A Test Bed for Multi-Agent Systems and Road Traffic Management

Alexander Th. van den Bosch a Maarten R. Menken a
Martijn van Breukelen b Ronald T. van Katwijk c

a Vrije Universiteit, Amsterdam, {atvdbosc, mrmenken}@cs.vu.nl
b TNO TPD, Delft, breukelen@tpd.tno.nl
c TNO Inro, Delft, r.t.van.katwijk@inro.tno.nl

Abstract

In this paper we present a test bed for multi-agent systems in road traffic management. This software environment is developed at TNO to aid in-depth research in this field. The test bed establishes a connection between the traffic simulator Paramics and the multi-agent platform JADE. Special Jess agents have been developed to accelerate the implementation of different agent-based traffic control strategies. Two example implementations demonstrate the functionality of the developed system and function as a starting point for further research.

1 Introduction

Throughout the world, traffic congestion forms a daily recurring problem. In The Netherlands, more and more instruments are deployed to stimulate an undisturbed flow of traffic on highways and urban road networks. For example, ramp metering installations (RMI) are used to regulate the inflow of traffic at on-ramps. Another example is the variable message sign (VMS), which keep motorists informed about the current road conditions ahead. They can also realize temporary lane blockings and overrule maximum speeds. Research is being conducted in the field of automatic coordination of these dynamic traffic management instruments. Van Katwijk and Van Koningsbruggen argue in [15] that the communication capabilities of multi-agent systems can be used to accomplish this coordination.

In the project ‘Verkeerscentrale van de Toekomst’, TNO aims at the development of new concepts for the changing role of traffic operators. In one of the concepts, multi-agent systems are used to assist the real-time coordination of automatic traffic management instruments. The task of the human traffic operator is shifted towards higher-level traffic control.

Yet, no consensus exists about the best configuration of the traffic managing multi-agent system. In the vision of TNO, 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, TNO formulated the need for a test bed. The test bed should facilitate the development of multi-agent systems for dynamic traffic management. It should be easy to use and test multi-agent strategies in a realistic simulated traffic environment and the development process should not be hindered by cumbersome implementation details. This paper describes an implementation of this test bed that was developed during an internship at TNO, conducted by the first two authors. It also describes the two experiments that have been conducted in order to demonstrate its functionality. For a detailed discussion of the material, we refer to [13], on which this paper is based.

2 Related Work

Several technologies from the field of Artificial Intelligence are being applied in dynamic traffic management including Evolutionary Algorithms, Knowledge-Based Systems, Neural Networks and Multi-Agent Systems [12]. For clarity, a distinction can be made between vehicle-oriented and measure-oriented traffic control:

Vehicle-oriented traffic control focuses on the control of the behavior of individual vehicles. For example, Moriarty, Handley and Langley model each vehicle as an intelligent agent that can be influenced [10]. Adler and Blue present a traffic control method through in-vehicle routing and navigation systems in [6]. Other examples are provided by [5] and [1].

Measure-oriented traffic control assumes that traffic consists of entities with goal-oriented behavior. Control of this behavior is performed through external signals at fixed locations, like traffic lights and variable message signs. Our research focuses on this form of control.

To research the role of multi-agent systems in measure-oriented traffic management, Hernández, Cuena and Molina have developed the TRYS model [9]. 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. Roozemond and Rogier also propose higher-level agents to perform conflict resolution [11]. The authors list adaptability, communication and proactive behavior as the most important benefits of multi-agent systems in traffic control.

Ferreira, Subrahmanian and Manstetten take another approach by presenting a fully decentralized traffic control mechanism using agent technology in [8]. No higher-level agents are present for conflict resolution. The sharing of ‘opinions’, numerical estimations of the traffic situation in particular areas, must result in conflict-minimizing multi-agent behavior. Simulations have showed that the use of these opinions can result in coordinated and more efficient traffic control.

Van Katwijk and Van Koningsbruggen propose in [15] the use of multi-agent systems to relieve the task of the human traffic operator. They demonstrate by
means of two examples the need for coordination of traffic control measures and the benefits of multi-agent systems in this field. According to the authors, the main advantages of multi-agent systems are proactiveness and social behavior. A hierarchical multi-agent structure for the sake of conflict resolution is suggested. Van der Arend [14] was the first at TNO to actually implement a multi-agent system for dynamic traffic control. The new ideas and research questions that resulted from this first implementation, and the ambition of TNO to participate in research on this topic, gave rise to the desire for the test bed presented here.

3 Software Components of the Test Bed

In traffic control, two processes can be distinguished, as depicted in Figure 1. First of all, there’s the traffic process in which all motorists and other traffic participants are involved. Through the use of instruments the traffic situation can be observed. These observations are used in the traffic control process. In this second process, decisions are made when and how to perform dynamic traffic management measures in order to optimize the traffic process.

In our test bed the traffic process is simulated by Paramics; the control is delegated to a multi-agent system implemented in JADE. This way a mapping can be made from the actual traffic management cycle to a simulated environment. This is shown on the right-hand side of Figure 1.

This section discusses how the test bed simulates the traffic process and can be used to build implementations of multi-agent strategies for the traffic control process. Paramics, JADE and two other important software components that make up the test bed are presented in the next subsections.

3.1 Paramics

Paramics is a microscopic traffic simulation suite developed by Quadstone Ltd. [3]. Figure 2 contains a screen shot of the Paramics Modeller. The Modeller is used to define a traffic network and the amount of simulated traffic. After that, the traffic process is simulated on the level of individual vehicles. Paramics can be extended through an API. User-defined plug-ins, written in C, can retrieve traffic simulation information from Paramics and send back control actions.
3.2 JADE

JADE is an agent development environment implemented in the Java language [7]. It provides the facilities for agent autonomy, inter-agent communication, task execution and agent management. The agent developer extends the JADE Agent and Behaviour classes to develop application-specific agents and tasks.

3.3 Paramics-JADE Software Bridge

The third component of the test bed is the Paramics-JADE software bridge. This interface has been developed both to be able to provide agents with traffic information from the Paramics simulation and to allow agents to control the simulated traffic measures such as ramp metering installations (RMIs). The bridge has been implemented as a plug-in for Paramics using both the C and Java programming languages.

3.4 Generic Jess Agent

For our test bed we developed a generic Jess agent of which the agent-specific task is executed by Jess, a Java rule-based reasoning engine [2]. Incoming messages are converted to Jess facts and asserted into its working memory. Derived facts describing messages to be sent are translated into corresponding JADE messages, after which JADE takes care of their delivery. This Jess agent has a number of advantages.

- It enables the user to make use of the pattern-matching capabilities of Jess. As a result, the user can focus on the logical aspects of the agent’s knowledge.
• An agent window displays the state of the agent and gives insight in internal agent processes. This includes its knowledge about the world and other agents, and a history of its derivations. This way the reasoning process can be traced easily.

• With the generic Jess agent, different agents can be defined without the need to compile the specific instances. Agents are defined by providing an instance of the generic Jess agent with agent-specific knowledge in Jess format. When the agent is created in JADE, this knowledge is loaded into the rule-engine.

From the perspective of the multi-agent system developer, the following two 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.

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 their corresponding agent-specific knowledge.

4 Scenario

To demonstrate the test bed, we implemented two alternative multi-agent systems for the first example described in [15] by Van Katwijk and Van Koningsbruggen. Figure 3 depicts the traffic situation, containing one highway and three entrance ramps. The inflow from each ramp to the highway can be regulated by a ramp metering installation. Suppose congestion starts to build up on the highway downstream of the third ramp. In this case, the third RMI will detect this and will reduce its inflow. The second RMI will not start to do the same until the congestion has reached the area directly downstream of the second entrance point. This also applies to the first metering installation.

Using our software, we have developed a multi-agent system, capable of communicating about the actions to be performed. 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 timelier.

In our first agent implementation the highway is divided into three parts, each provided with two detectors and one agent. Based on the detector data the agent forms an image of the current traffic situation. Depending on this estimate and on incoming messages from agents downstream, the state ‘no problems’, ‘request for help, urgency low’ or ‘request for help, urgency high’ is communicated to the neighboring agents upstream. Apart from these three highway agents, three RMI
agents are defined. Their only task is to control the RMI, based on incoming help requests. They are not responsible for the traffic situation on the ramps. This multi-agent configuration is shown in Figure 4. The arrows indicate the communication possibilities of each agent. In this way the knowledge about the current traffic situation propagates upstream through the network.

Figure 5 shows an example of a Jess rule. It is part of the rule base of the first highway agent and describes the situation in which congestion occurs near Detector 2 and no problems are reported by the second highway agent. In this case, the first highway agent decides to send a request for help to the first RMI agent.

In the agent implementation of Figure 4 messages travel along the infrastructure. Another way to arrange the communication is to make the instruments responsible for the communication upstream. In that case the arrows between the highway agents are moved downwards to the RMI agents. Simulations show that in both agent implementations congestion is tackled earlier. This can also be demonstrated by means of traces, which are elaborated in [13, pp. 65–68].

The two agent implementations proved that the system enables users to create
different multi-agent systems for dynamic traffic management. Furthermore, the development process is simplified and accelerated by the incorporation of rule-based reasoning.

5 Conclusions and Future Work

To aid the ongoing research in this field, we developed a software environment for rapid development of multi-agent systems able to interact with a traffic simulation. 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 reasoning. The implementation of two multi-agent systems with the test bed is described. The two existing implementations show that multi-agent systems can be created easily.

In our opinion the presented test bed will be of great value for developments in traffic management. However, the developed system has opportunities for further extension. A graphical user interface can be developed in which agents can be created with only a few mouse-clicks and in which agent communication is visualized. This would further accelerate the implementation of the desired multi-agent system. Extending the number of available traffic control instruments 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. The presented agent implementations can be a starting point for further research. For example, the mechanism can be extended with agents who are responsible for traffic on the ramps. Negotiation can be a means of weighing the flow on the highway against the interests of vehicles on the entrance ramp.

References


