Technical report 05-007

Test bed for multiagent control systems in road traffic management*

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

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

* This report can also be downloaded via https://pub.bartdeschutter.org/abs/05_007.html

Test Bed for Multiagent Control Systems in Road Traffic Management

R.T. van Katwijk[*][†], P. van Koningsbruggen[*], B. De Schutter[†],
and J. Hellendoorn[†]

[*] Netherlands Organization for Applied Scientific Research (TNO)
P.O. Box 6041, 2600 JA Delft, The Netherlands
email: R.VanKatwijk@inro.tno.nl, P.VanKoningsbruggen@fel.tno.nl

[†] Delft Center for Systems and Control, Delft University of Technology
Mekelweg 2, 2628 CD Delft, The Netherlands
{r.t.vankatwijk,b.deschutter,j.hellendoorn}@dcsc.tudelft.nl

Abstract

In this paper we present a test bed for multiagent control systems in road traffic management. As the complexity of traffic control on a network grows it becomes more difficult to coordinate the actions of the large number of heterogeneous traffic management instruments that are available in the network. One way of handling this complexity is to divide the coordination problem into smaller coherent subproblems that can be solved with a minimum of interaction. Multiagent systems can aid in the distribution of the problem (over the various agents that comprise the multiagent system) and facilitate the coordination of the activities of these agents when required. In the literature no consensus exists about the best configuration of the traffic managing multiagent system and how the activities of the agents that comprise the multiagent system should be coordinated. The decomposition of a problem into various subproblems is an active field of research in the world of distributed artificial intelligence. This paper starts out with a survey of the approaches as they are reported in the literature. Subsequently the test bed is introduced and the modules it is comprised of. Finally an application is presented that illustrates some of the research the test bed has made possible.

1 Introduction

The current traffic situation results from the superposition of many events, both foreseen and unforeseen. In traffic management one therefore has to balance between control and contingency:

1. The control part deals with the redistribution of network traffic flows in order to prepare for foreseen traffic situations. The weather condition, morning and evening rush hour, planned road works, and bridge openings are events that can be foreseen and that can be prepared for. Given the foreseen available network capacity and traffic demand the road network can be tuned using the available traffic control instruments which can be instructed to influence the road-user's route preference, to adjust the current speed limit, to reduce the inflow of on-ramp traffic to the mainline, etc.

2. The contingency part deals with events that are difficult to foresee, such as minor accidents or incidents, a truck that is loading on the street, a vehicle that fails to start at a traffic light etc.

> It is intractable for a traffic operator to prepare for and respond to all these minor disturbances even if he is supported with advanced decision support systems. This is largely due to the complexity of the control instruments and the speed with which the operator should respond to these disturbances.

One way to more effectively handle contingencies is to make the traffic control instruments more intelligent and have them deal with the intricacies of configuring the traffic control instrument to the situation at hand. The traffic operator can then focus on the direction of traffic over the network, since a part of the problem is dealt with by the traffic control instruments. This delegation of tasks into various sub tasks is an active field of research in the world of distributed artificial intelligence. [4, 7, 8, 17, 18, 20, 21, 31] argue that multiagent systems can aid in the distribution of the problem and facilitate the coordination of the activities of the traffic control instruments when needed.

We present a test bed for multiagent control systems in road traffic management. In the literature no consensus exists about the best configuration of the traffic managing multiagent system and how the activities of the agents that comprise the multiagent 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.

To be able to experiment with different strategies for the application of multiagent systems for dynamic traffic management and to examine their applicability we need a test bed. Such a test bed facilitates the development of multiagent systems for dynamic traffic management. The main requirements of the test bed are that

1. The traffic managing multiagent 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 traffic managing multiagent system can be evaluated in a realistic simulated traffic environment.

4. The traffic managing multiagent 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. An example application is described in Section 4. We conclude this paper in Section 5 with our future research.

2 Decentralized Traffic Control Concepts

In the 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 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, which is largely due to amount of data involved and the computational complexity of the problem.

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 the literature. We look at *hierarchical controller coordination*, *inter-controller coordination* and *intra-controller coordination*.

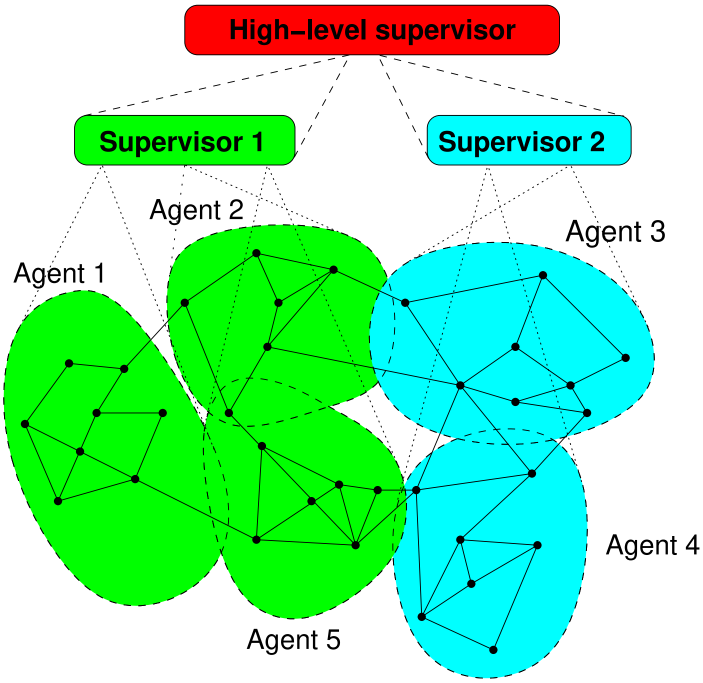


Figure 1: Schematic representation of hierarchical controller coordination

2.1 Hierarchical controller coordination

In order to maintain a network manager's overall control objective, given that part of the 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 Figure 1.

In *SCOOT* [15], and *OPAC-VFC* [11, 12], and *MOTION* [2] traffic controllers are centrally coordinated. A traffic model is used to adapt the cycle time and the offsets of the intersection controllers.

In *TRYS* [14] 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. The authors of [4] 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 of these approaches there is no communication between agents at the same hierarchical level.

In the *SCATS* [22] model the network is subdivided into regions with homogeneous flow characteristics. Coordination is achieved through communication that takes place on the same hierarchical level. Coordination between regions takes place through the adoption of a common cycle time. Coordination within a region takes place through the change of the offsets.

In *SPOT* [23], and *PRODYN* [13] the individual intersection controller are implicitly coordinated through the exchange of forecasted traffic outputs. The controller actions are not explicitly coordinated. SPOT-controllers can be coordinated on a network level through use of UTOPIA.

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 the literature many different approaches are taken to overcome this shortcoming. The most common approach is to share information among controllers (see also section 2.1). Another approach is to make the road infrastructure responsible for controller coordination.

The advantage of using an infrastructure-centric approach instead of a controller-centric approach is the fact that the former abstracts more easily to the network control objectives used at a higher level. When capacity is a constraint, a traffic operator needs to decide which traffic streams and thus which infrastructure must be given priority to. When making the road-infrastructure responsible for controller-coordination, it can be left to the road-infrastructure to come up with a new signal plan. This is the approach we take in [31].

In most cases information is only shared upstream. In fewer cases [7, 8, 20] 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 (planned, future, control actions).

2.3 Intra-controller coordination

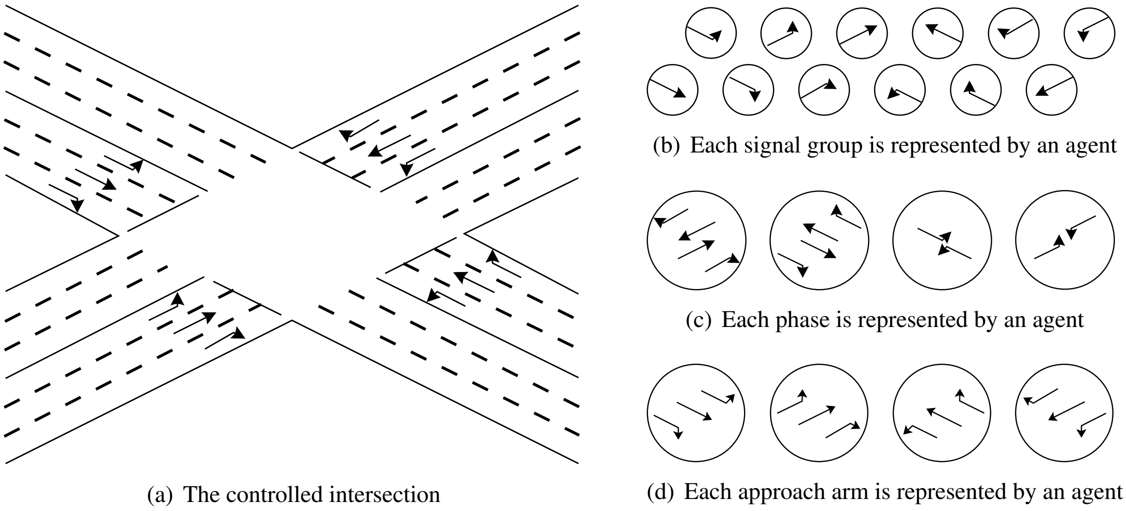
The task of controlling a single, isolated intersection is often perceived to be one, undividable, centralized control problem, but numerous examples exist in which the control of a single intersection is stated as the result of a negotiation process between multiple intelligent agents, each having their own control objective. These agents can either represent the individual signal groups [17, 18], phases [21] or the arms of the intersection. Figure 2 illustrates these different approaches for a simple intersection in which bicyle, pedestrian, and public transport signal groups are not represented.

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

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

The test bed consists of an interaction model, intelligence models, and a world model. The *interaction model* is used to model the interactions between the agents. The *intelligence models* are used to model the intelligence of the agents that collectively give shape to the traffic control process. The *world model* is used to represent the outside world, i.e. the traffic process. These models are presented in the next subsections. Figure 3 shows the relations between these models. For a more detailed discussion of the material, we refer to [30].



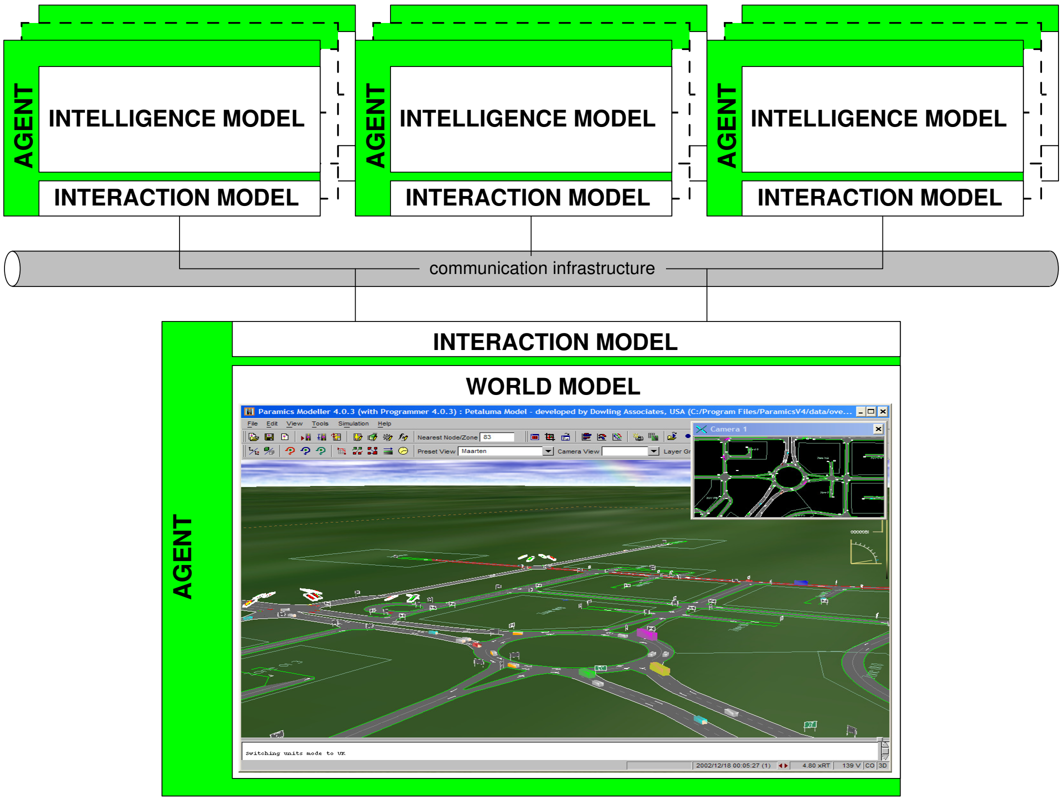
(a) The controlled intersection

(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: Distributed intersection control



Nearest Node/Zone 83

Preset View Maarten

Camera View

```
Switching units mode to UK
```

2002/12/18 00:05:27 (1) | 4.80 xRT | 139 V | CO | 3D

Figure 3: Overview of the components of the test bed

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) [9], an approach also taken by [3] for a video-based traffic monitoring system. FIPA has adopted and is working on specifications that range from architectures to support agents' communicating with each other, communication languages and content languages for expressing those messages, and interaction protocols which expand the scope from single messages to complete transactions.

The FIPA standards require that each agent publishes the services it provides to a directory facilitator. This directory facilitator is a component of the multiagent system that provides a yellow-pages directory service to agents. At least one directory facilitator must be present in the multiagent 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 through 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 (Figure 4(a)), contract net interaction protocol (Figure 4(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]. The agent development environment also provides the tools needed to evaluate the performance of the multiagent control system with respect to communication requirements.

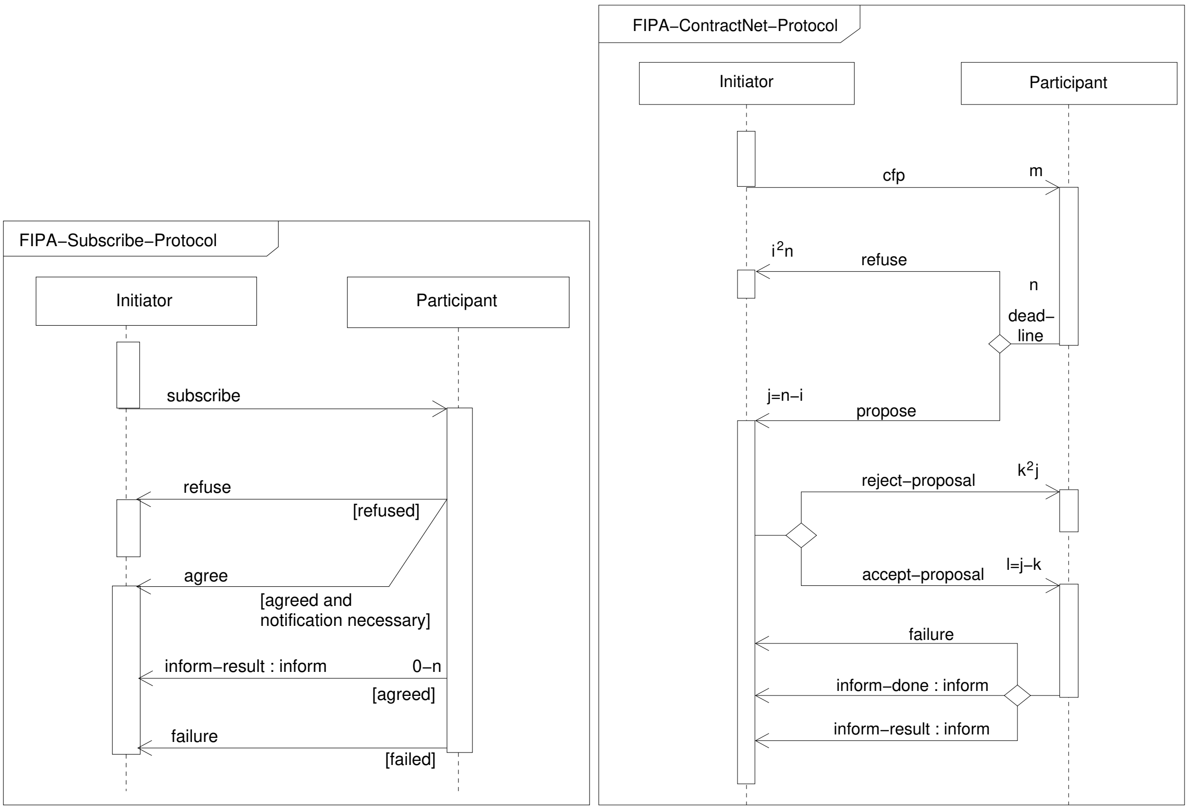
3.2 Intelligence model

The intelligence models are used to model an agent's intelligence. A fundamental decision in defining a problem is deciding how to model it. 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 multiagent systems [6, 19, 24–26, 29, 32]. Sometimes experience is available to aid in choosing the best paradigm. Often a paradigm is 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 of an agent using a conventional programming paradigm using the C(++) and Java languages, but is not limited to these languages.

Currently the test bed supports the following approaches:

1. *Rule-based inference*
   For our test bed we have developed a generic rule-based agent using *JESS*, a rule-based reasoning engine [10]. 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. The business logic of traffic engineers can be easily implemented in an expert system in the form of decision rules, which take the form of simple IF-THEN statements. A simplified example of an IF-THEN statement 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



(a) FIPA subscription interaction protocol

(b) FIPA contract net interaction protocol

Figure 4: FIPA interaction protocols

the creation and manipulation of Bayesian networks [5]. Bayesian inference is a form of statistical inference in which probabilities express degrees of belief. Bayesian inference involves the collection of evidence pointing towards or away from a given hypothesis (e.g. regarding the current traffic state). There can never be certainty, but as evidence accumulates, the degree of belief in a hypothesis changes. 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.

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 to represent the world. 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. Paramics was one of the first models providing this capability. The test bed

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

can however easily be modified to also use other traffic models that provide a suitable application programming interface. By means of a user-defined plugin, information can be retrieved from the simulation environment for use by the agent-controllers and control actions can be sent back. The simulation environment also provides the tools to evaluate the performance of the multiagent control system with respect to traffic flow.

Traffic simulation models often employ a time-step based method to simulation as opposed to a discrete event based method. 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 Figure 5. The agents in a multiagent 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 multiagent 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 (i.e. a piece of code that acts

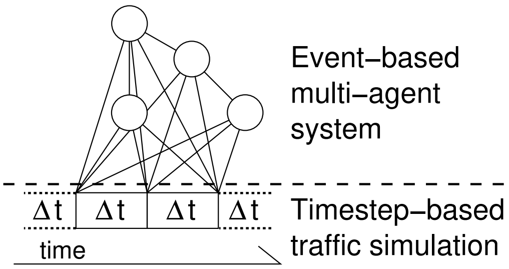


Figure 5: Interaction in discretized time

as an interface between the original application and other agents).

Simulation is used to test various real-world application scenarios for multiagent systems. In order to test whether the configured multiagent 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 multiagent system with regard to traffic flow. In that case it is required that the multiagent system gets sufficient time to formulate the control actions and that the simulations are repeatable.

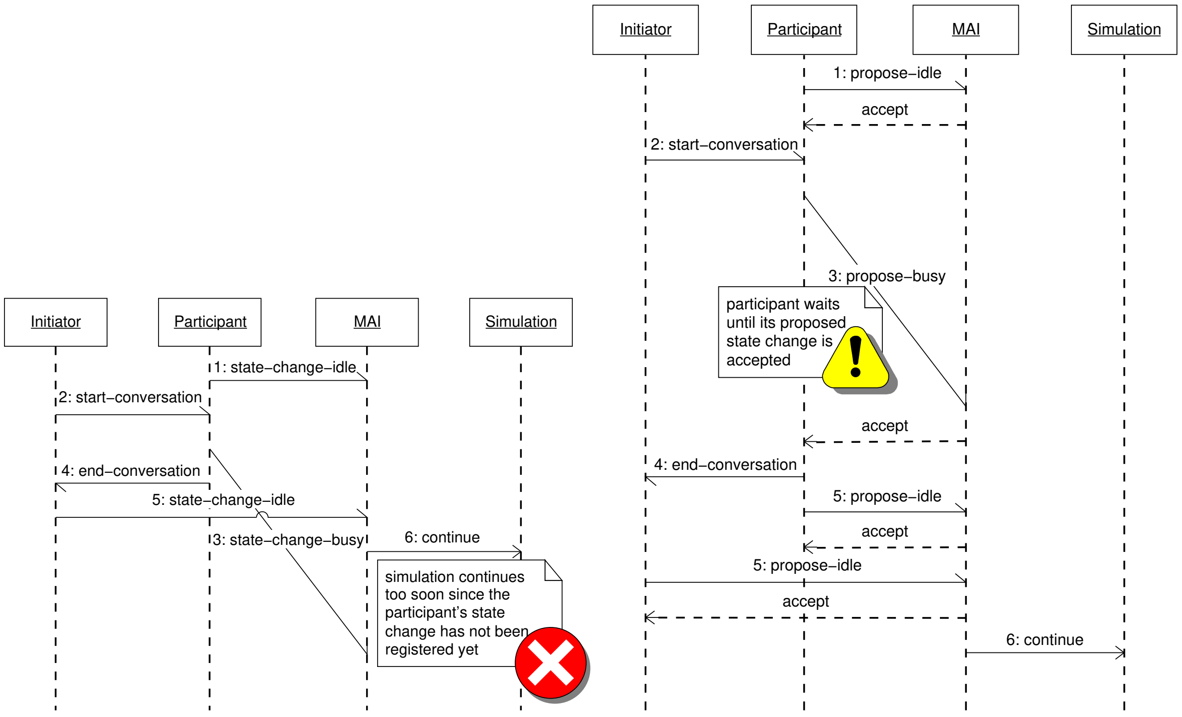
In order to guarantee that the multiagent system is given sufficient time to formulate the control actions, it has to be determined when the agents in the multiagent system have finished formulating their control actions. This is done using a special purpose agent, that, when present, requires an agent 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 multiagent system, the simulation is allowed to continue.

The FIPA propose interaction protocol is used to convey an intended state change from an agent to the MAI. This protocol mandates that proposals are explicitly accepted or refused. The explicit acceptation is required 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 Figure 6(a) where the participant message informing the MAI that it changed its state to busy (at the initiation of the conversation) arrives at a later time than the protocol initiator's message informing the MAI that it has changed its state back to idle (when the conversation has ended). Figure 6(b) shows the same communication trace where each proposed 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 MAI to communicate intended state changes, all agents operate following the higher-level state chart as shown in Figure 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 is changing state from idle to busy and vice versa, it first enters an intermediate pre-idle or pre-busy state, where it remains until it receives an accept-proposal message from the MAI in reply to the proposed state change.

4 An Application

Figure 8 encompasses all the approaches to inter- and intra-controller coordination of sections 2.2 and 2.3. Each potentially conflicting interest surrounding an intersection in this figure is represented by a



(a) State changes are not confirmed: Unsuccessful synchronization

(b) State changes are confirmed: Successful synchronization

Figure 6: Synchronization sequences

different agent. Each individual signal group (depicted by an encircled s) therefore is represented by an agent as well as each individual entry or exit link.

Given this configuration the answer to the question which signal groups get the right to green in the next phase then becomes the result of a negotiation process that takes place among the signal-group agents in which coalitions of signal-group agents are formed. For this application the network depicted in 8 was coded into the traffic simulation model which represents the world in our test bed. The JESS intelligence model of the test bed was used to equip the agents with the knowledge necessary for them to be able to protect their interests and thus give shape to the negotiation process. The interaction model provided the necessary protocols in order for the signal-group agents to actually negotiate with one another.

The resulting coalitions consist of signal-group agents that do not conflict with one another, meaning that the signal groups in the coalition can be safely given green simultaneously. The coalitions resulting from the coalition formation process are depicted in the figure as an encircled C. As the coalition formation process is a continuous process, coalitions are constantly formed and dissolved depending on the current traffic demand at the intersection. The strength of a coalition is determined by the combined interests of the individual signal groups the coalition is representing. Coalitions that contain a larger number of signal groups are therefore not necessarily stronger than smaller coalitions.

The coalition formation process roughly corresponds to finding a maximal independent set in a graph [16], which is a well-known problem in graph theory. When the conflicts between signal groups are represented in a graph G, where the set of vertices V correspond to the signal groups of the intersection and where the edges connect the signal groups that are in conflict with one another,

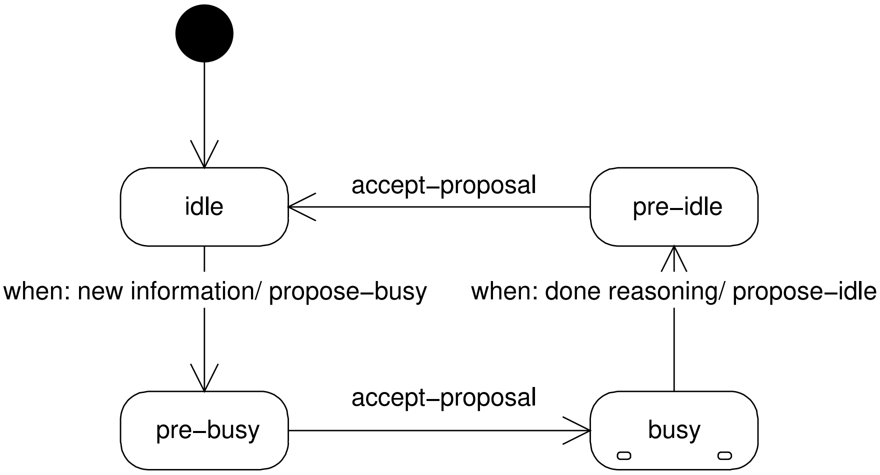

idle
accept−proposal
pre−idle
when: new information/ propose−busy
when: done reasoning/ propose−idle
pre−busy
accept−proposal
busy

Figure 7: States and transitions needed to ensure synchronized operation

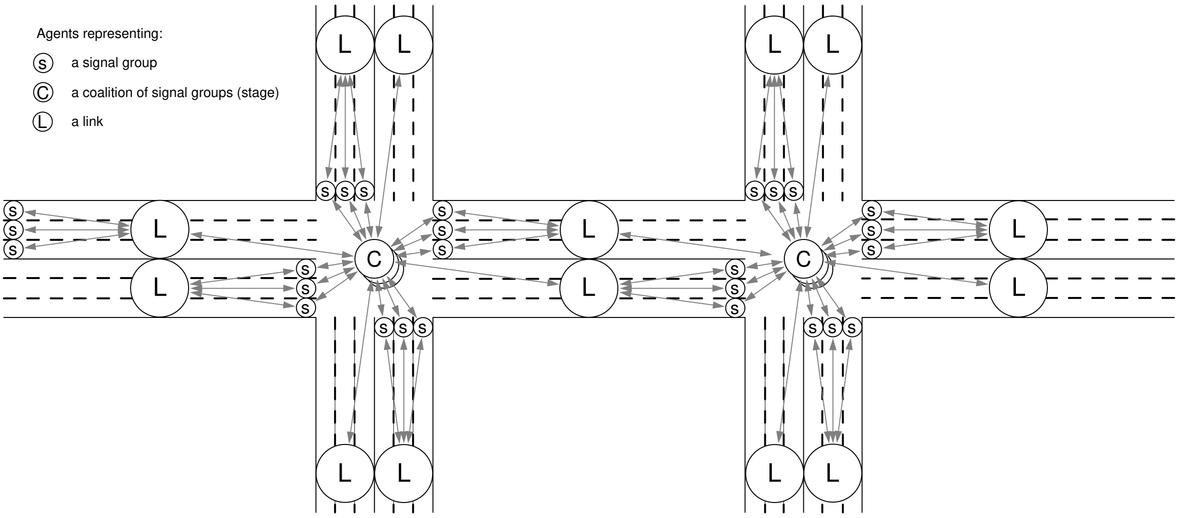


Figure 8: Multiagent coordinated intersection control

then finding the largest possible coalitions of signal-group agents corresponds to finding all maximal independent sets of vertices (signal-group agents) in that graph. An independent set in a graph is a set of vertices V' such that for every two vertices in V', there is no edge connecting the two, meaning that there is no conflict between the signal-group agents that these two vertices represent.

The maximum independent set problem involves finding the largest maximal independent set in the graph, which corresponds to finding the largest possible coalition, which not necessarily has to be the strongest coalition. The difference lies in the fact that it is not the number of vertices (or signal groups), but the weight of the vertex (or the benefit of given the signal group green) that defines the maximum. Where the opposite of an independent set is called a clique, the opposite of the strongest coalition roughly corresponds to the dominant conflict group, which is determined in the determination of the cycle time when the signal plan of an intersection is determined offline.

The agents representing the signal groups and the resulting coalitions enable us to model the operation of a single, isolated, intersection as a multiagent system. This enables us to overcome the deficiencies identified by [27] of other fully adaptive traffic control algorithms. Even given these deficiencies literature reports substantial benefits of fully adaptive control compared to existing control settings. The lack of a standardized benchmark to compare these algorithms with, makes it difficult to report exactly how much improvement is gained through the use of these algorithms, however delay reductions of up to 30% have been reported by [28]. The identified deficiencies are for large part due to concessions made during implementation of these algorithms due to their computational complexity. Since multiagent systems are naturally distributed the multiagent approach has a computational edge compared to other fully adaptive control algorithms. On a conceptual level however, when not regarding the computational complexity of these algorithms and when only a single intersection is regarded, the added value of the multiagent approach compared to these algorithms is limited. This changes however when we introduce the agents that represent the approach and exit links of the intersections.

The agents representing the approach and exit links of the intersection allow us to create a more network aware intersection controller. Entry links forewarn downstream signal groups of incoming traffic, which can lead to the formation of an emergent green wave. Simulation results show that a green wave indeed emerges when the green wave is carrying significantly more vehicles than the crossing directions. The reason for the emergence of the green wave lies in the fact that the directions that constitute the green wave hold the best cards in the negotiation process executed at each intersection.

Our simulation results also show that when traffic is equally spread among all directions of an intersection a green wave almost never emerges. We presume that this is due to the randomness in the arrival process of vehicles approaching the intersection from the north or south. In our future research we will therefore extend the network to a grid network. The randomness of the arrival process can be significantly reduced when the intersections downstream of an intersection are controlled. The exit link agent of an intersection can relate information about the downstream intersection to the upstream signal groups. We estimate that the negotiation taking place between a link and a coalition of upstream signal groups will lead to a "zipper"-like arrival process at an intersection where the "teeth" of the zipper correspond to the platoons of vehicles arriving from the conflicting directions.

5 Conclusions and Future Research

To aid the ongoing research in the field, we have developed a software environment for rapid development of multiagent control systems in road-traffic management. 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 research we are conducting using the test bed.

The presented test bed will be of great value for developments in traffic management. The compliance to FIPA-standards allows us to easily configure a multiagent system thanks to the FIPA-required directory facilitator. The compliance to FIPA-standards allows us furthermore to transfer the traffic managing multiagent system to a real-world application more easily. The rule-based and Bayesian intelligence models allow us to easily model the business logic of the traffic engineers.

However, the developed system still has opportunities for further extension. A graphical user interface can be developed in which agents can be created and the multiagent system can be configured with only a few mouse clicks. This would further accelerate the implementation of the desired multiagent 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 multiagent systems in dynamic traffic management. It proved to be a good starting point for our research in decentralized traffic control.

6 Acknowledgments

Research funded by the TNO spearhead program ”Sustainable Mobility Intelligent Transport Systems (SUMMITS)”, the TU Delft spearhead program “Transport Research Centre Delft: Towards Reliable Mobility”, and the BSIK project “Towards Sustainable Mobility (TRANSUMO)”.

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] C. Bielefeldt and F. Busch. Motion - a new on-line model for urban traffic signal control. In *Proc. 7th IFAC-IFORS Symposium on Transportation Systems: Theory and Applications of Advanced Technology*, pages 603–607, 1994.

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

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

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

[6] J. Cuena, J. Hernández, and M. Molina. Knowledge-based models for adaptive traffic management systems. *Transportation Research Part C*, 3(5):311–337, October 1995.

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

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

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

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

[11] Nathan H. Gartner. Opac: A demand-responsive strategy for traffic signal control. *Transportation Research Record*, (906):75–81, 1983.

[12] Nathan H. Gartner, P.J. Tarnoff, and C.M. Andrews. Evaluation of the optimized policies for adaptive signal control strategy. *Transportation Research Record*, (1683):105–114, 1999.

[13] J.J. Henry, J.L. Farges, and J. Tuffal. The prodyn real time traffic algorithm. In *Proc. 4th IFAC-IFIP-IFORS Congerence on Control in Transport Systems*, pages 307–312, 1983.

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

[15] P.B. Hunt, D.I. Robertson, R.D. Bretherton, and R.I. Winton. Scoot - a traffic responsive method of coordinating signals. Technical Report 1014, TRL, http://www.trl.co.uk/1024/SEARCHPUBS.asp, 1981.

[16] G. Improta. Signal control at individual junctions: Phase based approach. In Markos Papageorgiou, editor, *Concise Encyclopedia of Traffic and Transportation Systems*, pages 478–483. Pergamon Press, 1991.

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

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

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

[20] Jia Lei and Ü. Özgümer. Decentralized hybrid intersection control. In *Proc. IEEE Decentralized Hybrid Intersection Control*, volume 2, pages 1237–1242, December 2001.

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

[22] P.R. Lowrie. The sidney co-ordinated adaptive traffic system: Principles, methodology, and algorithms. In *Proc. of the IEE Conference on Road Traffic Signaling*, number 207, pages 67–70, 1982.

[23] V. Mauro and C. Di Taranto. Utopia. In *Proc. 2nd IFAC-IFIP-IFORS Symposium on Traffic Control and Transportation Systems*, pages 575–597, 1989.

[24] 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(4):577–600, October 1998.

[25] S.G. Ritchie. A knowledge-based decision support architecture for advanced traffic management. *Transportation Research Part A*, 24(1):27–37, January 1990.

[26] 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(5):347–356, September 1999.

[27] Steven G. Shelby. Single intersection evaluation of real-time adaptive traffic signal control algorithms. In *Proc. 84rd Annual Meeting of the Tranportation Research Board (TRB'04)*, January 2004.

[28] Charlie Stallard and Larry E. Owen. Evaluating adaptive signal control using corsim. In *Proc. 1998 Winter Simulation Conference*, 1998. http://www.informs-cs.org/wsc98papers/156.PDF.

[29] Henk Taale. Optimising traffic signal control with evolutionary algorithms. In *Proc. 7th World Congress on Intelligent Transport Systems (ITS'00)*, November 2000. bron: e-mail correspondentie met Henk Taale.

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

[31] Ronald T. van Katwijk and Paul van Koningsbruggen. Coordination of traffic management instruments using agent technology. *Transportation Research Part C: Emerging Technologies*, 10(5–6):455–471, oct–dec 2002.

[32] H. Zhang and S.G. Ritchie. Real-time decision-support system for freeway management and control. *Journal of Computing in Civil Engineering*, 8(1):35–51, 1994.