Towards an Artificial Traffic Control System

Rosaldo J. F. Rossetti, Member, IEEE, Paulo A. F. Ferreira, Rodrigo A. M. Braga, Eugénio C. Oliveira

Abstract—This work reports on the use of the concept of Artificial Transportation Systems to implement a framework to allow the specification and test of new generation intelligent traffic control systems. A JADE implementation of a real agent application is linked to a virtual traffic domain to test control behaviour of traffic semaphores. Mixing reality and virtual environments is expected to foster the development of new generation of urban transport solutions. First experiments were carried out, which demonstrated the feasibility of the approach.

I. INTRODUCTION

Current discussions on future urban transport (FUT) systems bring about a series of concerns related to mobility and the quality of life in highly populated areas that challenge both the scientific community and society alike. Indeed, contemporary transportation systems have been experiencing considerable changes throughout decades, and the rapid growth of urban areas has defied engineers, decision-makers, and practitioners, who strive to cope with traffic congestions frequently encountered in most commuter journeys. In fact, congestion yields considerable economic, social and environmental losses, leading to a strong degradation of the quality of service in network infrastructure (and consequently to a reduced throughput). Minimising these losses is a hard task as the whole urban environment is very complex in nature, composed by a huge amount of heterogeneous interacting entities among which those that form the traffic system are also included.

Some attempts to cope with the potential limitation of road capacity have been put into practice, such as physical modifications to the infrastructure and the improvement of control systems. The former is no longer the best alternative to tackle such a problem due to a series of constraints. Besides the high cost of implementation, it causes disruptions and damages the environment. In the latter situation, some good advances and successful experiences have contributed to the reduction of problems related to traffic jams. However, and in spite of relatively good improvements they are able to produce, they can neither be considered a lasting solution. Alternatively, different approaches have been proposed that, on the other hand, rely on maximising the use of the current road capacity by directly influencing user behavioural patterns. The contribution of intelligent transportation systems (ITS) has been increasingly significant in this way, whose goal, among others, is to ensure productivity and efficiency by making better use of existing transportation systems. Recent advances in communication and in computer technologies have encouraged the use of such systems as a means to tackle problems in the field of traffic and transportation engineering, applying distributed solutions aimed at specific issues of user needs on an individual basis [1]. Thus, integrating all factors, both static and dynamic, which can somehow affect the traffic flow, is central to ITS. In this scenario, all components are expected to work together on a co-operative basis, seeking to maximise the efficiency of the system in which improvement results from directly influencing the real-time behaviour and interaction of all its elements.

FUT solutions bring the user to a central spot then, and strive to address a series of issues such as sustainability, privacy, equity, safety, accessibility, preservation of the environment, comfort, and so on, which are rather related to the qualitative assessment made by users with different perceptual abilities. This suggests a number of new performance measures that need to be accounted for and assessed through powerful and expressive tools. In this way, much work has been carried out either to adapt traditional approaches or to develop new-generation traffic and transport network models to meet FUT requirements, in which an explicit representation of interactions between demand and supply [2] should be supported.

In this work we intend to give first steps towards mixing reality and virtual environments as a means to better understand and devise intelligent transportation solutions accounting for FUT requirements. We use the concept of Artificial Transportation Systems to implement a framework to allow the specification and test of new generation intelligent traffic control systems. A JADE implementation of a real agent application is linked to a virtual traffic domain to test the proposed approach.

II. MAS AND ARTIFICIAL TRANSPORTATION SYSTEMS

The relationships identified in contemporary transportation
systems bring into evidence the social aspects involved in the interaction among various autonomous entities that inhabit a common environment. In fact, current transportation solutions, especially those related to traffic control and management systems, are achieving a high degree of autonomy and start interacting with users in a different dimension, as their peers. This emerging scenario rather turns society into a system formed of multiple heterogeneous components, with different abilities to perform their actions and achieve their goals. Their relations are quite complicated and can assume different forms of interactions, ranging from competition, impelled by selfish behaviours, to collaboration and cooperation, inspired by a more altruistic need to help others. This is especially true when resources are rather scarce and all elements are expected to make rational use of them. Affecting interactions at such an emergent transportation society becomes then imperative to the success of FUT, and appropriate approaches to model and assess their specificities are necessary.

Recently, a new paradigm in Computer Science has been widely spread and used by researchers to study social aspects of complex systems with applications to a variety of domains. The multi-agent systems (MAS) metaphor stems from Distributed Artificial Intelligence (DAI) and is especially well-suited to represent systems whose entities exhibit autonomy and some degree of interaction with one another and with their environment so as to accomplish individual or collective goals. Coined agents, these entities may also present other characteristics including reactivity, adaptability, pro-activity, and the ability to communicate and to behave socially. Their basic structure features “sensors” through which they can collect information from the environment, and “effectors” through which they can act and behave according to their plans [3]. Owing all these characteristics, multi-agent systems have been widely applied to modelling and simulation of a wide range of applications, naturally including transportation systems [4].

Not surprisingly, most works report on applying agents to urban traffic control and management so as to make them more autonomous and responsive to recurrent demand (e.g. [5] and [6]). Adaptability is also an important characteristic investigated by members of our team through the development of machine-learning techniques [7], [8], [9]. One reason for such an easy coupling between MAS and urban traffic control (UTC) can be found in the traffic domain’s characteristics of being naturally distributed, both spatially and functionally, such as agents in a multi-agent system. In agent-based approaches, there seems to be a common effort toward allowing UTC to work on a distributed-fashion, rather than in a centralised traditional way, as suggested in [10], [11], and [12]. Again, control is associated with local activities of controllers, whereas management is now associated with the coordination of various control agents [13], [14]. Albeit interaction between control and demand remains indirect, via interpretation of raw data read through surveillance systems, efforts have also been dedicated to representing driver behaviour and its underlying decision-making mechanism. Thus, both social interactions and the influence of real-time information are also approached [15], [16], [24]. Some other works concern applications to freight transport and interaction with service providers [17].

As the many examples demonstrate, the MAS metaphor represents a very propitious way to model FUT and all its complexity. Moreover, it also ease the representation of the social interactions now playing a decisive role in the overall system performance as users explicitly interact with each other, with their environment, and with newly deployed technologies, whose interfaces are more and more friendly reaching a wider range of users. With much work being carried out, there is clearly a trend toward a unified perspective of a multi-agent based model integrating all aspects of urban transportation. Among first attempts, Boucheffa et al. [18] suggested a scalable multi-agent model in which traffic systems were represented in terms of agents in different levels of granularity. Recently, a modelling methodology for urban traffic systems was suggested on the basis of more standardised and consensual agent-based modelling approaches [19]. In this kind of view, supply and demand are not dealt with as separate modules within a framework. Rather, their interaction is expected to emerge naturally from the interactions of various agents within the traffic environment, which can play their role either in the demand or in the supply side. Such a perspective evolved to the concept of Artificial Transportation Systems (ATS) as it has been defined in [20] and [21].

ATS allow FUT requirements to be modelled in a more straightforward way and tested in a controlled environment where different kinds of interactions can be verified. If on the one hand supply entities, such as control elements, act in a rather collaborative way trying to find an optimum performance set-up for the system, on the other hand users usually act in a competitive way, seeking self improvement and maximisation of own benefits disregarding the effects their actions can have upon others’ performance. Moreover, ATS give traditional transport analysis the possibility to work the other way round.

![Fig. 1. An integrated perspective for artificial systems analysis.](image-url)

Indeed, the customary way to analyse virtually any systems would include i) to devise a model of the real world,
ii) to apply assessment and analysis tools onto the model, so as to test different scenarios and draw conclusions on the system performance, which will support iii) the generation of new control strategies and management policies that will be applied back onto the real world. These iterations are illustrated in Fig. 1, through the solid lines. However, ATS bring a new approach to this traditional methodology of analysis, which rather integrates all three aspects. The application domain is conceptualised in terms of agents and three basic subsystems are identified, namely the real world, the virtual domain, and the control strategies and management policies inductor, which are multi-agent systems themselves. These three subsystems interact with one another, adding to the discussion above new channels of interactions as illustrated through the dashed lines. In this direction, Andrade et al. [22] gave a contribution bringing back into evidence the need for a robust representation of the environment as a means to ease parameterisation, allowing interactions among entities through an indirect communication mechanism. This is an important issue regarding scalability and integration, which was first approached as a meta-data structure to represent the network domain are intended to emulate the individual behaviour of traffic control systems, and intelligent transport solutions. It is conceptualised in terms of agents and three basic subsystems are identified, namely the real world (RW), the virtual domain (VD), and the control strategies inductor (CSI), which resembles the modules of Fig. 1.

The real world designates the real transport system in urban areas, where physical components, such as travellers, traffic control systems, and intelligent transport solutions cohabit and interact. These components are replicated into the virtual domain where their delegates (or agents) are instantiated. Therefore, the software agents in the virtual domain are intended to emulate the individual behaviour of physical components encountered in the real world. The whole population of active components in the real world is synthesised this way. Finally, the control strategies inductor is a subsystem formed of expert agents that observe such a synthetic population, can directly intervene on and experiment with it, and apply coordination policies to tune the behaviour of some elements in order to improve overall performance. Then operational parameters of the real world are adjusted to reflect the control policies tested in the virtual domain.

The interaction among these three subsystems is dynamic and iterative, allowing for real time intervention on the real world. First efforts toward the accomplishment of such an integrated environment are focused on the implementation of the virtual domain subsystem, which relies on an agent-based microscopic traffic subsystem, which is an important issue regarding scalability and integration, which was first approached as a meta-data structure to represent the network domain are intended to emulate the individual behaviour of traffic control systems, and intelligent transport solutions. It is conceptualised in terms of agents and three basic subsystems are identified, namely the real world (RW), the virtual domain (VD), and the control strategies inductor (CSI), which resembles the modules of Fig. 1.

A JADE container is nothing more than different JADE --the Directory Facility Agent (DFA) that is sort of a --the Agent Management Service (AMS) that is the core --the Remote Management Agent (RMA) that handles the graphical user interface; --the Agent Management Service (AMS) that is the core agent that keeps track of all JADE programmes and agents in the system; and,

--the Directory Facility Agent (DFA) that is sort of a yellow pages structure where agents publish their services.

A JADE container is nothing more than different JADE environments that can be used, e.g. to aggregate agents that conceptually have the same objectives. A typical JADE-based application will possibly encompass several different containers where the various agents might be executing from several different computers (or processing units). There is one central agent (the AMS) that is responsible for keeping
track of all agent addresses. Similarly to AMS, there can be only one DFA. The JADE platform that runs the AMS and the DFA is called the “Main-Container” and must be the first programme to be started. All other agents may be started in different machines by specifying the host address of the Main-Container so it can connect to the existing AMS and DFA agents. The host specification is not necessary if agents are to be started in the same machine as the Main-Container.

This architecture was used to the implementation of a traffic light manager application (TLMA) in Java, which is an agent able to control a single traffic light intersection. Fig. 2 illustrates the architecture overview of TLMA.

![Fig. 2. The TLMA Architecture, implemented in JADE.](image)

The agent architecture is based on the generic JADE agent architecture and is detailed in Fig. 3. The agent is composed of a simple ontology where it organises and represents its knowledge of the world state.

![Fig. 3. Semaphore Agent Architecture](image)

Such ontology is basically formed by three classes:

* **Intersection Semaphore class** represents the intersection that the agent controls. It contains information about the intersection position, the neighbour intersections that have roads connecting them to it, the intersection traffic lights and the manoeuvres they control and a set of possible traffic light plans that can be applied to this intersection.

* **Intersection Connection class** represents the neighbour intersections that have roads connecting them to the controlled intersection. It contains information about the traffic flow that arrives or leaves the controlled intersection that is coming from this neighbour intersection. It also saves the references of the traffic lights that regulate traffic coming from the neighbour intersection to the controlled intersection.

* **Traffic Light class** represents a traffic light of the controlled intersection. It contains information about the manoeuvres it controls (given by neighbour intersections where traffic comes from and the ones where traffic goes to, when exiting the intersection regulated by the current traffic light), its current state, remaining time to change state, total green, yellow and red timings, a list of traffic lights controlling conflicting movements and a list of those controlling non-conflicting ones, and finally the number of vehicles that performed the manoeuvre it regulates.

* **Traffic Light Plan class** represents a possible traffic light plan of the controlled intersection. It contains information about the traffic light phases and their different times.

The implemented agent has a GUI that allows user interaction to configure the connection parameters governing the interaction with the simulator. It also allows the user to choose the intersection that the agent should control. Through this interface the user can analyse the data that the agent receives, such as traffic light states and intersection traffic flows.

The **behaviour manager** module is responsible for activating the correct agent behaviours. For example, before the agent starts to regulate a traffic light intersection, it is necessary that a user chooses the intersection that it should manage. To give this option to the user the behaviour manager must activate the **request traffic lights intersection** (TLI) behaviour, so it renders in its GUI a list of manageable TLI. After the user chooses an intersection the **Behaviour Manager** must activate the **request registration** behaviour. If the registration is successful the agent will start receiving updates from the running simulation, and based on this information it may or may not decide to change the intersection control plan. If it decides so, a **request plan change** behaviour must be activated.

The **private inbox of ACL messages** module exists in every JADE agent, and allow communication between agents. The current prototype does not use this module yet, but when it will be decided to implement the cooperation between semaphore agents this module will be responsible for receiving, sending and interpreting messages.

The **simulator communicator** module basically contains two classes, namely the CommandHandler and the **ConnectionsConfiguration**. This module is responsible for message exchanging between the agent and the simulator.

In the current prototype the **capabilities** set contains user defined traffic light plans. An agent can choose from this list a possible traffic light plan to be applied to the traffic light intersection based on new received data. This will be better detailed later on in this paper.

IV. Connecting Real Agents to Virtual Domains

In this section we will describe some experimental results obtained for a simple example network, with the objective to
evaluate the architecture presented and test some of its features.

A. Scenarios Definition

The test scenarios are based on a very simple network that has been coded in the network XML format accepted by the simulator prototype, as illustrated in Fig. 4. Intersection 1 is regulated by traffic lights and its configurations are described later on. Intersection 4 is a prioritised junction, where vehicles approaching from the right of other vehicles have priority to go through the intersection.

![Fig. 4. Schematic representation of the example network.](image)

Three different traffic configurations were tested with the objective of understanding the influence of the implemented traffic control agent on the traffic flow of the roads that connect intersection 1. The test scenarios were set as follows:

- **Low Traffic scenario**, where the flow generated at source intersections are: 250veh/h at intersection 0, 220veh/h at intersection 2, 150veh/h at intersection 3, and 150veh/h at intersection 5.

- **Medium Traffic Scenario**, where the flow generated at source intersections are: 700veh/h at intersection 0, 600veh/h at intersection 2, 200veh/h at intersection 3, and 800veh/h at intersection 5.

- **High Traffic Scenario**, where the flow generated at source intersections are: 1250veh/h at intersection 0, 1250veh/h at intersection 2, 900veh/h at intersection 3, and 900veh/h at intersection 5.

B. The TLMA Agent Behaviour

At the current stage of this work, the TLMA is provided with a fixed number of control plans, with the same cycle but different phase times. The TLMA agent basically selects one of these plans whenever one of the phases needs to be improved. Such a selection is made on the basis of a greedy behaviour, accounting for the difference between the highest (“worst”) traffic volume and the lowest (“best”) traffic volume observed by the agent in all downstream links. Whenever such a difference is greater than a predefined threshold, the control plan that maximises the green time of the degraded approach into the junction is selected for execution. So, let us say that that $\bar{v}_{ij}$ is the worst traffic volume at junction $i$ relative to phase $j$, whereas $\bar{v}_{ij}$ is the best volume observed. The difference $\Delta v$ is then given by

$$\Delta v = \bar{v}_{ij} - \bar{v}_{ij}.$$ 

The decision on whether to change plan, will depend on the threshold $\tau$, which at the moment is chosen a priori, and can be summarised as follows:

$$\begin{align*}
&\text{if } \Delta v > \tau, \text{ select another priori;} \\
&\text{ otherwise, keep same plan.}
\end{align*}$$

Despite this simple behaviour, the proposed architecture could be tested. Nonetheless, the implementation of more complex behaviours to dynamically generate alternative plans, as well as to define adaptive values for parameter $\tau$ can be easily attached to the behaviour of the TLMA and will be the focus of further developments.

C. Some Results

Some simulation experiments were carried out so as to test the proposed architecture. The TLMA agent was connected to the simulator in order to control intersection 1 accounting for the recurrent traffic volumes generated by the sources as previously described in the three different scenarios, namely under low, medium and high traffic flows. These scenarios were inspired in a typical 24-hour traffic flow profile, which could be interpreted as the off-peak, lunch peak and morning peak periods respectively.

In the first scenario, although modifying the threshold value most certainly influences the results, even so improvements are not significant. This is mostly due to the recurrent low flow at the junction, which would not justify the adoption of different control plans.

As for the second scenario, contrary to a single-plan control approach, the TLMA agent, despite its simple behaviour, proved to be very effective in keeping a relatively stable traffic flow at acceptable journey times for all roads approaching intersection 1. Table I illustrates this trend.

<table>
<thead>
<tr>
<th>Road</th>
<th>1st Plan</th>
<th>2nd Plan</th>
<th>3rd Plan</th>
<th>TLMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Road 1</td>
<td>442.5</td>
<td>121.5</td>
<td>38.8</td>
<td>94.4</td>
</tr>
<tr>
<td>Road 2</td>
<td>41.5</td>
<td>515.9</td>
<td>417.1</td>
<td>117.3</td>
</tr>
<tr>
<td>Road 3</td>
<td>35.6</td>
<td>32.8</td>
<td>35.1</td>
<td>35.2</td>
</tr>
<tr>
<td>Road 4</td>
<td>20.5</td>
<td>20.3</td>
<td>22.35</td>
<td>24.1</td>
</tr>
<tr>
<td>Average</td>
<td>135.03</td>
<td>172.62</td>
<td>128.33</td>
<td>67.75</td>
</tr>
</tbody>
</table>

As it can be seen from Table I, three different plans were firstly simulated on a single-plan control approach, which
In the-loop simulation can be used to test new gene ration transport. In this work we have demonstrated how software-technique, especially in critical systems, such as traffic and the implementation of more complex behaviour for the traffic control systems. Next steps in this research include applications in virtual domains is a very promising and calibration are still issues to be addressed in the integration of physical agents is also intended, and validation implementation of our virtual domain.

V. CONCLUSION

In this work we have proposed the integration of real agents and virtual environments as a means to effectively implement artificial transportation systems. The JADE framework is an open-source software platform to implement real agent-based systems in Java, and is only limited by aspects of the Java programming language. In practical terms, traffic controllers could perfectly be implemented in JADE. On the other hand, testing real applications in virtual domains is a very promising technique, especially in critical systems, such as traffic and transport. In this work we have demonstrated how software-in-the-loop simulation can be used to test new generation traffic control systems. Next steps in this research include the implementation of more complex behaviour for the controller agent and integration of multiple real agents to improve performance of a more complex network. The integration of physical agents is also intended, and validation and calibration are still issues to be addressed in the implementation of our virtual domain.

ACKNOWLEDGMENT

Authors gratefully acknowledge the financial support from GRICES-CAPES Cooperation Programme and from the Dept of Electrical and Computer Engineering, Faculty of Engineering, University of Porto.

REFERENCES