Multi-Agent Based Control for Urban Traffic Networks

Joel A. Trejo Sánchez and Julio C. Ramírez Pacheco

Universidad del Caribe, SM 78, Manzana 1, Lote 1, 77528, Cancún QRoo, México
{jtrejo,jramirez}@ucaribe.edu.mx

Abstract. This paper deals with the traffic management within an urban traffic network (UTN). The paper shows the necessity of an adaptive control system that detects and takes decisions when traffic congestion occurs. A multi-agent control scheme is proposed in which a control agent is charged to manage the traffic lights of an intersection; the agent communicates with other agents of the neighbor intersections. The architecture of such a control agent is described as well as relevant issues of a prototype implementation. The simulation test of an adaptive control policy is presented.

Keywords: Urban traffic control; Agent based approach.

1 Introduction

Urban traffic congestion causes considerable cost due to time losses, has a negative impact on the environment and increases the probability of accidents [1]. Coupled with an expanding population, urban traffic networks are unable to efficiently handle the daily movements of traffic through urban areas. Adjusting the traffic lights according with the traffic conditions is a costly solution if it involves human interaction. Traffic light management is a complex problem and there exist several intelligent algorithms that have been used to solve it.

The main urban traffic control objectives are to maximize the performance of the system and to reduce the total time of accumulated delay through real-time decision making processes. A way to reach these objectives is the updating of traffic lights timing according to the demand of traffic and an appropriate coordination strategies among the local traffic controllers within a specific area.

In this paper a multi agent based approach for addressing the problem of traffic control is presented. An agent is charged to control the operation of a traffic light in an intersection; it determines the timing and policies according to the vehicle flow in the streets and the current functioning of other traffic light controller agents in the traffic network.

This paper is organized as follows. Section 2 overviews the background of the presented work. Section 3 describes the control system integration with an urban traffic simulator. Section 4 presents the architecture for traffic control. Some experimental results using micro simulation are presented in section 5.

© S. Torres, I. López, H. Calvo. (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 27, 2007, pp. 191-201

Received 23/02/07
Accepted 08/04/07
Final version 20/04/07

2 Urban Traffic Control

Adaptive control systems must have the capability to adjust the traffic signals based on current traffic. There exist several approaches to implement an adaptive traffic control system; some of them are based on learning refoircement, fuzzy logic, multi-agent systems(MAS), discrete events systems, and others approaches. The proposed model is based on multi-agent systems because of the agent's capacities including autonomy, interaction, and modularity of the software.

In [2] a Petri net model is proposed to control the traffic lights. The model is divided in two categories: 1) to determine the signal sequence to follow in the control policy and 2) how to implement the logic to manage the signals. Models based on hybrid Petri nets are proposed in [3] and in [4]. Traffic flow and density are represented with continuous Petri nets, and traffic lights are modeled as discrete event models.

Three types of agents for the traffic management are proposed by [5]. The intersection agents decide the pertinent control strategy of the intersections based on their objectives, capacities, perception, and data. In [6] a multi-agent methodology is developed focusing on learning reinforcement to distribute the traffic control. With the aim of coordinate the intersections, in [7] is proposed the use of mobile agents to set coordination with the rest of the intersections.

3 An Urban Traffic Simulator

In order to prove our methodology, a way to measure our results is needed. Nowadays analysis and optimization of traffic systems are preferred as an alternative to develop a traffic control system; analytical and simulation methods have been proposed for the study of such systems since it permits to predict the control system behavior in the real world.

There exist two main simulation approaches: event oriented and time (slicing) oriented. Our traffic control methodology is integrated into the discrete event approach since it allows more flexibility and advantages over the time oriented [8].

3.1 Interaction With the Simulator

The simulator was developed in Java using the facilities of *ProActive* (a Java GRID middleware library for parallel, distributed and multi-threaded computing) for communication and managing the connection among various nodes (modules). When the simulation engine is executed it registers in a port of the local machine, allowing registering other node through the network if it is included in the grid.

As we have mentioned above, the simulator can be modeled as a discrete event system. The way in which the simulator works is as follows: each time an event occurs in the traffic network, this event models a change in the state of the system and is responsible to generate more events which depend on it. Each event has a time in which it is expected to be triggered. When events occur, they represent actions to be executed on the system. The events are stored in a future event list (FEL). Next picture shows graphically the previous ideas.

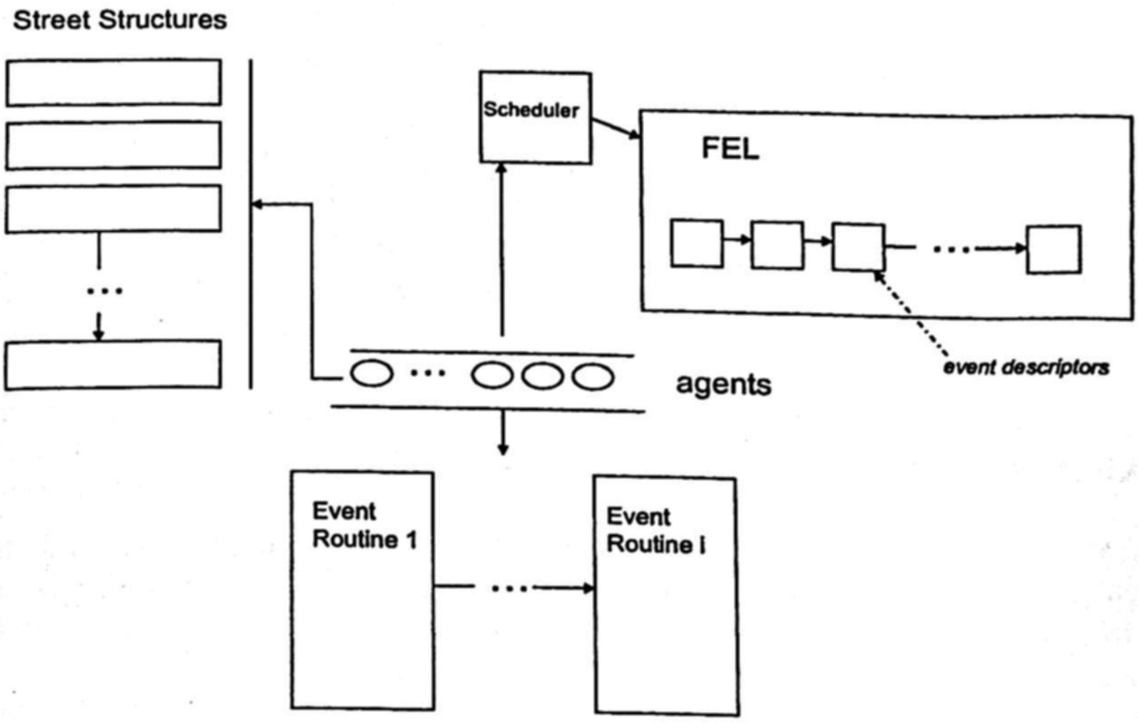

Fig. 1 Simulation Engine

The simulation engine generates a file configuration to save the results of the simulation (executed activities), including the following information: *Vehicle Identifier, Event Time, Link, Node, Lane, Distance, Velocity, Event Type, Acceleration, Driver, Passengers*, etc. that allows to get performance indexes such as traffic density, flow, etc.

Each time an activity is executed in the simulation engine this module sends the pertinent information about such activity to all the modules connected to the middleware. This resumes the operation of both control and visualization modules. The visualization module uses this information to process and visualize the events in a graphical manner. The control module uses this information to take decisions according to the state of the streets.

The control module is composed by several control agents. The control agent interacts with its environment, and it is capable to interact with others agents when it is necessary to coordinate a control policy. In figure 2 the interaction between the simulator and the control module is illustrated. This interaction is performed through a middleware that handles a future event list (FEL). The system operates as follows: the control module generates a set of events representing the light change instants, according to the current control policy; these events are stored in the FEL; the interface provides to the control module the information on traffic flow and density yield during the simulation execution through the middleware. This allows the agent the updating of the phase time of the current control policy.

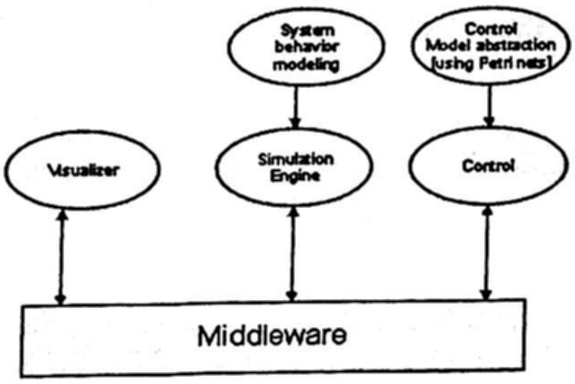

Fig. 2 Interaction between the control module and simulator

4 An Agent Architecture for Traffic Control

The urban traffic network approach is composed by several intersections and streets; each intersection is controlled by an agent with the following features: *interaction with its environment, decision making, proactive, communication with other agents in the system, unable to solve the problem by itself.*

The control agent is composed by several interacting modules charged to perform specific functions. In figure 3 it is shown the control agent architecture. The input of the control agent is provided by the simulation engine; this information regards the current traffic flow and density of all the streets in the traffic network. The output consists of an event sequence regarding the traffic light change instants; this sequence is added to the FEL.

Every control agent is concerned with the involved streets in the intersection where the agent is operating. Now the paper describes the functionality of each one of the agent's components.

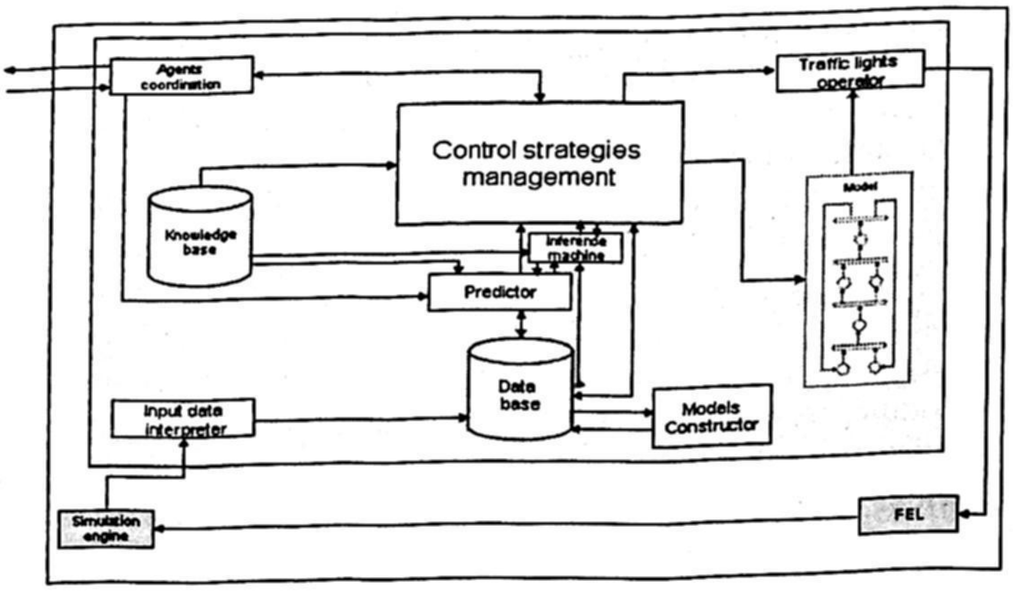

Fig. 3 Control Agent Architecture

4.1 The Control Agent Components

• The data base includes the initial configuration of the intersections, namely, the intersection layout, the allowed phases (stages of a control policy), initial time assigned to phases, and the state of the streets related to intersection.
• The model constructor is charged to transform (offline) the control policies stored on data base into timed Petri net models, to be used by other components of the agent.
• The input data interpreter analyzes the input information and computes traffic flow and density values on the adjacent streets to the intersection that is controlled by the agent.
• The knowledge base includes the rules decision used by the control management module and predictor modules to take decisions about the controller behavior. The rules decisions are organized as subsets according to the kind of decisions. Access to each group of rules is determined by metarules.
• The predictor computes future traffic conditions according to rules decision and supported by the surveillance function. The surveillance function ensures that the predictions are precise.
• The control strategies management module proposes the control policy according to the observed traffic conditions on the intersection. This module receives as input the flow Φ_i and density δ_i for every street i involved on intersection; the input is analyzed and then the module can decide if the current control policy must be changed.
• The model describes the current control policy defining the operation of the traffic light in the intersection; this policy is determined by the control management module. The model is a timed Petri net that specifies which movements have right of way in each phase of the policy, and the time elapse assigned to each phase.
• The traffic light operator executes the policy model. The sequence of events (firing of transitions) and their instants are translated into relevant events for the simulator; they are added to the FEL respecting the order of the events already included in the list.
• The agent coordination module is responsible to establish the communication with the other agents informing about the current control policy or requesting the performance indexes of streets concerning other intersections.

4.2 Control Agent Operation

The controller is a proactive agent that operates according the current situation of the environment simulation. It applies a pertinent control policy and delivers to the simulation engine the traffic lights behavior in the form of an event list. The behavior of the control agent is determined by the control strategies management module, which coordinates the rest of the agent components.

The agent adaptation to new traffic conditions is performed by the change of the control policies. This change may be accomplished in two ways: a) updating the timing of a current policy or b) replacing the control policy by another one.

Timing Updating

The policies management module decides, according to the traffic demand in the involved streets, if the current time elapses assigned to the phases of the policy must be modified. The change consists in the updating of some or all delays associated to places in the Petri net model of the policy.

Policies Replacement

When the change of traffic conditions is significant, the control management module may decide to substitute the current policy with another one, which is best adapted to the traffic behavior; the module requests to the model execution module to stop the current policy, updates the Petri net model with the new policy, and resumes the activity of the execution module.

Combining Policies

It is possible to define complex policies including more than one phase sequence and alternate their operation triggered by external events representing namely, pedestrian request, different traffic conditions, and priority vehicles. In figure 4 the model shown includes two sequences of phases (which can be considered as two policies).

The execution module may fire the sequence σl= (tl t2 t3 t4 t5 t6)* while the external event associated to tc1 does not appear; when such an event is present then tc1 fires and the sequence σl=(T1' T2' T3' T4')* is fired. The back to execute the first sequence of phases may be accomplished when tc2 (according to the associated external event) is fired.

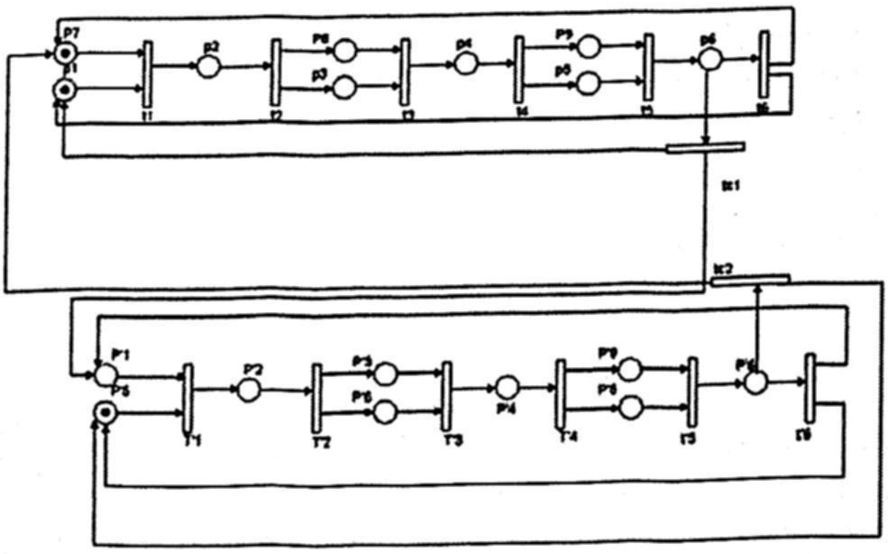

Fig. 4 Timed Petri Net model including two sequences of phases

Agent Interaction

Any change in the control policies, as described above, must be informed to the neighbor control agents through the interaction module. This information may be used by these agents to initiate a policy adaptation process. This means that if an agent perceives saturation on any street belonging to the intersection, this agent can decide to interact with other agents on adjacent intersections to decide the change of a control strategy.

5 Implementation and Results

The traffic control agent has been developed according to the proposed architecture using Java. A control strategy is executed by the traffic lights operator according to the state equation for the timed Petri net model. Traffic lights operator translates the transition triggers into significant light change events. The interaction with the rest of the simulator is performed through active objects from the middleware. Some methods are defined for communication between agents: *lookup_agent, start_dialog,accept_invitation, reject_invitation and send_information.*

Several tests have been performed on a LAN in order to test the functioning of the control agents. For illustration purposes the paper includes the simulation results obtained in a simple case study.

Consider the scheme of figure 5; it describes four intersections separated by a distance of 1Km. Parameters referring to the vehicles speed, random distributions of the vehicles, and others, are modified to observe the behavior of the control agent, when the traffic conditions change.

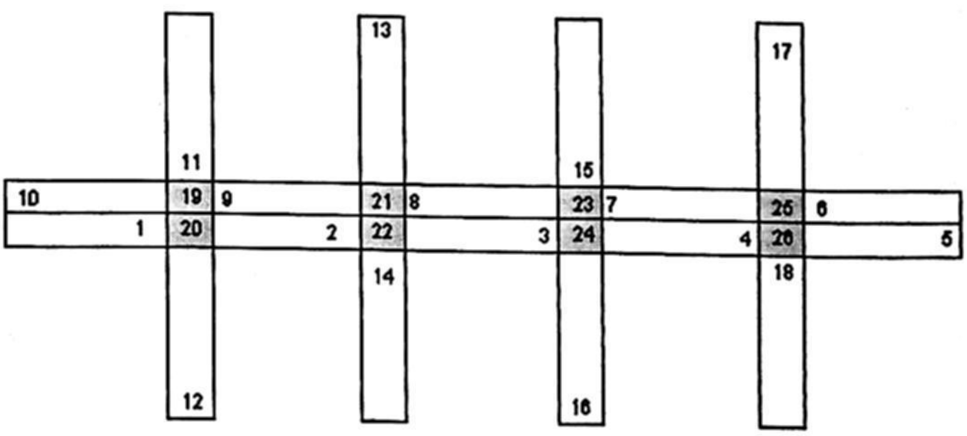

Fig 5 Traffic Network

Figures 6 and 7 show respectively, the starting and the ending of a simulator execution for the network described above. When simulation starts the queue length in the inflow streets is higher than outflow streets as we can see in figure 6; but at the end of the simulation, we can notice that the queue length on all streets is similar (figure 7). This suggests that the traffic conditions get stable.

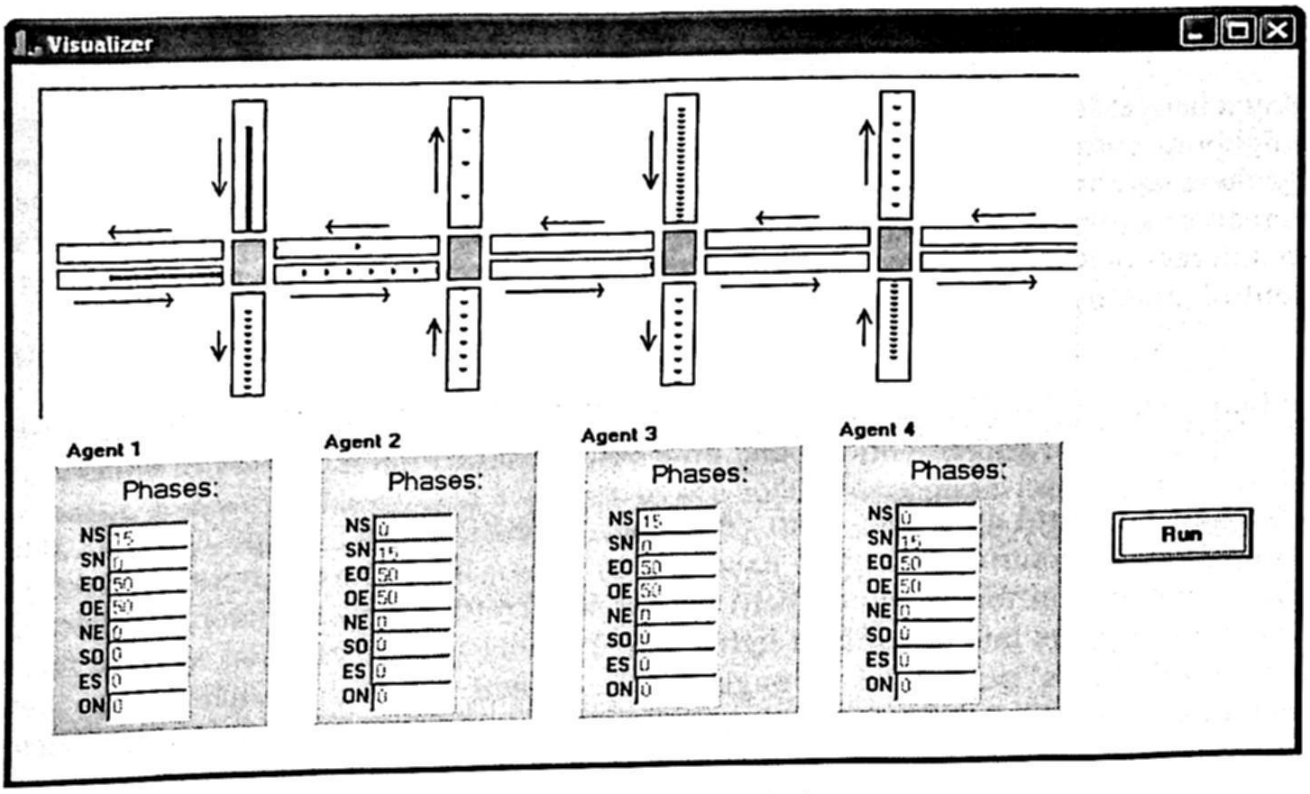

Fig. 6 Start of the simulation

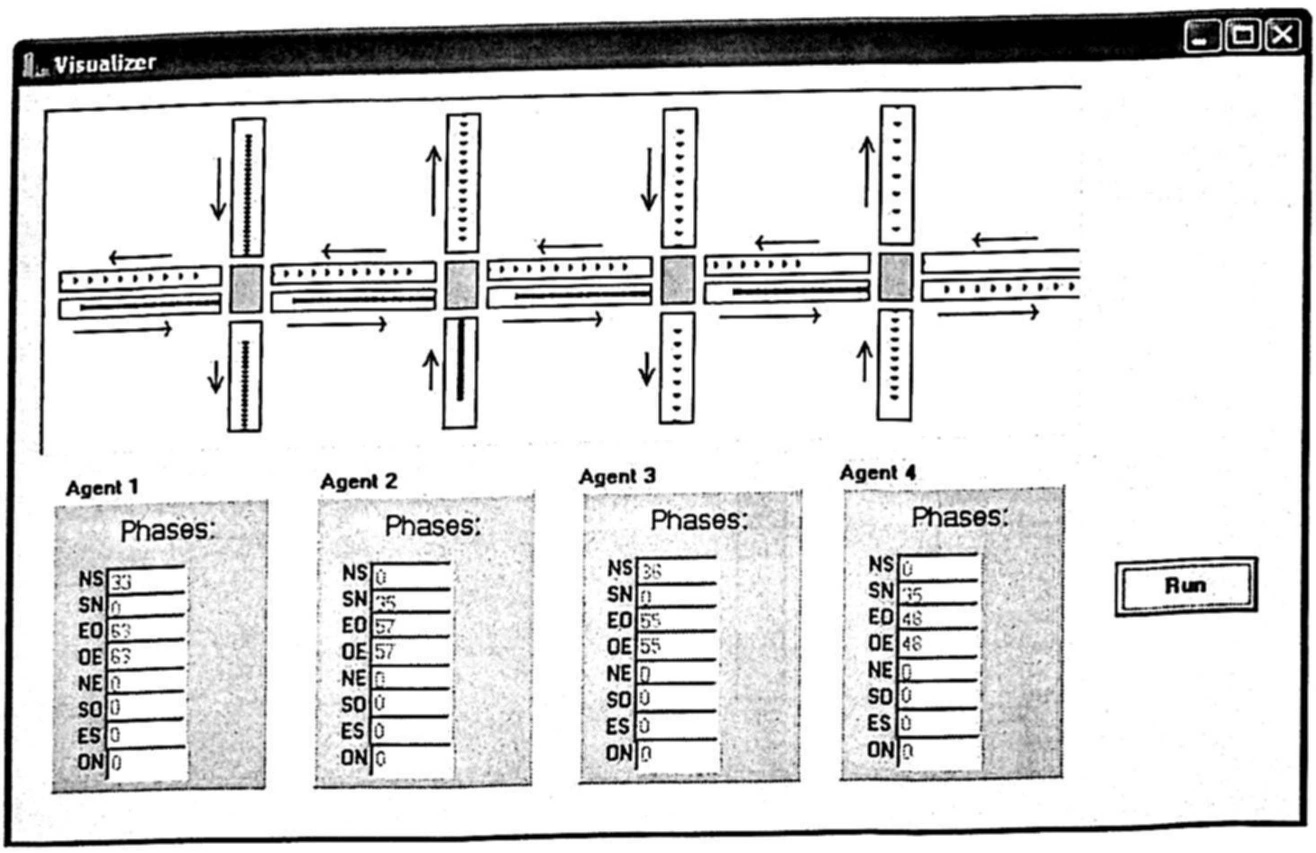

Fig. 7 End of simulation

In figure 8 the graphic shows the flow and density of vehicles regarding the 1 and 11 street segment, using a fixed control policy. The two-phase policy assigns 50 seconds to the west-east phase (link 1) and 15 seconds to the north-south phase (link 2) during all the simulation time. We can notice in both graphics the flow and density values are variable during the simulation time; that indicates that the fixed-time policy it is not appropriate for variable traffic conditions.

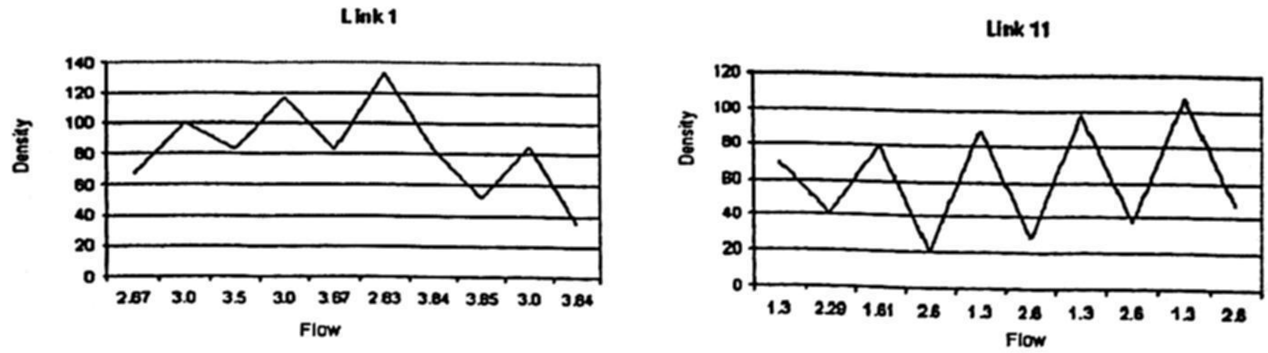

Fig. 8 Pretimed control for street segment 1 and street segment 11

Now we are going to analyze the results of applying an adaptive time policy. In figure 9 it is shown the flow and density of vehicles for the 1 and 11 street segments respectively; vertical lines in graphic represents the instant in which the times assigned to phases are updated. In the graphics we can notice several stages: at the beginning, density is high and flow is low; then while simulation time is growing the density is reduced and trends to be constant and the flow is increased and also trends to be constant during the rest of simulation; that is flow and density values trend to stabilize because of the dynamic updating of the time elapses assigned to phases.

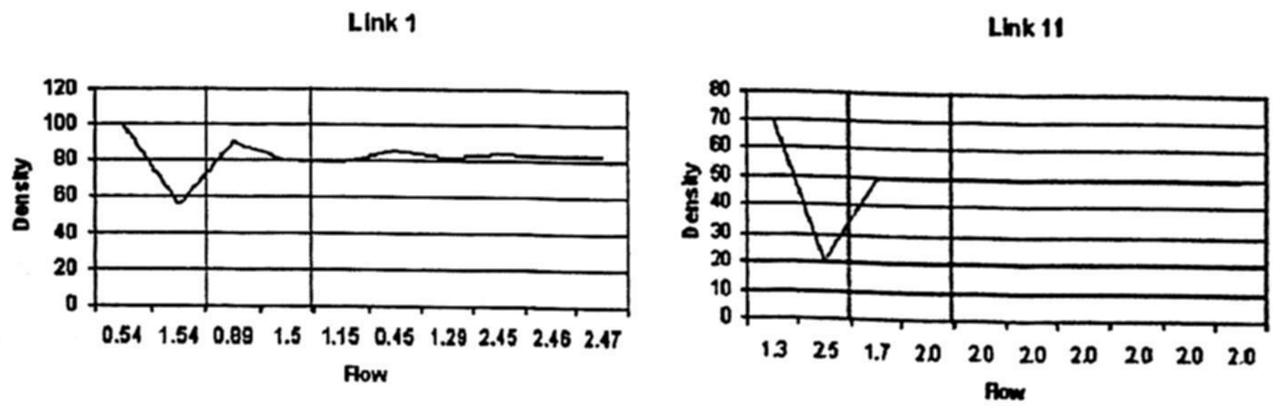

Fig. 9 Time updating control for street segment 1 and street segment 11

Now the paper presents the results of applying an adaptive control, in which besides of the times updating control, the model could be replaced by a model more appropriated to the traffic conditions. In figure 10 it is shown the flow and density of vehicles for the 1 and 11 street segments respectively; the first vertical lines in each graphic represents that the model has been replaced. As we can notice in the graphics, after the control policies are replaced, traffic flow increase and density decrease, but density and flow values are variable; when the time assigned to phases is updated then the density and flow values trend to be constant. As we can notice, the traffic conditions trends to stabilize when an adaptive control strategy is established.

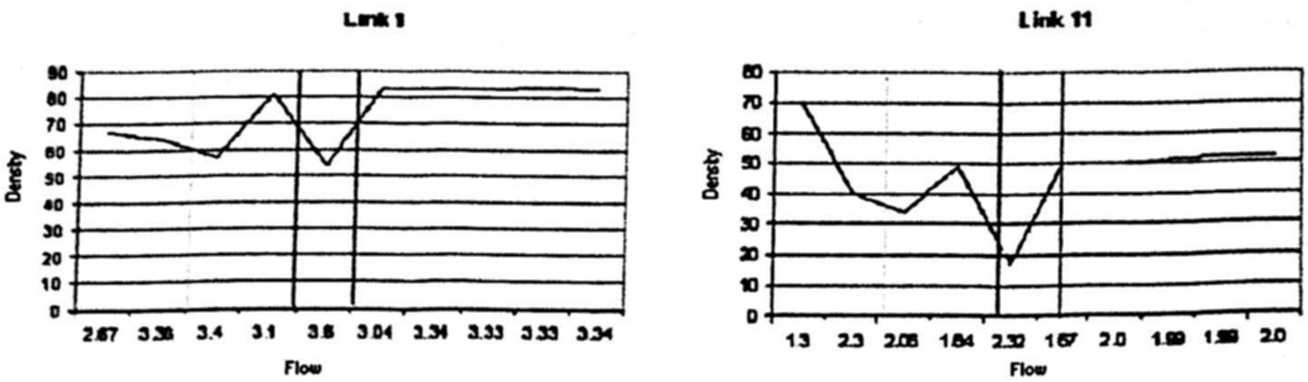

Fig. 10 Policies control replacement for street segment 1 and street segment 11

6 Conclusions

Urban traffic control is proposed based on a multi-agent approach. In our scheme the traffic light of each intersection is controlled by an agent; this agent proposes a control strategy according to the traffic demand. The control agent interacts with other agents assigned to the neighbor intersections. As we noticed during several tests the performance indexes of a traffic network are vastly improved when dynamic strategies for updating control policies were adopted.

Current research address: a) complex control policies dealing with external events (pedestrian requests, priority vehicles), and b) the definition of coordination strategies among control agents.

References

1. Carbone C. ."Control Oriented Discrete Event Models for Manufacturing and Transport Systems". Phd Thesis, Facoltà di Ingegneria Elettrica. 2003-2004.
2. G.F. Cetin, M. "Modeling traffic signal control using Petri nets". Intelligent Transportation Systems, IEEE Transactions on, Sept 2004, pp 177-187.
3. Julvez J., Boel R.. "Modelling and Controlling Traffic Behaviour with Continuous Petri Nets". D.G.A. ref B106/2001 and a European Community Marie Curie Fellowship, CTS, contract number: HPMT-CT-2001-00278.
4. DiFebbraro Angela, Giglio D, Sacco. "Urban Traffic Control Structure Based on Hybrid Petri Nets". Intelligent Transportation Systems, IEEE transactions on, Dec 2004, pp 224-237.
5. Danko A. Roozemond , Jan L.H. Rogier. "Agent controlled traffic lights". ESIT 2000, 14-15 September 2000, Aachen, Germany.
6. Marco Wiering, Jelle Van Veenen, Jilles Vreeken, Arne Koopman. "Intelligent Traffic Light Control". Institute of information and computing sciences, Utrecht University Technical Report. UU-CS-2004-029
7. Fei-Yue Wang. "Agent-Based Control for Networked Traffic Management Systems". IEEE Computer Society September 2005. Vol 20, No. 5. pp 92-96.

8. 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.
9. E. López-Neri, E. López-Mellado, A. Ramírez-Treviño. "Hierarchical Models of Urban Traffic Systems for Agent Based Simulation". International Workshop Optimization of Urban traffic Systems. Julio 2005, pp.1-12
10. Nwana H. S., Software Agents: An Overview, In Knowledge Engineering Review, Cambridge University Press, 1996.
11. Proactive V3.0.1 Documentation.
12. Fujimoto, R. M. 1999. Parallel and distributed simulation. In Proceedings of the 31st Conference on Winter Simulation: Simulation---A Bridge To the Future - Volume 1 (Phoenix, Arizona, United States, December 05 - 08, 1999).p-p 122-131.
13. Wooldridge M. "Intelligent Agents ". In G.Weiss, Editor: Multiagent Systems (MIT Press 1999).