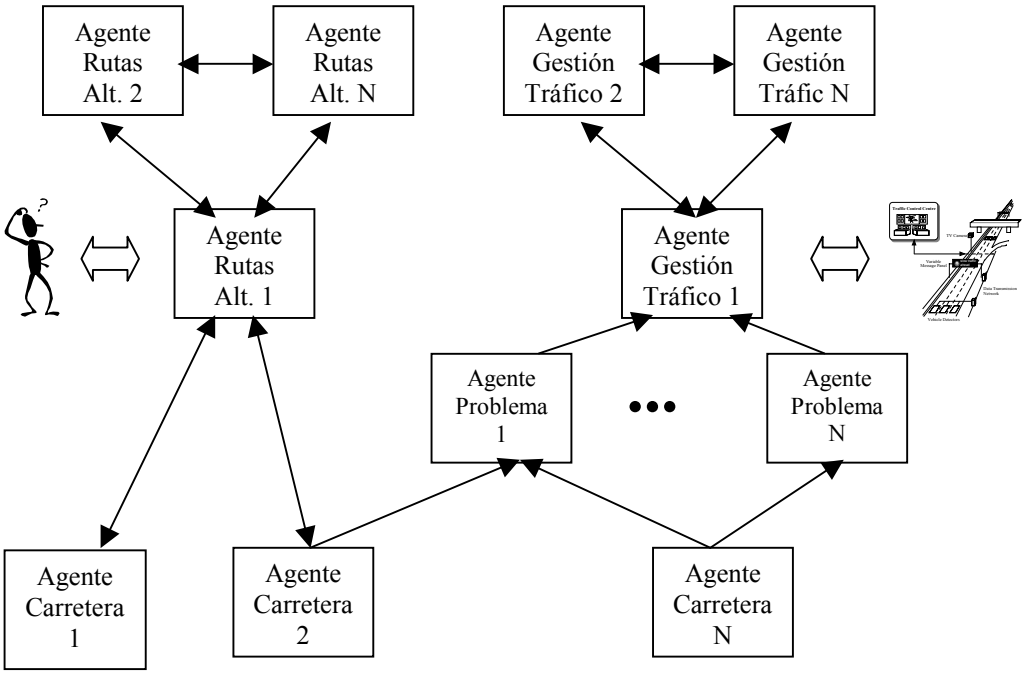

Fig. 1. Propuesta de arquitectura multiagente

A la vista de la arquitectura se pueden distinguir dos modelos de interacción:

- *Modelo de ejecución bajo demanda*, es el modelo que sigue la interacción entre los agentes de carretera y los agentes TRACK-R. En caso de que sólo esté activo el servicio de recomendación de rutas, no es necesario que los agentes de carretera estén monitorizando continuamente el estado de la misma, sólo deberán hacerlo bajo demanda en caso de que el agente TRACK-R lo solicite. Si por el contrario los dos servicios están activos, los agentes de carretera estarán continuamente monitorizando el estado de la misma, pero la comunicación del estado de la carretera al agente TRACK-R sólo se realizará bajo demanda de éste.

- *Modelo de ejecución continuo*, es el modelo que sigue la interacción entre los agentes de carretera, los agentes problema y los agentes de gestión de tráfico. Continuamente los agentes de más bajo nivel informarán a los agentes que están en un nivel superior: los agentes de carretera continuamente informarán a los agentes problema del estado de la carretera y estos, a su vez, continuamente informarán a los agentes de gestión de tráfico sobre cómo solucionar los problemas que se produzcan.

A continuación se identifican los módulos con los que debe contar la arquitectura de cada tipo de agente para que sea capaz de exhibir el comportamiento antes mencionado:

- *Agentes de Carretera*:
  - Módulos de Interacción: Para comunicarse con los Agentes Problema y los Agentes de rutas alternativas.
  - Bases de Conocimiento: Estos agentes tendrán la información topologica de la zona de la carretera de la cuál son responsables y el conocimiento de cómo detectar problemas.

- *Agentes Problema*:
  - Módulos de Interacción: Para comunicarse con los Agentes de gestión de tráfico, los Agentes de Carretera y con otros Agentes Problema.
  - Bases de Conocimiento: Cada tipo de agente tendrá que tener el conocimiento de cómo resolver el tipo de problema del cual se encargan.
  - Módulos de Planificación y Acción: Tienen las distintas estrategias de inferencia para poder resolver los problemas. Será el encargado de construir el plan para solucionar el problema.
  - Módulo de Coordinación: Estos agentes tendrán que coordinarse con los distintos Agentes si es que para resolver un problema en su zona necesitan conocer información relativa a zonas contiguas o cuando es necesario que distintos Agentes Problema se coordinen para llegar a una solución común.

- *Agentes de Gestión de Tráfico*:
  - Módulos de Interacción: Para comunicarse con los distintos *Agentes Problemas,* con los agentes (humanos o software) encargados de supervisar la carretera y con otros Agentes de Gestión de Trafico.
  - Módulo de Coordinación: Este módulo es necesario ya que distintos Agentes encargados de controlar distintas carreteras o distintas zonas de estas, han de coordinarse para llegar a una solución común cuando existen problemas que afectan a la vez a las zonas controlados por dichos agentes.

- *Agentes de recomendación de rutas alternativas (TRACK-R)*:
  - Módulos de Interacción: Para comunicarse con los Agentes de carretera y con otros Agentes TRACK-R si es necesario.
  - Bases de Conocimiento: Tendrá que tener la información topologica de la red de carreteras.
  - Módulos de Planificación y Acción: Tendrá que tener los métodos de planificación necesarios para poder clasificar las rutas alternativas según el estado de las distintas zonas de la carretera.
  - Módulos de Coordinación: El agente TRACK-R tendrá que coordinarse con distintos agentes TRACK-R situados en distintos centros de control cuando la ruta de viaje propuesta involucre el paso por distintas ciudades controladas por distintos agentes.

4 Conclusiones y trabajo futuro

En este trabajo, se ha incluido el diseño de los agentes que van formar parte del sistema (Agentes de Gestión de Trafico, Agentes de Problemas, Agentes de carretera y Agentes de rutas alternativas) y la arquitectura multiagente en que se integran.

Tras esta etapa de diseño, el siguiente paso a realizar será permitir que los distintos agentes puedan comunicarse, para lo que habrá que definir una ontología [7] de trafico que permitirá la interoperabilidad semántica entre (1) los distintos agentes desarrollados para un mismo centro de control, (2) entre distintos agentes TRACK-R cuando la recomendación de una ruta alternativa involucre el paso por distintas ciudades controladas por diferentes centros de control y (3) el Agente TRACK-R y el Agente de gestión de tráfico con distintos agentes externos a nuestro sistema.

Otra de las tareas a realizar será el diseño (o reutilización) de técnicas de coordinación [14] para que distintos agentes puedan cooperar para poder alcanzar sus metas.

Referencias

1. Boero M., Kirschchfink H. And Barcelo J. Real-Time Traffic Management Supporting Intermobility and Strategic Control. Proc. 4th World Congr. On Intelligent Transport Systems (no. 2351), Berlin, Germany (1997)
2. Brenner, W., Zarnekov, R., Wittig, H. Intelligent Software Agents. Foundations and applications. New York. Springer-Verlag Berlin Heidelberg. (1998).
3. Burmeister, B., Sundermeyer, K. Cooperative problem solving guided by intentions and perception. In Werner, E. and Demazeau, Y., editors, Decentralized AI 3 (MAAMAW-91) pages 77-92. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands. (1992).
4. Cuena J., Hernández J.Z. and Molina M: Knowledge Oriented Design of an Application for Real Time Traffic Management: The TRYS system. Eur. Conf. on Artificial Intelligence, ECAI'96, W. Wahlster (ed.), John Wiley and Sons Ltd. (1996a)
5. Cuena J., Hernández J.Z. and Molina M: An Intelligent Model of Road Traffic Management in the Motorway Network Around Barcelona. Proc. 14th Int. Federation on Information Processing (IFIP) World Computer Congr., (1996) 173-180, Chapman & Hall.
6. Garcia-Serrano, A., Ossowski, S. Inteligencia Artificial Distribuida y Sistemas Multiagentes. Revista Iberoamericana de Inteligencia Artificial**.** Numero 6, Volumen 2, Otoño (1998)
7. Gómez-Perez A., Benjamins R. Overview of Knowledge sharing and reuse components: Ontologies and problem Solving Methods. Proc. of the IJCAI'99 workshop on Ontologies and Problem-Solving Methods (KRR5) Stockholm (1999)
8. Hernández, J., M.; Ossowski, S.; García-Serrano, A: Evaluating Multiagent Co-ordination Architectures - A Traffic Management Case Study In: 34th Hawaii International Conf. on System Sciences, Hawaii (HICCS), EEUU. (2000)
9. Jennings, N.R., Sycara, S., and Wooldridge, M. A Roadmap of Agent Research and Development. Autonomous Agents and Multi-Agent Systems I. Kluwer, (1998) 275-306
10. Kirschfink H. Collective Traffic Control in Motorways. 11th EURO-Mini Conference on Artificial Intelligence in Transportation System and Science and 7th EURO Working

Group Meeting on Transportation. Tutorial on Intelligent Traffic Management Models. Helsinki (1999)
11. Molina M., Robledo M., Fernández A.: A propose-and-revise System for Real-time Traffic Management. Proc. European Symposium on Intelligent Techniques ESIT'00 (2000) Aachen, Germany.
12. Molina M., Robledo M.: A knowledge model for automatic configuration of traffic messages The 14th International Conference on Industrial and Engineering Application of AI and Expert Systems. LNAI 2070, Springer-Verlag (2001).
13. Ossowski S., Cuena J. And García-Serrano A. : A case of Multiagent Decision Support: Using Autonomous Agents for Urban Traffic Control. Proc. Iberoamerican Conf. of Artificial Intelligence, IBERAMIA (1998).
14. Ossowski S. Co-ordination in Artificial Agent Societies. LNAI 1535, Springer-Verlag (1999).
15. Weiss, G. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence . London. (1999).