Delft University of Technology

Delft Center for Systems and Control

Technical report 05-018

A test bed for multi-agent control systems in road traffic management*

R.T. van Katwijk, P. van Koningsbruggen, B. De Schutter, and J. Hellendoorn

If you want to cite this report, please use the following reference instead:
R.T. van Katwijk, P. van Koningsbruggen, B. De Schutter, and J. Hellendoorn, "A test bed for multi-agent control systems in road traffic management," in *Applications of Agent Technology in Traffic and Transportation* (F. Klügl, A. Bazzan, and S. Ossowski, eds.), *Whitestein Series in Software Agent Technologies*, Basel, Switzerland: Birkhäuser Verlag, ISBN 3-7643-7258-3, pp. 113–131, 2005. doi:10.1007/3-7643-7363-6_8

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

A Test Bed For Multi-Agent Control Systems In Road Traffic Management

R.T. van Katwijk, P. van Koningsbruggen, B. De Schutter, and J. Hellendoorn

Abstract. In this paper we present a test bed for multi-agent control systems in road traffic management. In literature no consensus exists about the best configuration of the traffic managing multi-agent system and how the activities of the agents that comprise the multi-agent system should be coordinated. The system should be capable of managing different levels of complexity, a diversity of policy goals, and different forms of traffic problems. The test bed aids in-depth research in this field, which we demonstrate by means of two example scenarios we have implemented.

1. Introduction

Throughout the world, traffic congestion forms a daily recurring problem. In many countries, more and more instruments are deployed to stimulate a smooth and safe flow of traffic on highways and urban road networks. Examples are ramp metering installations, which are used to regulate the inflow of traffic at on-ramps, variable message signs, which are amongst others used to inform road users about the current road conditions ahead or to present them with the current speed limit, and traffic signal installations, used to control traffic at intersections. Research is being conducted in the field of automatic coordination of these dynamic traffic management instruments. In [3, 6, 7, 11, 12, 14, 22, 23] it is argued that the communication capabilities of multi-agent systems can be used to accomplish this coordination.

No consensus exists about the best configuration of the traffic managing multi-agent system. The system should be capable of managing different levels of complexity, a diversity of policy goals, and different forms of traffic problems.

To be able to experiment with different strategies for the application of multi-agent systems for dynamic traffic management and to examine their applicability we need a test bed. Such a test bed facilitates the development of multi-agent

Research funded by the TNO spearhead program “Sustainable Mobility Intelligent Transport Systems (SUMMITS)” and the TU Delft spearhead program “Mobility of People and Transportation of Goods (TRM)”.

systems for dynamic traffic management. The main requirements of the test bed are, that

1. The multi-agent traffic management system can be configured easily.
2. The business logic of traffic engineers can be easily implemented, if possible by the traffic engineers themselves.
3. The multi-agent traffic management system can be evaluated in a realistic simulated traffic environment.
4. The multi-agent traffic management system can be easily transfered to a real-world application

This paper presents a test bed that satisfies the above requirements, and is organized as follows. In Section 2 an overview is given of current, different approaches to decentralized traffic control. The test bed is described in Section 3. Two example scenarios are also described in Section 4, which are meant to demonstrate the flexibility of the test bed in configuring a multi-agent system. We conclude this paper in Section 5 with our future research.

For clarity, a distinction can be made between vehicle-oriented and measure-oriented traffic control:

1. *Vehicle-oriented traffic control* focuses on controlling traffic through vehicle-based (internal) signals, like advanced driver assistance systems.
2. *Road-oriented traffic control* focuses on controlling traffic through roadside-based (external) measures at fixed locations, like traffic signals and variable message signs. This paper focuses on this form of control.

2. Decentralized Traffic Control Concepts

In literature many examples exist where the answer to the dynamic traffic control problem is sought in the form of a traffic control center that monitors the traffic network and performs a global, or network-wide, optimization to set up new parameters for its local controllers. Much of this past work has focused on centralized, and typically predictive, control. Although this approach is very appealing it just is not always possible to do this efficiently and effectively.

Therefore, a partial solution to network-wide traffic control is sought in problem distribution or decentralization. This section discusses the different approaches taken to traffic controller coordination in decentralized control in literature. We look at *hierarchical controller coordination*, *inter-controller coordination* and *intra-controller coordination.*

2.1. Hierarchical Controller Coordination

In order to maintain a network manager's overall control objective, given that part of this control is delegated to local controllers, many authors make use of a hierarchical structure in which higher-level agents are able to monitor lower level agents and are able to intervene when necessary. An example of such a structure is given in Fig. 1.

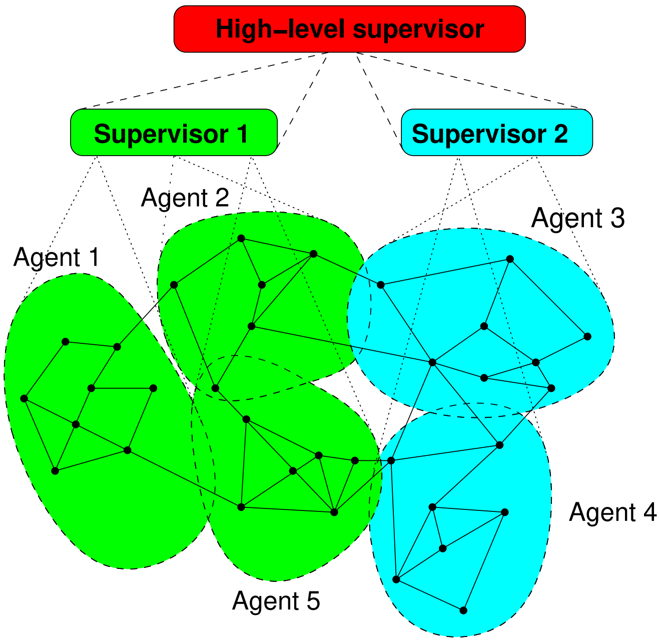


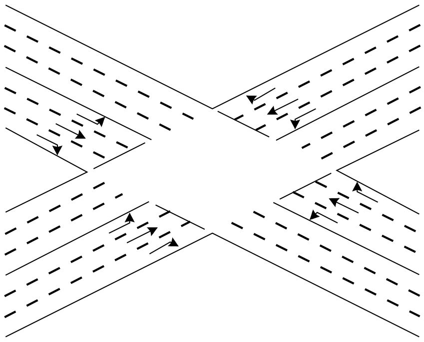
FIGURE 1. Schematic representation of hierarchical controller coordination

In the *TRYS* model developed by Hernández, Cuena and Molina [10] so-called '*problem areas*' are defined in a particular traffic situation. Each problem area has an agent assigned to it. The agents formulate actions to be performed and propose them to a 'coordinator', who makes a final decision in case of conflicting plans. Choy, Cheu, Srinivasan and Logi [3] introduce middle-level *zone controller agents* and highest-level *region controller agents* to coordinate the actions of the *intersection controller agents* that are present at the lowest level. In both approaches their is no communication between agents at the same level. In [3] France and Ghorbani use only two hierarchical levels, the lower level consisting of *local traffic agents* and a higher level consisting of *coordinator traffic agents*.

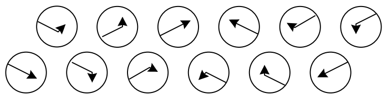
2.2. Inter-Controller Coordination

Many applications of intelligent agents in dynamic traffic control aim to make the local controllers more intelligent. The added intelligence aims to make the local controller more susceptible to the interest of the network as a whole. In principle, a local controller works on the basis of local information and can therefore perform only local optimization. In literature many different approaches are taken to overcome this shortcoming.

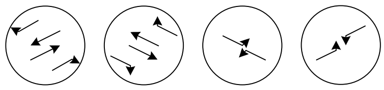
The most common approach is to share information among controllers. Another approach is to make the road-infrastructure responsible for controller-coordination. It is after all in the interest of the road segment that connects two



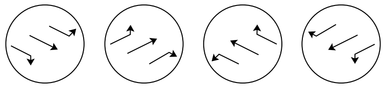
(a) The controlled intersection



(b) Each signal group is represented by an agent



(c) Each phase is represented by an agent



(d) Each approach arm is represented by an agent

FIGURE 2. Intra-controller coordination

intersections that the control actions at these intersections are coordinated. This is the approach we take in [22].

In most cases information is only shared upstream. In fewer cases [6, 7, 14] this information is also shared downstream. The sharing of information can be done on the level of:

- operational information (often raw detector data),
- tactical information (processed, derived data), and
- strategic information (expert advice).

2.3. Intra-Controller Coordination

The task of controlling a single, isolated intersection is often perceived to be one, undividable, centralized control problem, but numerous examples exist in which the control of a single intersection is stated as the result of a negotiation process between multiple intelligent agents, each having their own control objective. These agents can either represent the individual signal groups [11, 12], phases [23] or the arms of the intersection as shown in Fig. 2.

To date, no literature can be found that compares the merits and downsides of each of the chosen approaches to traffic controller coordination. The test bed will enable us to make this comparison.

3. Components of the Test Bed

In traffic control, two processes can be distinguished. First of all, there is the *traffic process.* This process can be observed by means of monitoring equipment (e.g.

induction loop detectors and floating-car data) and influenced by traffic control instruments (e.g. variable message signs, ramp metering installations, and traffic signals), which form the working material for the traffic control process. The second process, namely the *control process*, is, in the case of our test bed, comprised of multiple interacting intelligent agents.

This section discusses how the test bed is set up. The test bed consists of an *interaction model*, *intelligence models*, and a *world model*. These models are presented in the next subsections. For a more detailed discussion of the material, we refer to [19].

3.1. Interaction Model

The interaction model is used to model the interactions between the agents. All communications in our test bed conform to the specifications as set by the Foundation for Intelligent Physical Agents (FIPA) [8], an approach also taken in [2] for a video-based traffic monitoring system. The core message of FIPA is that through a combination of speech acts, predicate logic and public ontologies, standard ways can be offered of interpreting communication between agents in a way that respects the intended meaning of the communication. Ontologies provide the vocabulary for representing and communicating knowledge about a topic and a set of relationships and properties that hold for the entities denoted by that vocabulary.

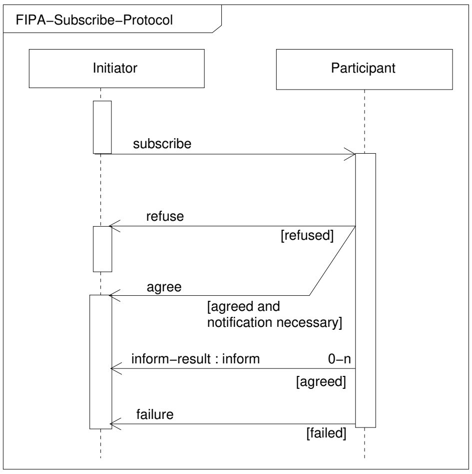
The FIPA standards require that each agent publishes the services it provides to a Directory Facilitator. This Directory Facilitator is a component of the multi-agent system that provides a yellow-pages directory service to agents. At least one directory facilitator must be present in the multi-agent system. The presence of a directory facilitator enables a dynamic configuration of the agent system. This way the location of a service an agent needs for its own operation does not have to be hard coded in the agent, but can be found at run-time though means of the Directory Facilitator.

FIPA's standard interaction protocols and communicative acts are currently sufficient for our purposes. Examples of these are the subscription interaction protocol (Fig. 3(a)), iterated contract net interaction protocol(Fig. 3(b)) (for negotiations), the propose interaction protocol and the request interaction protocol, all of which we need for our cooperating traffic agents. For this we rely on the JADE agent development environment [1].

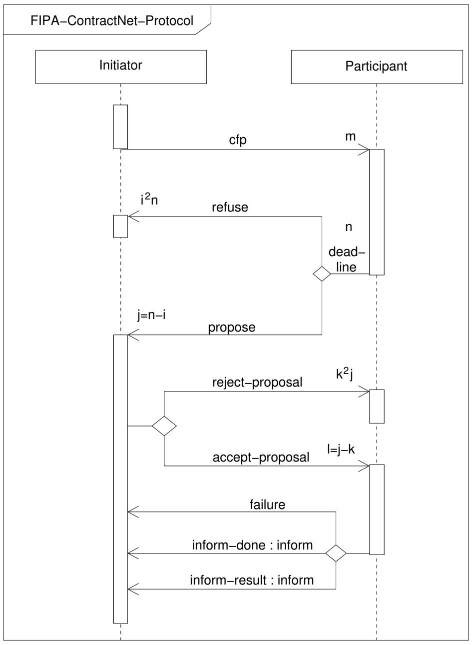
Currently there is no ontology available at FIPA that relates to the domain of traffic control. Our aim is to construct a broadly applicable ontology for traffic control through our use of the test bed for different cooperative traffic control problems.

3.2. Intelligence Model

The intelligence models are used to model an agent's intelligence. A fundamental decision in defining a problem is deciding how to model it. Sometimes experience is available to aid in choosing the best paradigm. Often a paradigm is



(a) FIPA subscription interaction protocol



(b) FIPA contract net interaction protocol

FIGURE 3. FIPA interaction protocols

selected on the basis of the applicant's familiarity with it. This is why conventional programming paradigms are often considered first. The test bed we have designed allows programming the intelligence model using a conventional programming paradigm using the C(++) and Java languages, but is not limited to these languages. The dynamic traffic management domain has always been open to unconventional approaches from the field of artificial intelligence, including evolutionary algorithms, knowledge-based systems, neural networks and multi-agent systems [5, 13, 15, 17, 18, 24].

Currently the test bed supports the following approaches:

1. *Rule-based inference*
   For our test bed we developed a generic rule-based agent using *JESS*, a rule-based reasoning engine [9]. Incoming messages are converted to facts and asserted into its working memory. Derived facts describing messages to be sent are translated into corresponding FIPA messages, after which JADE takes care of their delivery. The rule-based agent is typically used to program the expertise of a human expert and is as such an ideal prototyping and training tool for traffic managers [21]. Decision rules in an expert

Business rule
If there is a route named *?route* that has an alternative named *?route-alt* for which the quality of traffic flow is higher **Then** direct traffic from the former route to the latter route using the following message ....

JESS rule

```
(defrule take-action
  (RouteAlternative
     (route ?route)
     (alternative ?route_alt))
  (Route
     (name ?route)
     (quality ?quality))
  (Route
    (name ?route_alt)
    (quality ?quality_alt&:(> ?quality_alt ?quality))
=>
  (assert
  (VMSSignal
    (text "Congestion on route:     " ?route
          "Please take alternative: " ?route_alt)))
)
```

Table 1. Rule-based intelligence

system take the form of simple IF-THEN statements. A simplified example of an IF-THEN statements as used for network traffic management is given in Table 1.

2. *Bayesian inference*
   For our test bed we have developed a generic Bayes agent using *JavaBayes*, a set of tools for the creation and manipulation of Bayesian networks [4]. Bayesian inference is an approach to statistics in which all forms of uncertainty are expressed in terms of probability. Traffic control is guided by uncertainties, namely uncertainties regarding the current traffic state (due to limits in the amount and quality of the available monitoring data), uncertainties regarding the progression of the current traffic state (due to limits in traffic forecasting models), and uncertainties about the effects of a control action on traffic flow. Bayes' law describes how events from the past influence the probability of those events that are still about to

happen (or not). Because the theorem explicitly describes the relation between events, causality and their context, it is an excellent mathematical foundation for automated traffic control [20].

Incoming messages are converted and assigned to variables in the network. The set of variables that has assigned values is called evidence. The resulting expectations corresponding to messages to be sent are translated into corresponding FIPA messages, after which JADE takes care of their delivery.

3.3. Virtual World

We use the microscopic traffic simulation package *Paramics* developed by Quadstone [16] as our traffic model. Fig. 4 contains a screen-shot of the model. Paramics simulates traffic at the level of individual vehicles. Our prime motivation for choosing for Paramics for our test bed is that it can be programmatically extended through an application programming interface. The application programming interface allows the agent-controllers to retrieve information from and to enter control actions into the simulation environment. Paramics was one of the first models providing this capability. The test bed can however easily be modified to also use other traffic models that provide a suitable application programming interface. Traffic simulation models often employ a fixed-time step based method to simulation. Paramics is no exception. It is possible to retrieve detector data and modify the actuators in-between these time steps, which is shown schematically in Fig. 5. The agents in a multi-agent system in contrast operate in continuous time. In order to bridge this gap, the Paramics-World Interaction Agent stores the request and subscriptions from other agents until it is time to continue to the next time step. The decision to go to the next time step depends on the type of synchronization one wishes to apply. Since the traffic system is simulated using a single simulation process, there is only *one* agent that handles *all* outside world requests and subscriptions. In the real world each detector and actuator could in principle be represented in the multi-agent system by a specific agent. This is however a theoretical deployment scenario, which will be difficult to attain in practice. Traffic control centers are often equipped with different control applications, each representing a group of detectors or actuators from one manufacturer. A more realistic deployment scenario is that these applications are retrofitted with an agent wrapper.

Simulation is used to test various real-world application scenarios for multi-agent systems. In order to test whether the configured multi-agent system will function under real-time conditions synchronization can be performed by slowing down the simulation such that simulation time equals wall-clock time. However, the test bed will typically be used to test the performance of a configured multi-agent system with regard to traffic flow. In that case it is required that the multi-agent system gets sufficient time to formulate the control actions and that the simulations are repeatable.

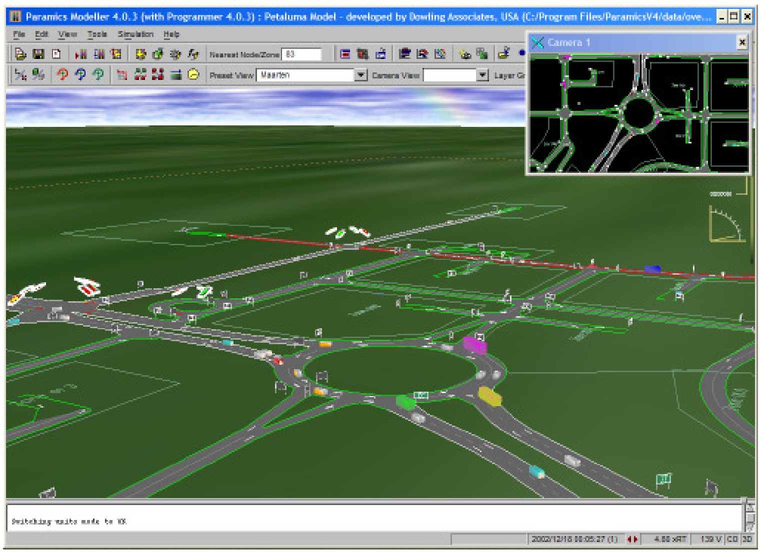


FIGURE 4. Screen-shot of Paramics

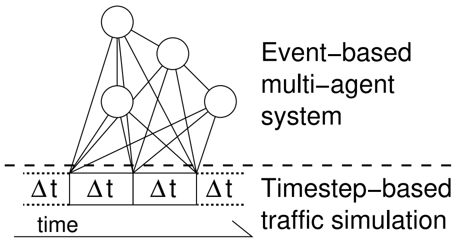


FIGURE 5. Interaction in discretized time

In order to guarantee that the multi-agent system is given sufficient time to formulate the control actions, it has to be determined when the agents in the multi-agent system have finished formulating their control actions. This is done using a special purpose agent, that when present, requires an agent to report when it wants to change its state from busy to idle. This special purpose agent is named MAI, short for Maintainer of Agent Information. When all agents have reported to be idle, and thus all information on the basis of which control actions can be formulated has been processed by the multi-agent system, is the simulation allowed to continue.

The FIPA propose interaction protocol is used to convey an intended state change from an agent to the maintainer of agent information. This protocol mandates that proposals are explicitly accepted or refused. In this case the latter will never be the case. The explicit acceptation is needed since there is no way to guarantee that messages from the agents informing about an intended state change arrive in the order they are sent. Without explicit acceptation the simulation can sometimes be allowed to continue before the agents are done formulating their control actions. An example of this is shown in Fig. 6(a). In this figure an initiator initiates a conversation (message 2) with a participant which has previously registered itself as being idle (message 1). In order to participate in the conversation, the participant has to register itself as being busy (message 3). If this message is delayed it is possible that the message arrives at the maintainer of agent information at a moment when the initiated conversation has already ended (message 4) and the initiator's has registered itself as being idle (message 5). In this case the simulation is allowed to continue while the participant might still be busy formulating its control actions. Fig. 6(b) shows the same communication trace where each intended state change is explicitly accepted. In this case it is guaranteed that the simulation continues only when all agents are done formulating their control actions.

When agents are mandated by the maintainer of agent information to communicate intended state changes, all agents operate following the higher level state chart as shown in Fig. 7. In this figure the busy state is a composite state which encompasses the regular state charts of the agent when it is operating in unsynchronized mode. When an agent intends to change state from idle to busy or vice versa it communicates this intent to the maintainer of agent information though means of a propose-message. When the propose-message is sent, the agent enters an intermediate pre-idle or pre-busy state. The agent remains in this intermediate state until it receives an accept-proposal message from the maintainer of agent information in reply to the proposed state change. Only then is the agent allowed to change its state to either the busy or the idle state.

4. Application Scenarios

From the perspective of the multi-agent system developer, the following three phases can be distinguished:

1. Configuration phase. The user creates the traffic situation to be simulated and the interacting multi-agent system.
2. Simulation phase. The simulation is started, simulation data is collected and control actions are derived and communicated by the multi-agent system.
3. Analysis phase. The simulation data is analyzed using the tools provided for this purpose by the traffic simulation model.

A special XML file lists the agents that make up the multi-agent system. Adding an agent is done by adding a line containing its name and the location of the agent-specific knowledge. The next step is to actually construct the JESS files containing the knowledge. When the simulation is started, the XML file is parsed, the listed agents are created in JADE, and they are provided with the corresponding agent-specific knowledge.

To demonstrate the flexibility of the test bed in configuring a multi-agent system, we implemented two alternative multi-agent system configurations for the two examples of [22] describing:

1. The coordination of traffic management instruments on a freeway corridor, and
2. The coordination of traffic management instruments in a network.

For both examples traffic control was at first delegated to agents representing each part of the road infrastructure. In this configuration information is exchanged between agents representing neighboring infrastructure elements. From a functional perspective it seems natural to have the agents directly represent the road infrastructure. However, from a more practical perspective it is more natural to have the agents represent the traffic control instruments present in the network, since the technical infrastructure for this kind of delegation is largely already there. We therefore subsequently converted the multi-agent traffic management system into a system in which traffic control was delegated to agents representing the traffic control instruments. In that case information is exchanged between traffic control instruments that are in relative close proximity of one another.

The directory facilitator in combination with the XML file allowed us to easily configure the infrastructure-based and controller-based agent implementations once the agent-intelligence was defined. The files containing the agents' knowledge were assigned to different agents depending on the configuration chosen. The agents' knowledge itself could remain unchanged. The ease with which the two agent configurations were configured, showed that the system enables users to create and experiment with different multi-agent control structures for dynamic traffic management. The two alternative multi-agent configurations for each of the two examples of [22] are described below.

4.1. Scenario 1: Coordination on a Freeway

To alleviate traffic congestion due to recurrent congestion, ramp metering has been applied throughout the world. Ramp metering aims to limit the number of vehicles entering the freeway so that freeway flow in the ramp metering installation's area of influence can be maintained at the desired quality level. Excess demand is forced to wait at the entrance ramp. The intention of ramp metering is, therefore, to maintain uninterrupted, non-congested flow on the freeway as long as possible by transferring delay from the freeway to the entrance ramp.

In Fig. 8(a) a corridor is represented. In this representation traffic drives from the left side of the picture to the right side. Traffic enters the freeway represented in the figure from the freeway upstream (left of the picture) and the three on-ramps (numbered 1, 2 and 3). Each on-ramp is metered by a ramp metering installation. Each ramp metering installation acts upon the information it receives from the measurement loops directly up- or downstream of the on-ramp (depending on the chosen implementation). As such, no matter what the objective of a ramp metering installation may be, it will always be local.

Coordinated ramp metering refers to the application of ramp control to a series of ramps where the interdependency of ramp operations is taken into account. The primary objective of integrated ramp control is to prevent or reduce the occurrence of congestion on a longer stretch of freeway. Therefore, the control of each ramp is based on the demand-capacity considerations for the whole stretch rather than on the demand-capacity constraint at each individual ramp. Whenever traffic demand on the freeway near on-ramp 3 approaches the capacity of the road in the bottleneck, the ramp-metering installation at on-ramp 3 will maximally delay traffic on the on-ramp. In case traffic demand on the freeway near on-ramp 3 amply exceeds the capacity of the road in the bottleneck then a queue will start forming. This queue will rapidly build up to on-ramp 2. As soon as this happens the metering installation will want to maximally delay traffic on its on-ramp. Too late, since the queue has already formed and has probably reached on-ramp 1. The situation illustrated above could have been delayed as long as possible or could have been prevented altogether when the actions of the ramp-metering installations would have been coordinated.

As soon as the ramp-metering installation at on-ramp 3 notices that the traffic demand on the freeway near on-ramp 3 is going to approach the capacity of the road in the bottleneck, it could have called upon the ramp-metering installation at on-ramp 2. The ramp-metering installation at on-ramp 3 can have a pretty good idea of how much traffic is going to arrive based upon the information it can get from the ramp-metering installations upstream and it's own experience. Instead of waiting for the problem to occur (the queue forming at the bottleneck) the problem can thus be pro-actively combated.Whenever a ramp-metering installation calls upon a ramp-metering installation upstream, the ramp-metering installation that is called upon can try to meet this request as good as it can. If it fails to fully meet the request it can then itself call upon a ramp-metering installation further upstream.

Although the example above is certainly an improvement compared to the example without co-ordination it still isn't exactly ideal. Traffic that wants to enter the freeway using on-ramp 3 for example is put at a disadvantage compared to the traffic that wants to enter the freeway using on-ramp 2. Traffic that wants to enter the freeway using an on-ramp further upstream is generally put at a disadvantage in this example.By allowing the individual ramp-metering installations to negotiate about how much traffic they plan to let through, they can also take care that the delay imposed at the traffic at the on-ramps is evenly distributed.

Using the test bed, we have developed a multi-agent system, capable of communicating about the required amount of metering. As a result of this communication, the ramp metering installations are better aware of future congestion. In this way, they can react in advance and solve the congestion in a more timely way, resulting in a 5 percent improvement in freeway travel time [19].

In the agent implementation of Fig. 8(b) traffic control is delegated to the infrastructure. Information is exchanged between neighboring infrastructure agents. Another, and more practical way of delegation is to delegate traffic control to the ramp-metering installations (the control instruments) as shown in Fig. 8(c). In that case information is exchanged between the ramp-metering controller agents. The arrows indicate the communication possibilities of each agent. In this way the knowledge about the current traffic situation propagates upstream through the network.

4.2. Scenario 2: Coordination on a Network

Variable Message Signs (VMS) are programmable traffic control devices that display messages composed of letters, symbols or both. They are used to provide information about changing conditions in order to improve operations, reduce accidents, and inform travelers. They may advise or urge drivers to change travel speed, change lanes, divert to a different route, or simply to be aware of a change in current or future traffic conditions.

A VMS system can directly benefit:

- Routing choice, saving vehicle traveled miles and hours, and
- Congestion reduction.

The objective of a VMS depends on the application, for this example we'll focus on the application of a VMS for route guidance.

In Fig. 9(a), a small freeway network is represented. The example given below focuses on the traffic that travels from the left side to the right side of the picture. The main arteries (the arteries denoted with the numbers 1, 2, 3 and 5) are drawn fatter than the regular arteries (artery number 4). Traffic enters the network by means of a main artery (1) and a regular artery (4). Main artery 1 branches off into main arteries 2 and 3. Road users are informed just before this branching point about the traffic conditions on both branches by means of a VMS. A ramp-metering installation meters the traffic that wants to enter the main artery (5) using the regular artery (4).

In case a queue starts forming caused on freeway 5 the ramp-metering installation will only start metering when the end of the queue approaches the junction of arteries 3,4, and 5. This will somewhat slow down the growth of the queue on main artery 3. The road user will be informed by the VMS at the junction of arteries 1, 2, and 3 about the worsening traffic conditions at main artery 3. This will slow down the growth of the queue even more. When the end of the queue reaches the junction of arteries 1,2, and 3, traffic will be faced with the consequences of the incident that took place on freeway 5 that never planned to take a route including freeway 5 in the first place. The queue will then start to grow rapidly since road users wanting to follow freeway 2 will be blocked passage by road users wanting to follow freeway 3.

Using the test bed, we have developed a multi-agent system, capable of coordinating the actions of the ramp-metering installation and the VMS. As a result of this coordination, we were able to further delay the situation illustrated above and even prevent the situation from occurring altogether. As soon as the density of traffic on the main artery near the ramp-metering installation approaches a level that entails that traffic entering the main artery from artery 4 will have to be significantly delayed, the ramp-metering installation calls upon the VMS to start rerouting traffic via freeway 2. By combining the efforts of both the ramp-metering installation and the VMS the growth of the queue can be delayed significantly. By delaying the growth of the queue it will take longer for the end of the queue to reach the junction of arteries 1,2, and 3. The queue might even dissolve before it reaches the junction. This way, road users that want to follow freeway 2 can do so as long as possible without being blocked passage.

In the agent implementation of Fig. 9(b) traffic control is delegated to the infrastructure. Information is exchanged between neighboring infrastructure agents. Another, and more practical way of delegation is to delegate traffic control to the traffic control instruments as shown in Fig. 9(c). In that case information is exchanged between the ramp-metering controller agents. The arrows indicate the communication possibilities of each agent. In this way the knowledge about the current traffic situation propagates upstream through the network.

5. Conclusions and Future Research

Our contribution to the research in the road-traffic management domain is the development of a software environment for rapid development of multi-agent control systems. In this paper a test bed for agent-based road traffic management is presented. The organization of the software is discussed, as well as the role of rule-based and Bayesian reasoning. The implementation of two multi-agent systems with the test bed is described. The implementations show that multi-agent systems can be created easily.

The presented test bed will be of great value for developments in traffic management. The developed system still has opportunities for further extension. A graphical user interface can be developed in which agents can be created and the multi-agent system can be configured with only a few mouse clicks. This would further accelerate the implementation of the desired multi-agent system. Extending the number of available intelligence models could be another improvement.

With the test bed, a tool has been developed to study the possibilities of applying multi-agent systems in dynamic traffic management. It is a starting point for our further research in decentralized traffic control. E.g. the different decentralized control concepts described in Section 2 can be implemented using the test bed. Another line of research regards the scalability properties of multi-agent based decentralized control versus centralized control.

References

[1] Fabio Bellifemine, Giovanni Caire, Tiziana Trucco, and Giovanni Rimassa, *Jade programmer's guide*, TILab S.p.A., July 2002, URL: http://sharon.cselt.it/projects/jade/.

[2] Luís Botelho, *A control structure for agent interaction*, Proc. IEEE Intelligent Vehicles Symposium (IVHS'00), October 2000, pp. 398–403.

[3] Min Chee Choy, Ruey Long Cheu, Dipti Srinivasan, and Filippo Logi, *Real-time coordinated signal control using agents with online reinforcement learning*, Proc. 83rd Annual Meeting of the Tranportation Research Board (TRB'03), November 2003.

[4] Fabio Gagliardi Cozman, *Bayesian networks in java*, URL: http://www-2.cs.cmu.edu/˜javabayes, May 2003.

[5] J. Cuena, J. Hernández, and M. Molina, *Knowledge-based models for adaptive traffic management systems*, Transportation Research Part C **3** (1995), no. 5, 311–337.

[6] Enrique D. Ferreira, Eswaran Subrahmanian, and Dietrich Manstetten, *Intelligent agents in decentralized traffic control*, Proc. IEEE Intelligent Transportation Systems (ITSC'01) (Oakland (CA), USA), IEEE, August 2001, pp. 707–711.

[7] Nicholas V. Findler, Sudeep Surender, and Ş. Catavra, *A semi-autonomous decentralized system for controlling street traffic signals*, Proc. IEEE Autonomous Decentralized Systems, April 1995, pp. 377–383.

[8] FIPA, *The foundation for intelligent physical agents*, URL: http://www.fipa.org, 2003.

[9] Ernest Friedman-Hill, *Jess, the expert system shell for the java platform*, URL: http://herzberg.ca.sandia.gov/jess/, May 2003.

[10] Josefa Hernández, José Cuena, and Martín Molina, *Real-time traffic management through knowledge-based models: The TRYS approach*, ERUDIT Tutorial on Intelligent Traffic Management Models (Helsinki, Finland) (ERUDIT, ed.), 1999.

[11] Sandy Irani and Vitus Leung, *Scheduling with conflicts, and applications to traffic signal control*, Proc. ACM-SIAM Symposium on Discrete Algorithms, January 1996, pp. 85–94.

[12] Iisaki Kosonen, *Multi-agent fuzzy signal control based on real-time simulation*, Transportation Research Part C **11** (2003), 389–403.

[13] B. Krause and C. von Altrock, *A complete fuzzy logic control approach for existing traffic control systems*, Mobility for Everyone, Proceedings of the 4th World Congress on Intelligent Transportation Systems (Berlin, Germany), October 1997, Paper no. 2045.

[14] Jia Lei and Ü. Özgümer, *Decentralized hybrid intersection control*, Proc. IEEE Decentralized Hybrid Intersection Control, vol. 2, December 2001, pp. 1237–1242.

[15] M. Molina, J. Hernández, and J. Cuena, *A structure of problem-solving methods for real-time decision support in traffic control*, International Journal of Human-Computer Studies **49** (1998), no. 4, 577–600.

[16] Quadstone Ltd., *Paramics: Microscopic traffic simulation*, URL: http://www.paramics-online.com/, May 2003.

[17] S.G. Ritchie, *A knowledge-based decision support architecture for advanced traffic management*, Transportation Research Part A **24** (1990), no. 1, 27–37.

[18] A.W. Sadek, M.J. Demetsky, and B.L. Smith, *Case-based reasoning for real-time traffic flow management*, Computer-Aided Civil and Infrastructure Engineering **14** (1999), no. 5, 347–356.

[19] Alexander Th. van den Bosch and Maarten R. Menken, *Multi-agentsystemen en verkeersbeheersing – De ontwikkeling en demonstratie van een testbedomgeving*, Tech. Report 03-7N-109, TNO, Delft, May 2003, MSC Thesis (in Dutch), Vrije Universiteit, Amsterdam, The Netherlands.

[20] Sicco Pier van Gosliga, Paul van Koningsbruggen, and Ronald van Katwijk, *Bundling the available knowledge on local traffic situations in network wide traffic management*, Submitted to the 11th World Congres on Intelligent Transportation Systems (ITS'04), 2004.

[21] Ronald van Katwijk, Paul van Koningsbruggen, and Marcel Westerman, *On the application of an expert system for network traffic management*, Submitted to the 11th World Congres on Intelligent Transportation Systems (ITS'04), 2004.

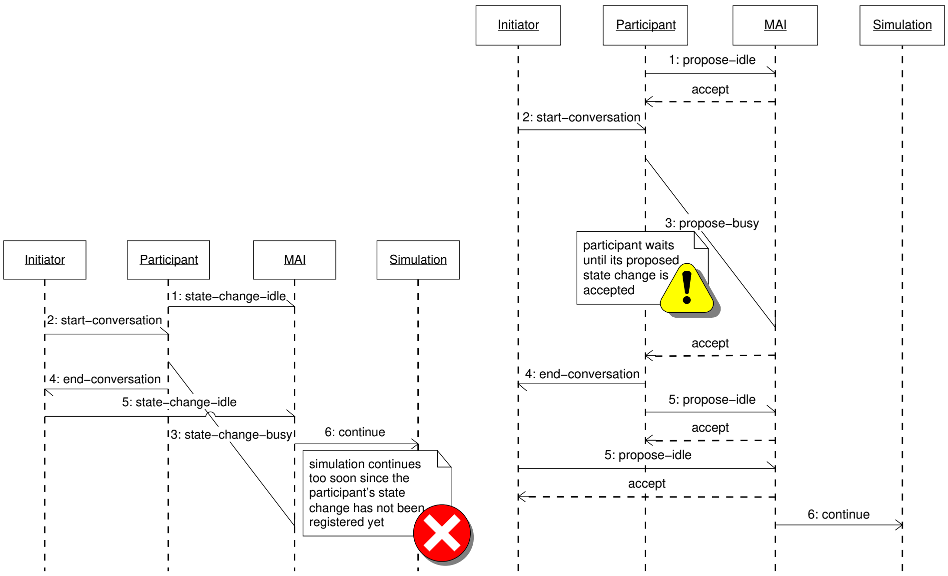
[22] Ronald T. van Katwijk and Paul van Koningsbruggen, *Coordination of traffic management instruments using agent technology*, Transportation Research Part C: Emerging Technologies **10** (2002), no. 5–6, 455–471.

[23] Liu Xiaoming and Gong Xiaoyan, *Study of intersection traffic flow control on the basis of agents*, Proc. IEEE Intelligent Transportation Systems (ITSC'03), vol. 2, October 2003, pp. 1755–1758.

[24] H. Zhang and S.G. Ritchie, *Real-time decision-support system for freeway management and control*, Journal of Computing in Civil Engineering **8** (1994), no. 1, 35–51.

Netherlands Organization for Applied Scientific Research (TNO), 2600 JA Delft, The Netherlands
E-mail address: {`r.vankatwijk, p.vankoningsbruggen`}`@inro.tno.nl`

Delft Center for Systems and Control (DCSC), Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands
E-mail address: {`r.vankatwijk, b.deschutter, j.hellendoorn`}`@dcsc.tudelft.nl`



(a) State changes are not confirmed: Unsuccessful synchronization (b) State changes are confirmed: Successful synchronization

Figure 6. Synchronization sequences

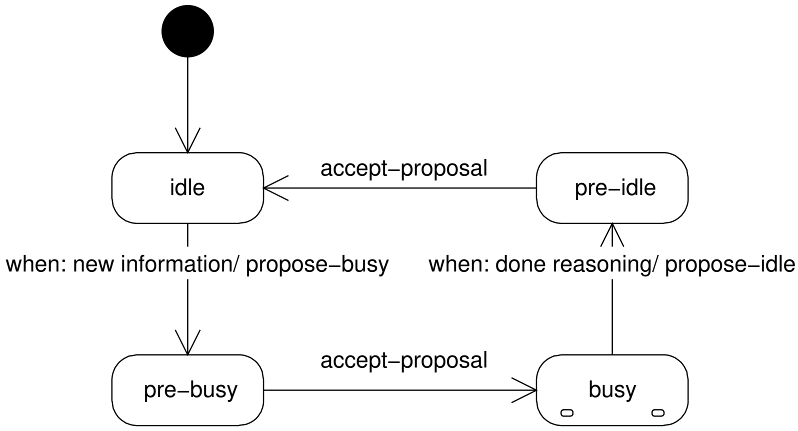
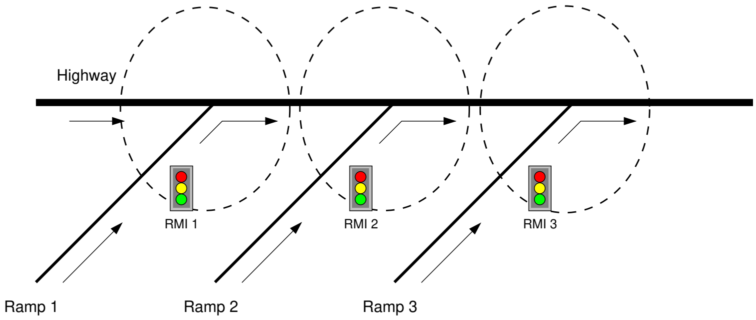
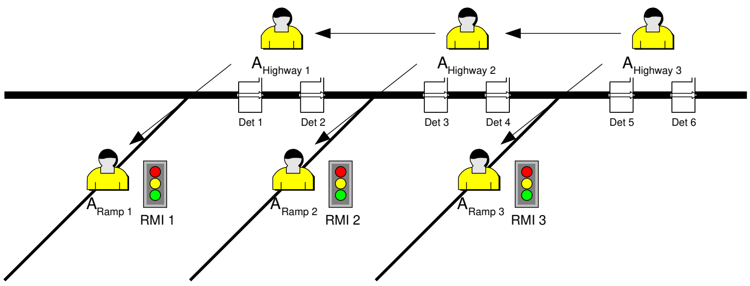


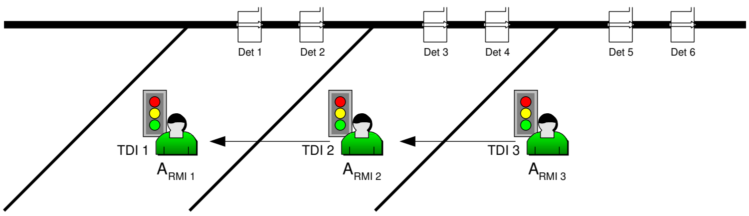
Figure 7. States and transitions needed to ensure synchronized operation



(a) Traffic network used in the scenario

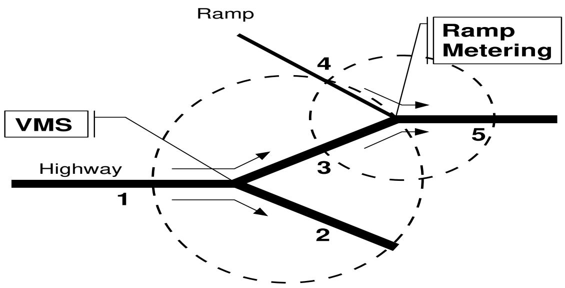


(b) Coordination through infrastructure agents

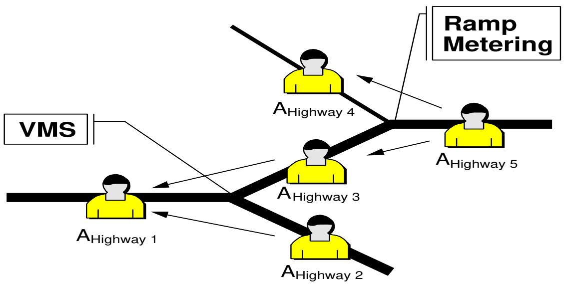


(c) Coordination through controller agents

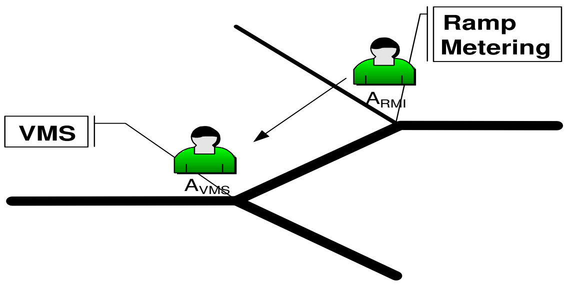
FIGURE 8. Different multi-agent control system structures



(a) Traffic network used in the scenario



(b) Coordination through infrastructure agents



(c) Coordination through controller agents

FIGURE 9. Different multi-agent control system structures