MULTI-AGENT ARCHITECTURE FOR SIMULATION OF TRAFFIC WITH COMMUNICATIONS

Pedro Fernandes and Urbano Nunes
Institute of Systems and Robotics, University of Coimbra, Polo II, Coimbra, Portugal

Keywords: Multi-agent systems, Traffic simulation, Network simulation, Intelligent Transportation Systems.

Abstract: Inter-vehicle communications, in the context of Intelligent Transportation Systems, will probably bring a significant improvement in both traffic safety and efficiency. In order to evaluate in what measure this is true, traffic simulations that take into account the communications between vehicles are needed.
In this paper, we propose an agent-based architecture, in which the simulation and management of the inter-vehicle communications are integrated in the simulation of vehicles, in a hierarchical multi-agent environment. An overview of multi-agent methodologies, platforms, among other, is also presented.

1 INTRODUCTION

Human transport in urban spaces relies mostly on individual vehicles, congesting the transportation networks. Studies and simulations of traffic have been made for decades, through macroscopic, mesoscopic and microscopic traffic simulators.

Recently, in the context of Intelligent Transportation Systems (ITS), vehicle to-vehicle (V2V) and vehicle to infrastructure communications (V2I) are being developed, namely the DSRC (Dedicated Short Range Communications), operating in 5.9 GHz band. The standardization process is almost finished under IEEE 802.11p/IEEE 1609.x (also designated by WAVE: Wireless Access in Vehicular Environments) and IEEE 1556 standards. In EU, the International Organization for Standardization (ISO), under the Technical Committee TC204, is working in similar standards – Communication Air Interface Long and Medium Range (CALM) – to ensure European-wide inter-vehicle communications interoperability.

To study the impact that such systems may have in the near future, efforts to integrate traffic and network simulators have been pursued. However, a useful solution has not been reached yet.

The integration of both traffic and network simulations in a system may be considered a complex task, due to a vast set of reasons, such as the intrinsic complexity of traffic theory, the wireless network transmissions involved, the real-time constraints and the distributed nature of the system, among others. At the present, traffic theory does not account to driver behavior changes due to the existence of communications. Therefore, equation-based modeling is not the most appropriate method to use in simulation. Agent-based modeling allows the development of a more adaptive system, and although system validation may be more difficult, it can be done at both system and individual levels.

2 RELATED WORK

The use of intelligent agents in traffic simulation is an emergent area of research. Table 1 presents some of the works in this area and simulators integration.

Table 1: Related work.

Vogel and Nagel (2005)	Multi-agent simulation model with application to Berlin traffic.
Hallé et al., (2004)	Agent-based architecture to develop centralized and decentralized platoons.
Li et al., (2006)	Urban traffic control system using multi-agent technology.
Dresner and Stone (2005)	Agent-based simulation of a traffic intersection.
Eichler et al., 2005	Coupling traffic and network simulators and a V2V messaging application.
Piorkowski et al., 2006	Network- and application-centric evaluation oriented architecture.
Avila et al., 2005	Intersection warning system, coupling traffic and network simulators.

Fernandes P. and Nunes U. (2008).
MULTI-AGENT ARCHITECTURE FOR SIMULATION OF TRAFFIC WITH COMMUNICATIONS.
In *Proceedings of the Fifth International Conference on Informatics in Control, Automation and Robotics - RA*, pages 215-218
DOI: 10.5220/0001501202150218

3 DEVELOPMENT ISSUES

According to Wooldridge (2002), “an agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives”. Autonomy, situatedness, reactivity and proactivity are some important characteristics of agents. In a multi-agent architecture, issues like organization, coordination and security are also relevant.

To develop a MAS system, a disciplined approach should be followed, and an appropriate platform should be chosen, along with communication standards between agents – preferably based on open standards – and appropriate ontologies. The simulation platform must also be selected or developed.

3.1 Methodologies

Several proposed methodologies to develop a MAS may be considered. Prometheus, Gaia and Tropos are some of the examples in the literature. However, not all existing methodologies are appropriate for every problem. Some of them aim at generality. Others focus more on specific platforms and languages, gaining in detail and adaptability.

Prometheus methodology was proposed by Padgham and Winikoff (2002). According to the authors, the reason why they proposed a new methodology was the methodology claimed detail, support of BDI (Beliefs, Desires and Intentions) agents, scaling ability and tool support. To support design and development of multi-agent systems using Prometheus, Padgham and Winikoff developed the Prometheus Design Tool, that implements the three phases of Prometheus and process some consistency checking.

Gaia methodology presents a general approach, to allow its use for a broad type of agent-based systems. However, this characteristic, which is one of its strengths, is also its most pointed weakness, since the detailed design phase and implementation have intentionally been left out.

Table 2: Methodology phases.

Methodology	Phases
Prometheus	1-specifications; 2-architectural design; 3-detailed design; 4-implementation.
Gaia	1-requirements; 2-analysis; 3-design.
Roadmap	1-requirements; 2-analysis; 3-design.
OPM/MAS	1-requirements; 2-analysis; 3-design; 4-deployment.

Other methodologies appear in literature, namely ROADMAP, Tropos, SODA, MESSAGE, MaSE, MAS-CommonKADS, AOR, OPM/MAS, MASSIVE, Ingenias, DESIRE, PASSI and AgilePASSI.

In Table 2, the phases of some methodologies are presented.

Prometheus seems an appropriate methodology for initial system development. All the relevant phases are covered conveniently, and PDT tool allows consistency and completeness checking through the steps of each of the phases.

3.2 Platforms

Choosing the right platform for the problem domain at hand is not a trivial task. The choice is closely connected with the methodology adopted.

Follows a short description of some platforms:

Jade framework is probably the most used agent-oriented middleware. Is an open source distributed middleware system, compliant with FIPA specifications, that implements both white and yellow pages, agent mobility, ontologies and content languages, among other features. JADE does not provide, however, direct support to the development of BDI agent architectures.

Jadex is a software framework for the development of goal-oriented agents following the BDI model. Since JADE platform does not allow direct implementation of this model, Jadex, using JADE, allows the creation of rational agents. Jadex agents have two main components: an agent definition file (ADF), coded in XML, and Java code. Jadex BDI metamodel is specified in XML Schema.

Jason is an interpreter of the an extended version of AgentSpeak(L), allowing agents to be distributed over the net using Simple Agent Communication Infrastructure (SACI). Jason is available as open source and uses jEdit (http://www.jedit.org) as IDE.

JACK™ is a commercial agent platform, which uses syntactic and semantic extensions of Java that allows the implementation of BDI agents.

The use of an open source platform is preferable. Moreover, the compliance with FIPA specifications is important to allow interoperability of the systems. JADE platform provides those and other features.

Table 3 presents some platform characteristics.

Table 3: Platform classification.

Platform	Open source	BDI	Compliance	White & yellow pages
JADE	Yes	No	FIPA	Yes
Jadex	Yes	Yes	FIPA*	Yes*
Jason	Yes	Yes	KQML	No
JACK™	No	Yes	FIPA	Yes

* with JADE

3.3 Ontologies and Languages

Communication is a valuable tool for agents to interact, exchange information and request services. At the present, Ontology Web Language (OWL) is the language of the Semantic Web that is being standardized by the World Wide Web Consortium.

An agent platform must allow the use of content language (e.g. FIPA-SL Content Language Specification), and communication languages (e.g. FIPA-ACL Agent Communication Language).

3.4 Simulation

Multi-Agent Based Simulation is considered the support of choice for the simulation of complex systems, replacing or integrating with other micro-simulation techniques, most of them object-oriented.

4 THE MODEL

The model proposed consists of a novel multi-agent system that manages the communications inside each vehicle and simulates the communications between each of them and the infrastructure. Inter-vehicle communications are managed by an agent-based module that simulates real wireless communications between vehicles, using the appropriate standards. To allow interoperability, the platform supporting the development of the proposed multi-agent system complies with FIPA specifications.

The architecture will be tested in the context of an intersection, where the management of communications and localization of the vehicles will have both a distributed and a centralized component. This option aims to provide simulation functionalities at the communication level that, in the reality, would be provided by the transmission media. Moreover, localization of hazardous situations (vehicles without communications, pedestrians) is better provided by centralized facilities.

The architecture proposed to the multi-agent system is depicted in Figure 1.

4.1 Multi-agent Architecture

A brief description of the main agents involved in the proposed architecture follows:

Network Simulator: The main function of the Network Simulator (NS) is to receive all communications between Communication Manager agents, and simulate the network transmission between them, considering the environment and the location of each one. Appropriate communication standards must be used by this agent, namely DSRC and CALM.

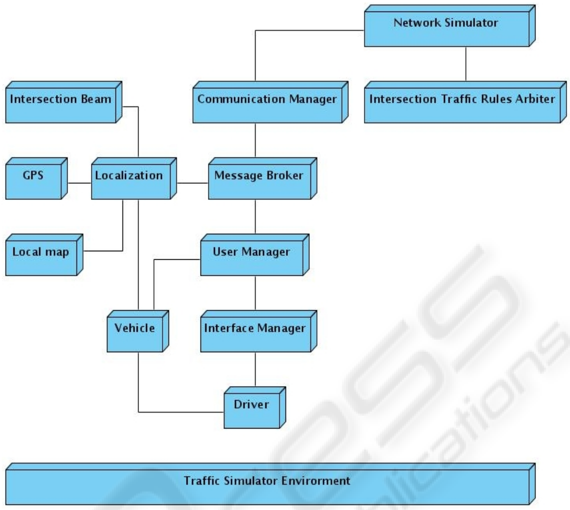

Figure 1: Multi-agent architecture.

Intersection Traffic Rules Arbiter (ITRA) must deal with intersection control of traffic, recording all traffic events and dealing with resolution of conflicts between User Managers (UM). With low traffic throughput, we may have a distributed control of traffic, where UM may agree with the priority of each other, always under ITRA supervision. As traffic flow grows, ITRA will have to validate all UM decisions, eventually overcoming some of them. In a high traffic flow scenario, all traffic rules decisions must be taken by ITRA, and vehicles become “data probes” of the centralized traffic rule management. Although this might seems contradictory with the choice of an agent-based system, real-time constraints impose the option presented above.

Communication Manager (CM) manages communications between the vehicle and external systems, such as the infrastructure and other vehicles. In both cases, NS is used as an intermediary, to simulate wireless network transmissions. Each vehicle communicates through its own CM.

Message Broker (MB) must manage all internal messages, and has the incumbency of filtering and its prioritization, ensuring that critical messages are dealt first by the appropriate agents. In this scheme, MB may delay low priority messages or, in some cases, even discard such messages.

User Manager (UM) main function has to do with decisions about the priority of the vehicle, with the agreement of all vehicles in its direct neighborhood, always under ITRA supervision. As stated before, as traffic flow grows, the decisions are taken by ITRA, in a centralized manner. To avoid deadlocks, all the decisions must be taken with anticipation, allowing the forecast of possible deadlocks and its resolution before they actually occur.

Interface Manager (IM) agent deals with the selection of the most appropriate message interface to the driver, taken in account the type of message.

Localization agent determines the localization of the vehicle in the intersection map, using GPS data and an intersection beam signal, and compares its position with neighbor vehicles positions, periodically transmitted through wireless communications. This agent must decide whether the situation is critical, based on position and vehicle data, and warn UM in case of imminent danger.

Vehicle agent gathers vehicle data (e.g. speed, acceleration, brakes, steering) and feeds Localization agent with that information. UM receives also similar feedback. Moreover, this agent gets commands issued by Driver agent.

Driver agent deals with the control of the whole vehicle. It receives information, whether critical or not, via IM agent and responds accordingly to that information and the type of driver modeled. For that purpose, Driver agent maintains a driver type database. This agent issues commands to Vehicle agent directly and indirectly through IM.

Traffic Simulation Environment represents the environment where the agents evolve. One of its main functions is to provide communications between agents, in the platform level, allowing appropriate management of agents' percepts and actions. Graphical presentation of simulation results will also be directly connected with this component.

5 CONCLUSIONS AND FUTURE WORK

In this paper we propose an architecture in which the simulation and management of the inter-vehicle communications are integrated in the simulation of vehicles, in a hierarchical multi-agent environment. We also present a short survey of existing methodologies, platforms, ontologies and languages, and suggest some possible choices to allow appropriate system implementation.

MAS development using the appropriate methodology, the implementation of the solution in the selected platform, the validation of the process and final deployment will follow.

ACKNOWLEDGEMENTS

This work was supported by Institute of Systems and Robotics and Fundação para a Ciência e Tecnologia under contract NCT04:POSC/EEA/SRI/ 58016/2004.

REFERENCES

Avila, A. et al., (2005). A complete Simulator Architecture for Inter-vehicle Communication Based Intersection Warning Systems. In *Proc. 8th Int. IEEE Conf. Intelligent Transportation Systems*, Vienna, Austria.

Braubach, L. Pokahr, A., & Lamersdorf W. (2006). Tools and Standards, Chapter of Multi-agent Engineering - Theory and Applications in Enterprises. Springer Series: *Int. Handbooks on Information Systems*, Editors: S. Kirn, O. Herzog, P. Lockemann, O. Spaniol.

Dresner, K. and Stone, P., 2005. Multiagent Traffic Management: An Improved Intersection Control Mechanism. In *The Forth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 05)*, pp. 471-477, Utrecht, The Netherlands.

Eichler, S et al. (2005). Simulation of Car-to-Car Messaging: Analyzing the Impact on Road Traffic. In *Proc. of the 13th IEEE Int. Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, vol. 00, pp. 507 – 510.

Hallé, S., Laumonier, J., & Chaib-draa, B. (2004). A Decentralized Approach to Collaborative Driving Coordination. In *IEEE Intelligent Transportation Systems Conference*, Washington, D.C., USA.

Li, Z., Wang, F., Miao, Q., & He, F. (2006). An Urban Traffic Control System Based on Mobile Multi-Agents. In *Proc. of IEEE Int. Conference on Vehicular Electronics and Safety*, Shanghai, China.

Padgham, L., & Winikoff, M. (2002). Prometheus: A Methodology for Developing Intelligent Agents. *In proceedings of the Third International Workshop on Agent-Oriented Software Engineering, at AAMAS'02*.

Piorkowski, M. et. al. (2006). Joint Traffic and Network Simulator for VANETs. *Poster Session, MICS 2006*, Zurich, Switzerland.

Sturm, A., & Shehory, Onn. (2003). A Framework for Evaluating Agent-Oriented Methodologies. In M. Winikoff P. Giorgini, B. Henderson-Sellers, editor, *Proc. of the Int. Bi-Conference Workshop on Agent-Oriented Information Systems, AOIS*.

Vogel, A., & Nagel, K. (2005). Multi-agent based simulation on individual traffic in Berlin, In *Proc. of the Int. Conference on Computers in Urban Planning and Urban Management (CUPUM'05)*, London.

Wooldridge, M. (2002). *An Introduction to multi-agent Systems*. John Wiley & Sons, Chichster, U.K.