Volatile knowledge for mobile agents : application to autonomous vehicles management

Utku KETENCI, Emmanuel ADAM, Emmanuelle GRISLIN, and René MANDIAU

Lille Nord de France, F-59000 Lille, France
UVHC, LAMIH, F-59313 Valenciennes, France
CNRS, UMR 8201, F-59313 Valenciennes, France
{emmanuel.adam,emmanuelle.grislin,rene.mandiau}@univ-valenciennes.fr
http://www.univ-valenciennes.fr/LAMIH/

Abstract. Completely autonomous vehicles in traffic should allow to decrease greatly the number of road accident victims, and should allow gains in terms of performance and economy. Interactions among the different vehicles allowing them to choose the best path, the best behaviour is one of the main challenges.
We propose in this paper a model of volatile knowledge dedicated to mobile agents on a traffic network. This model of knowledge and the principles of interactions allow to propagate new knowledge with a limited number of messages. For that, and to represent specificities of human driver, a degradation coefficient of the knowledge is proposed.
The principles have been validated by a simulation with software agents, and by a real application on mobile robots acting like autonomous vehicles.

Keywords: multi-agent coordination, mobile robots, trafic simulation, knowledge propagation

1 Introduction

In many studies on traffic supervision, the optimization of traffic flow as well as new road infrastructure [1], attempt to deal with collective interests and individual interests. We think that completely autonomous vehicles in traffic should allow to decrease the number of road accident victims greatly, and should allow gains in terms of performance and economy. Developing models of the interaction among the different vehicles is one of the main challenges [2] to optimize traffic flow with autonomous vehicles.

The simulation of traffic is often used to evaluate traffic flow optimization methods. Two approaches allow management policies for scheduling vehicles flows: centralized approaches and distributed approaches. A means to bypass the limitations of centralized approaches (lack of flexibility, difficulty to adapt to changes ...) is to decentralize the traffic simulation ([3], [4]); in this context agent-based approaches seem to be the most appropriate ([5], [6]) because of their modeling advantages ([7], [8]):

- The modeling system is composed by autonomous entities/agents,
- The entities/agents have different capacities and behave heterogeneously,
- The entities/agents are situated and their behaviors and interactions depend on their spatial properties,
- The entities/agents adapt their behaviors according to environmental changes and to dynamic relations among others agents,
- The number of entities/agents and number of interactions is high (more than dozens of agents),

These properties are typically those of the road traffic management where the drivers behave as such autonomous entities.

The behaviour of a driver is a multi-task activity: the tasks require different levels of cognition and therefore different levels of information. The driving tasks are classified in 3 groups according to level of cognition: Strategical, Tactical and Operational [9][10]. While the strategical tasks require a high level of cognition (search of the best roads to reach a destination for example), the operational ones require low level of cognition (reactive action to drive the vehicle according to the local perception (to brake, to turn to avoid an obstacle for instance)). The tactical tasks correspond to middle level of cognition (for example, they correspond to tasks allowing to find the best direction, or local short path to reach the other side of an intersection; or allowing to make the decision to let pass another vehicle in order to clear a possible situation of mutual blocking). The switch from the tactical level to the strategical level can be activated by the unability to reach the objective by using the available tactical tasks. In this schema, the higher tasks are the combination of the lowers. For example, lane changing can be seen as a combination of orientation and speed control.

In this paper, we will focus on a strategical task which is Itinerary Planning according to knowledge about the traffic network, and, in the implementation inside mobile robots, we focus on operational tasks regarding the control of the trajectory. As a traffic network is a dynamic environment where roads can be blocked or prone to slowing down, the knowledge must represent these dynamics, this volatility of the perceived information. So, our objective is not to build a new protocol of knowledge propagation (such protocols have already been proposed [11]), but to propose a management of dynamic knowledge, created from the perception of mobile agents.

The next section of the paper presents the model of volatile knowledge we propose and use to simulate knowledge sharing between drivers (human or not). The section 3 presents the simulator of road traffic that we use to validate our model, and the section 4 presents the application of knowledge sharing between mobile robot interacting with wireless connection. The last section draws our conclusions and gives some perspectives for future research.

2 Volatile knowledge for traffic network management

In the context of a project that aims at studying the impact of information communication between drivers (human or not) and infrastructure or elements

of the environment (like shops, car park for instance), we turn to multiagent systems, firstly to propose a simulator of a traffic network, and a prototype of autonomous vehicle management.

Communication of knowledge implies classically to take care of the confidence about this knowledge. If β_a is the set of knowledge of the agent a. A received knowledge b can be :

- out-of-date : dynamic environment implies to date the knowledge.
- from a doubtful origin : the knowledge comes from an unknown agent, or without the required signature.
- inconsistent : the receiver of the knowledge b is unable to store it in its own knowledge without getting an inconsistency.

A road network is, as its name suggests it, a network composed of vertices and edges. In the research field turned to ad-hoc networks [12] the vertices are called 'node' and represent mainly the intersections, the edges represent the roads.

Knowledge sharing between mobile agents along a network can be done : directly, by messages exchange, when agent are physically close enough to communicate; or indirectly through the environment (generally at the nodes that represent intersections some electronic devices allow the mobile agents to store/read information).

Most of the works relative to traffic road management propose to control the flow of vehicles at the level of the intersections/nodes/vertices (see for example [1], [2]). Our objectives is to give the largest autonomy to the vehicles, so we made the choice to give to the node/intersection managers the responsability of the tactical behaviour of the autonomous vehicles that have to pass a crossroad. This differs with the other approaches that give to the intersection managers the control of the vehicles at the operational level. So we propose an architecture where non mobile agents are located at the nodes and communicate with mobile agents are close enough to receive messages.

Moreover, we can easily argue that to give the possibility to a node to interact with one another could give efficiency to the propagation of an information along a network.

Multilevel architecture. In order to allow the local and distant communications between the different type of agents (mobile-agent and node-agent), we use a multi-level architecture, inspired from holonic principles. We have already used this kind of pyramidal architecture for the simulation of a flexible assembly cell, in order to correct myopic behaviour of mobile and autonomous shuttle [13], and in a first scenario of knowledge propagation [14].

At the bottom of the system (level 0), we have the mobile agents. At the level 1, there are the node agents, that can schedule, manage conflicts between 'mobile agents' that have to cross them. These agents (mobiles and nodes) and the environment (the network) are included in the 'network agent' that represents the multiagent system.
Each agent can interact with all the agents that are in its 'vision field'.

Elements of volatile knowledge model.

A knowledge is a partial view of the environment or of the other agents, namely for a given object o of the environment (the traffic network for example); it is (generally) an incomplete copy of it, so a representation of o with missing attributes and methods.
If $\mathcal{O}$ is the set of objects of the environment, an object o is classically defined by attributes and function:

$$o = (id, attributes_o = \{a_1^o, \ldots, a_n^o\}_{n>0}, functions_o = \{f_1^o, \ldots, f_m^o\}_{m\geq 0}) \quad (1)$$

with $f_j^o : \mathcal{O} \mapsto \mathcal{O}$.

When an agent a perceives an object o, then o'_a is a partial representation of it at the instant of the perception, the cardinalities of the set of attributes and functions of o' are lesser than those of the object o:
$\mid attributes_{o'} \mid \leq \mid attributes_o \mid$ and $\mid functions_{o'} \mid \leq \mid functions_o \mid$

We define a knowledge κ_o^a (cf. def. 2) on an object o for an agent a by: o'_a, a partial view of o from a; $date_{\kappa_o}$, the date when the knowledge has been created or updated (by a or by another agent if the knowledge has been received); $builderAgent_{\kappa_o}$, the 'builder' of the knowledge (name of the agent that has created/updated the knowledge from its perception); $senderAgent_{\kappa_o}$, the 'sender' of the knowledge (name of the agent that could have sent the knowledge to a); $conf_{\kappa_o} \in [0,1]$ the confidence that a has on κ_o^a; $deg_{\kappa_o} \in [0,1]$ the percentage of confidence degradation applied each 'step'; $threshold_{\kappa_o} \in [0,1[$ the threshold under which the knowledge is no more considered (and has to be removed); $shareable_{\kappa_o^a}$, the fact that the knowledge is shareable or not by a.

$$\kappa_o^a = \begin{pmatrix} o'_a, date_{\kappa_o}, builderAgent_{\kappa_o}, senderAgent_{\kappa_o}, \\ conf_{\kappa_o}, deg_{\kappa_o}, threshold_{\kappa_o}, shareable_{\kappa_o^a} \end{pmatrix} \quad (2)$$

In the context of road traffic management, the traffic network is represented by a weighted graph, the knowledge, in our model, for this application, concerns the roads speed limit (the weight of the edge): each driver agent stores, updates the speed limit allowed and really practical on a road when it passes this road (in fact, the speed limit is updated in the knowledge if it differs of at least 20% (value arbitrary chosen) of the speed limit already stored).

Volatility. A knowledge, especially regarding the state of a traffic network is often out-of-date; it is necessary to allow an automatic update, a cleaning of the outdated or invalidated beliefs.

In our model, at each step, each passage in the life cycle of an agent (perception-cognition-action) or at each 'tick' given by a simulator, the confidence on a knowledge is degraded : $conf_{\kappa_o} \leftarrow conf_{\kappa_o} \times (1 - deg_{\kappa_o})$
A knowledge κ_o can be perennial ($deg_{\kappa_o} = 0$) or ephemeral ($deg_{\kappa_o} = 1$).

The degradation coefficient on a knowledge κ_o^a is given by the agent a itself in order to represent different behaviours of human driver: an 'expert' or a 'suspicious' can put a important value to knowledge sent by other agents; conversely

a 'novice' or a 'confident' agent would believe 'for a moment' to received knowledge. Agents are also allowed to decrease the confidence stored in a received knowledge.

When the confidence on a knowledge goes under the threshold, the knowledge is removed from the list of current knowledge (named *knowledgeList*) and is put in a list of 'doubtful' knowledge (named *knowledgeToCheck*).

A doubtful knowledge is restored and put back in the *knowledgeList* if the agent perceived directly that its information is correct, or if it receives a more recent version of the same information by another agent b ($\kappa_o^b.date_{\kappa_o} > \kappa_o^a.date_{\kappa_o}$).

Acquisition. Initially, all the information owned by the agent are defined as perennial (a driver agent starts with the road map, knowing the speed limits of the road; i.e. the weight of the edges).

When an agent perceives or receives a new value about an object already stored in the knowledge (for example, the speed limit of a road differs between the perception and the knowledge), a new non perennial knowledge is added to the list.

If the information, that has been perceived by the agent it-self, is the same as the one stored in a perennial knowledge (the situation returns to a 'normal state'), the time of this knowledge is updated, and all the non perennial knowledge on the same information/object are removed.

Of course, if the information, perceived or received, is the same as the one stored in a non perennial knowledge, the time and the confidence of this knowledge is updated (if the information is more recent).

So an agent can possess different knowledge about the same object with different level of confidence.

Knowledge management. It occurs that different knowledge about a same object co-exist in the belief of an agent a. To choose which knowledge, which value of an object, that will be taken into account to evaluate a strategy, the agent a uses a fitness proportionate selection (i.e. roulette wheel selection) with the following distribution :

- let $\kappa_{o_1}^a, \kappa_{o_2}^a, \ldots, \kappa_{o_n}^a$ a list of knowledge on an object o,
  with $\kappa_{o_1}^a.deg_{\kappa_o} = 0$ ($\kappa_{o_1}^a$ is perennial)
- let x = a number randomly chosen in $[0, L]$ with
  $L = (\sum_{i=2}^{n} \kappa_{o_i}^a.conf_{\kappa_o}) + (1 - \max(\kappa_{o_i}^a.conf_{\kappa_o})_{i=2}^{n})$
  x is situated in an interval that defines the knowledge to select.
- the interval corresponding to the knowledge $\kappa_{o_j}^a$ with $j > 1$ is :
  $\kappa_{o_j}^a \leftarrow \left[\sum_{i=2}^{j-1} \kappa_{o_i}^a.conf_{\kappa_o}, \sum_{i=2}^{j} \kappa_{o_i}^a.conf_{\kappa_o}\right[$
- the interval corresponding to the knowledge $\kappa_{o_1}^a$ is :
  $\kappa_{o_1}^a \leftarrow \left[\sum_{i=2}^{n} \kappa_{o_i}^a.conf_{\kappa_o}, L\right[$

Example with traffic management.
For example, for a given road r_1, the following knowledge about the weight of the road (the speed limit) are stored by the agent a :
$\kappa^a_{r_{1_1}}(time = 0, weight = 50, conf = 1)$, $\kappa^a_{r_{1_2}}(time = 30, weight = 20, conf = 0.4)$, $\kappa^a_{r_{1_3}}(time = 80, weight = 40, conf = 0.9)$
To decide which value to use to compute the possible best path, the agent chooses a number x in $[0, 1.4]$,
- if $x \in [0, 0.4[$ then the value of the knowledge $\kappa^a_{r_{1_2}}$ is used;
- if $x \in [0.4, 1.3[$ then the value of the knowledge $\kappa^a_{r_{1_3}}$ is used;
- if $x \in [1.3, 1.4[$ then the value of the knowledge $\kappa^a_{r1_1}$ is used.
So, the agent beliefs primarly that the speed limit on r_1 is $40km/h$, it knows there is a possibility that the road should be limited to $20km/h$ (because it has been limited so in a near past), and there is a little possibility that the speed limit is 'normal' and equal to $50km/h$

This principle of knowledge management is interesting in dynamic environment, where a lot of modifications occur; it allows an agent to take into account the observation, and to be able to check if the information are still valid or not. *In the previous example, if 'a' perceives next that the speed limit of the road is* $20km/h$*, the confidence of* $\kappa^a_{r1_2}$ *is set to 1, and its date is updated.*

Adjustment of the degradation coefficient. The model of knowledge allows to adjust, more than learn, the value of the degradation coefficient.

Indeed, the more some knowledge about a same object o are added to the list, or the more there is a change of the most confident knowledge, the more the degradation coefficient of the knowledge about o increases. An agent can not be trustful at longterm on information about an object if it knows that the object changes often its value. Of course, an agent could try to observe a new precise cycle, for example in the context of traffic management, a cycle depending of the time window (morning, ...), of the day, or on the weather, but these informations are already known initially by the agent, they come from data given by the appropriate organism. The objective of our proposition is to enable agents, that are planned to be included inside small devices (like smartphone, eeepc) to react rapidly to unexpected events.

The notion of 'doubtful' knowledge allows to adjust the degradation coefficient of a knowledge : this coefficient decreases each time a knowledge is restored from the *knowledgeToCheck* list (list of 'doubtful' knowledge) to the *knowledgeList* (list of 'trustfull' knowledge). Indeed, the more a value of an object stays identical, the more the associated knowledge must stay trustworthy.

Communication. In the context of resource bounded agent, perception (or knowledge acquisition) is examined by Chrisman [15] and Kinny [16]. The agent must decide *what to perceive* and *when to perceive* because of its limited capacity which does not allow to take into account all details in the environment. However, perception is the receiver part of the communication. On the other

hand, sender agent must equally decide *what message to send* and *when to send the message*. This approach aims also at decreasing the volume of messages, in avoiding unnecessary exchange of information.

An agent sends a knowledge to all the agents in its 'zone' (local area) only if the knowledge brings a new information about an object (it is a non perennial knowledge), or if the knowledge is a perennial knowledge updated. So if an agent perceives a normal state of a perennial knowledge, it does not inform the other agents.

Two parameters have to be arbitrary chosen according to the application context : δo, the degree from which the value of an object differs enough to be considered as new (it is fixed at 20% of the speed limit in our example); and the *dimension* of the 'local area' that includes the agents that should be interested by the knowledge.

When an agent a_2 comes in zone z_1 from another zone z_2, it interacts only with the first agent met to update its knowledge about the zone z_1.

3 Implementation of urban traffic

3.1 Simulation of urban traffic

We illustrate our proposal on a traffic road simulator that we have developed in the context of a project (Plaiimob : a simulating Plateform dedicated to Mobility services) of CISIT (for International Campus on Security and Inter modality in Transports).
The aim of this project is to allow vehicle-to-vehicle (V2V) communication to allow human or software drivers to automatically exchange data about their environment (incident/traffic jam on a road, information about off-street parking ...) [17].

We developed a traffic road simulator [13] with the Jade Platform[1] :

- The *environment* is a traffic network in the OpenStreetMap format[2] (OSM). This allows us to use true traffic networks, or to define our own maps in order to test particular situations.
- The classes of *agents* are: the AgentDriver agent class (it allows to simulate the behaviour of a driver), the AgentCrossroad agent class (it allows to manage the priorities at a crossroad according to the road signs), the ObserverAgent class (that allows to draw statistics from the simulating exercise).
- The *roles* played by the PersonnageAI are: RandomBehaviourRole: to represent a driver that moves randomly on the traffic; BusRole: to represent a driver that starts from a particular point and has to reach objectives, by linking some bus stops, at earliest.

[1] See the web site of the Jade platform : `http://jade.tilab.com/`
[2] See the web site of the OpenStreetMap project: `http://www.openstreetmap.org/`

Jade has been chosen because the aim of the project is to test our proposal on a real case, with agents embedded in the smart-phones of the drivers, and in core of autonomous vehicle. So we reuse some classes of the simulation to build agents used in the prototype.

Scenario: travels in Toulouse.

In order to present the benefit to have a V2V communication, we show here a small case study that includes 1 bus lines in a small area of the Toulouse French city (cf. Figure 1a); the map is easily made from the data obtained from the JOSM software.

The buses start from the 'Saint-Michel' bus stop and have to reach the bus stop close to a museum, at the end of the street 'rue des Pyrénées'. This part of the city have been chosen because the traffic network represents a relative grid that facilitate the tests.

Forty buses start from 'Saint-Michel', separated by 15 *ticks*, and, if there is no perturbation, each of them reaches the museum in 28.46 *ticks* (red line in Figure 1a). In a second simulation, a perturbation is placed on the street 'rue des

Fig. 1. Map of the scenario and best path.

Pyrénées', the time spent to travel the road is multiplied by 3 (due for examples to some roadworks, or manifestation to see a V.I.P., accident, etc.). The first bus discovers the problem once on the street and reaches the bus stop near the museum after 44.22 *ticks*, it adds this knowledge and sends it to the other buses. The next buses will take this information into account and compute a new best path, taking a detour on the 'rue Demouilles' street and arriving each at the museum after 32.95 *ticks* (yellow line in Figure 1b).
In a third simulation, a new perturbation occurs on the second street, the time to travel In fact, as the information confidence degraded itself to 0.9, generally the 10th bus takes again the 'rue des Pyrénées' road and updates this information and sends it to the others buses. An average of 4 buses test this street during this simulation. this street is doubled, the first bus arrives at the museum after 44.22 *ticks*, the second after 40.66 *ticks*, it adds this knowledge to its list and informs the other buses. The third bus arrives at the museum after 36.62 *ticks*, taking a detour via the 'rue Picot' street (blue line on the Figure 1c). From the buses that decides to not take the 'rue des Pyrénées' road, as in the second simulation, an average of 4 buses test the 'rue Demouilles' street during this simulation.

So, when the street come back to its normal state, it takes a average of 4×15 *ticks* to be perceived by one bus, and to be communicated to the other buses that will take the street, computing their best paths.

3.2 Prototype of autonomous vehicles

In [18], the author criticizes AI researches, and he underlines the importance of working in a real environment in order to take into account the sensory-motor aspects of intelligence. Following this criticism, in the second phase, we apply our communication model to wifibots (cf. Figure 4)[3], that are mobile robots, with 4 wheels, a camera, two InfraRed sensors, and embedding a light OS (Window Embedded in our case). This second phase allows us to examine the effect of concrete environment and identify new problems due to really distributed nature (in terms of computational resource) of wifibot agents.

The conceptual model of wifibot agents has been done with respect to Strategical-Tactical-Operational levels distinction.

In this model, the operational level takes the responsibility of commanding the robot engines like forward, backward, go left, go right, etc. After sending a command, the engine gives a response including the information of current speed, voltage, odometric information, infrared sensors informations (IR sensors), etc. Infrared information is used to detect a blocked way (an incident in the former scenario). By a bottom-up manner (from environment to agent), these sensors help to be aware of an obstacle, from operational level to the tactical and strategical levels. Odometric values are used to control the robot's position, while it is moving or turning..

The tactical level accomplishes the composite behavior like intersection crossing, U-turn, etc. These composite behaviors are the combination of operational

[3] See the web site relative to the WifiBot : http://www.wifibot.com/

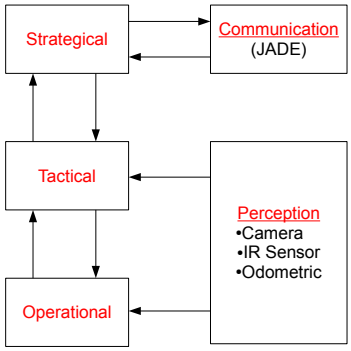

Fig. 2. Model of wifibot agents

behaviors because this level needs to identify road and intersection. Our wifibot is equipped by a camera and perception module relied to tactical level that processes images taken by camera in order to identify the behavioral triggers (e.g. an identified intersection triggers intersection crossing behavior).

The strategical level executes itinerary control and change. Agents have a destination point and a best itinerary to reach this destination point. In case of an unpredictable event like an incident or a blocked way somehow, agents deliberate for finding an alternative itinerary. Our communication model is applied at this stage, and allows to exchange among agents the information about the incident.

Scenario: communication between robot. In this scenario, illustrated Figure 3, robots have to make some loops between the 'start' and 'objective' points, with a shortest path [4].

Initially the robots make loops through the nodes $1, 2, 5, 6$. Some obstacles are added manually. In the example proposed Figure 3, the robot at the node 1 detects the obstacle on the edge $2-5$ and makes a U-turn to take the edge $3-4$. The second robot, informed by the first robot, decides also to use the same path $1-3-4-6$. When the first robot detects the obstacle on the edge $3-4$, it sends the information to the second robot; and stays on this edge, because it knows that the other robot will test the edge $2-5$. If the obstacles are not removed, the robots stay on their respective edge until one of the obstacle is removed. Then, the released robot informs the other of the opening of the edge and continues its tasks. The other robot make a u-turn to take the released edge.
If this one is the edge $3-4$, the two robots make the 'big loops' $1-3-4-6$ until one robot degrades enough of the confidence the edge $2-5$ to remove it and test this edge.

[4] A video can be read here : `http://www.youtube.com/watch?v=ifdCp76BKnM`

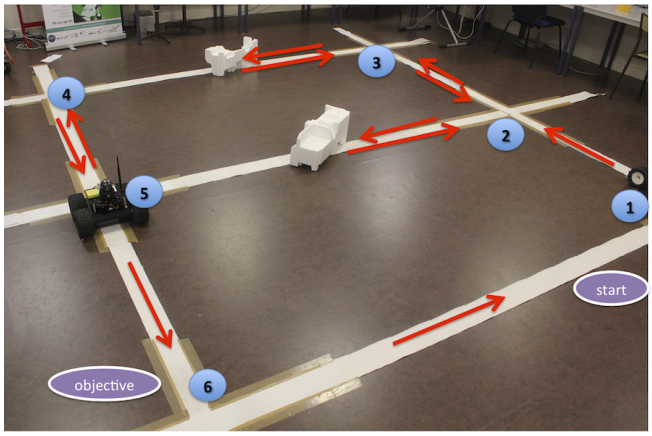

Fig. 3. Scenario for the WifiBots

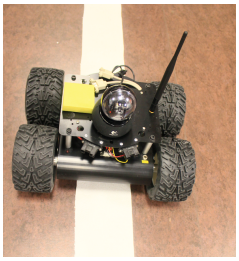

Fig. 4. WifiBot

Other scenarios have been tested, with one[5], two and three robots to validate our model of knowledge.

4 Conclusion

In order to allow the propagation of knowledge between mobile agents in a dynamic environment, a model of knowledge and a model of communication have been presented.

This model, simple, allows to represent different behaviours of drivers and allow to be easily embedded in light devices. It allows also to propagate knowledge about perturbations, and return to normal situation, in a distributed way without coordinator.

In the results presented in this paper, we make the assumption that all the agents are cooperative, and no defective; that is to say that they cannot send wrong knowledge, voluntary, or not (if a sensor has a dysfunction).

The trust on a knowledge depends only on the date from which the knowledge has been updated or created. We plan to introduce the notion of trust that depends of the sender; for example, if an agent a_1 perceives that an object has a different value than those in which it were confident at 90%, it increases the suspicious potential of the agent that sent it this wrong knowledge. When the suspicious potential of an agent reaches a threshold, this agent is considered as suspect and its communication are no more considered, or with a low level of confidence. This information is communicated to other agents, these ones can decide to put in quarantine the faulty agent, like in [19] for example.

[5] You can see a video here : `http://www.youtube.com/watch?v=gMiYeEH_KLU`

Acknowledgments. *This research was financed by the International Campus on Safety and Inter-modality in Transportation, the Nord/Pas-de-Calais Region, the European Community, the French Regional Delegation for Research and Technology, and the French National Centre for Scientific Research. We are grateful for the support of these institutions.*

References

1. Bazzan, A.L.: A distributed approach for coordination of traffic signal agents. Autonomous Agents and Multi-Agent Systems **10** (January 2005) 131–164
2. Dresner, K., Stone, P.: Mitigating catastrophic failure at intersections of autonomous vehicles. In: AAMAS Workshop on Agents in Traffic and Transportation, Estoril, Portugal (May 2008) 78–85
3. Ruskin, H.J., Wang, R.: Modeling traffic flow at an urban unsignalized intersection. In: Proceedings of the International Conference on Computational Science-Part I. ICCS '02, London, UK, UK, Springer-Verlag (2002) 381–390
4. Mandiau, R., Champion, A., Auberlet, J.M., Espié, S., Kolski, C.: Behaviour based on decision matrices for a coordination between agents in a urban traffic simulation. Appl. Intell. **28**(2) (2008) 121–138
5. Ferber, J.: Multi-agent systems - an introduction to distributed artificial intelligence. Addison-Wesley-Longman (1999)
6. Jennings, N.R., Sycara, K., Wooldridge, M.: A roadmap of agent research and development. Int. Journal of Autonomous Agents and Multi-Agent Systems **1**(1) (1998) 7–38
7. Macal, C.M., North, M.J.: Agent-based modeling and simulation: ABMS examples. In: Proceedings of the 40th Conference on Winter Simulation. WSC '08, Winter Simulation Conference (2008) 101–112
8. Klügl, F.: Agent-based simulation engineering. Habilitation thesis (July 2009)
9. Allen, T.M., Lunenfeld, H., Alexander, G.J.: Driver information needs. Highway Research Board **36** (1971) 102–115
10. Michon, J.A.: A critical view of driver behavior models: what do we know, what should we do? Springer (1986)
11. Delot, T., Ilarri, S., Cenerario, N., Hien, T.: Event sharing in vehicular networks using geographic vectors and maps. Mob. Inf. Syst. **7**(1) (2011) 21–44
12. Lochert, C., Hartenstein, H., Tian, J., Fussler, H., Hermann, D., Mauve, M.: A routing strategy for vehicular ad hoc networks in city environments. In: Intelligent Vehicles Symposium, 2003. Proceedings. IEEE. (2003) 156–161
13. Adam, E., Zambrano, G., Pach, C., Berger, T., Trentesaux, D.: Myopic behaviour in holonic multiagent systems for distributed control of fms. In Corchado, J., Pérez, J., Hallenborg, K., Golinska, P., Corchuelo, R., eds.: Trends in Practical Applications of Agents and Multiagent Systems. Volume 90 of Advances in Intelligent and Soft Computing. Springer Berlin / Heidelberg (2011) 91–98
14. Adam, E., Mandiau, R., Grislin, E.: Multilevel mas architecture for vehicles knowledge propagating. In Pérez, J.B., Sánchez, M.A., Mathieu, P., Rodríguez, J.M.C., Adam, E., Ortega, A., Moreno, M.N., Navarro, E., Hirsch, B., Lopes-Cardoso, H., Julián, V., eds.: Highlights on Practical Applications of Agents and Multi-Agent Systems. Volume 156 of Advances in Intelligent and Soft Computing. Springer Berlin Heidelberg (2012) 1–8

15. Chrisman, L., Simmons, R.: Sensible planning: focusing perceptual attention. In: Proceedings of the ninth National conference on Artificial intelligence - Volume 2. AAAI'91, AAAI Press (1991) 756–761
16. Kinny, D., Georgeff, M., Hendler, J.: Experiments in optimal sensing for situated agents. In: The second Pacific Rim International Conference on Artificial Intelligence, PRICAI. (1992)
17. Popovici, D., Desertot, M., Lecomte, S., Peon, N.: Context-aware transportation services (cats) framework for mobile environments. IJNGC **2**(1) (2011)
18. Brooks, R.: Intelligence without representation. Artificial Intelligence **47** (1991) 139–159
19. Vercouter, L., Jamont, J.P.: Lightweight trusted routing for wireless sensor networks. In Demazeau, Y., Pechoucek, M., Corchado, J.M., Pérez, J.B., eds.: PAAMS. Volume 88 of Advances in Intelligent and Soft Computing., Springer (2011) 87–96