# Application of multi-agent systems in traffic and transportation

B. Burmeister
A. Haddadi
G. Matylis

*Indexing terms: Multi-agent systems, Agent-oriented sytems*

**Abstract:** Agent-oriented techniques offer a new approach to support the whole software development process. All the phases in the software development process are treated with a single uniform concept, namely that of agents, and a system modelled by a collection of agents is called a multi-agent system. AOTs as a new advance in information technology can help to respond to the growing interest in making traffic and transportation more efficient, resource-saving and ecological. The authors give an overview of a diverse range of applications where multi-agent systems promise to create a great impact in this domain. To demonstrate the ideas behind AOTs and their applicability in this domain, two applications currently under development at Daimler-Benz Research are described in some detail.

## 1 Introduction

Agent-oriented techniques (AOTs) offer a new approach to support the whole software development process. In particular, they enable the analysis, design and implementation of large and complex systems by providing abstraction levels that make it simpler and more natural to deal with the scale and complexity of problems in these systems. The goal of AOTs is to handle all phases with a single and uniform concept, namely that of agents. Systems modelled by a collection of agents are called multi-agent systems (MASs).

We view agent-oriented techniques as a further development of techniques in distributed systems and object-oriented systems.

AOTs can lean on techniques provided by distributed systems (like sharing resources and synchronisation), but they complement these techniques by making subsystems more autonomous and enabling them actively to coordinate their activities instead of being coordinated by design.

AOTs extend object-oriented techniques in that the analysis, design and realisation of complex systems are performed on a higher level of abstraction, and agents are active and concurrent objects embedded in a sort of society. Furthermore, agents can be considered as a specialisation of objects, in that messages exchanged among agents are characterised by message types that have data/content-independent semantics. As part of this specialisation, some go further than this and distinguish agents from objects by making a commitment to structuring internal states of an object in terms of some abstract mental notions such as beliefs, goals, and intentions. They subsequently define an agent's behaviour in terms of the interplay of these attitudes [1].

From a technical point of view, four ingredients are considered to make agents distinct from objects in object-oriented systems and subsystems in distributed systems, namely internal/mental states, message types in communication, protocols for coordination, and roles within organisations. Although not all these features are commonly agreed to be the distinguishing features of agents, we believe that for a full-fledged agent-oriented technology and the problems promised to be solved by this technology, the present (rather diverse) approaches to agent systems will evolve and converge to a unified direction that in some way will include all these features.

As for internal states, we follow an operational approach that relates the internal states of an agent to its intentionality, especially to the triple: beliefs, desires and intentions (BDI). This model is known as a BDI agent architecture [2], which constitutes the backbone of our work in AOTs. This architecture is briefly described in Section 3, which also provides the arguments for our commitment to this model.

A feature that is more commonly accepted as the factor that distinguishes agents from objects or any other program, is that agents exchange messages characterised by message types. Message types have data/content-independent semantics, and thus they facilitate the general interpretation of messages. In our work we have defined a set of message types that have proved to be useful principal performatives for the type of communication that could take place among agents (in contrast to communication among other types of program). These message types can be further specialised for the requirement of applications, to facilitate further message interpretation. The underlying theory and the formal semantics of these message types have been studied in [3].

As mentioned above, the third distinguishing feature in agent systems is protocols for coordination. For this purpose we have devised a methodology to represent


*IEE Proceedings* online no. 19971023
Paper received 22nd October 1996
The authors are with Daimler-Benz Research Systems Technology, Alt-Moabit 96a, D-10559 Berlin, Germany; bur@DBresearch_berlin.de

and execute cooperation protocols [Note 1]. The preliminary work on cooperation protocols has been reported in [4]. Cooperation protocols can be used by designers to manage dialogues among agents, by specifying the possible courses of dialogue with respect to a particular context. With BDI agents they can be directly executed and used by agents as a guideline to indicate what to communicate and what types of reply to expect in a given state. The protocols developed for our applications make use of the set of message types mentioned above. These protocols are sufficiently generic that they can also be used to specify dialogues among heterogeneous agents, that is, agents with different internal structures, and they are configurable so that they can be specialised to the requirements of applications.

Still under research is the impact of roles that agents have in organisations, which we find as an important topic of study in AOTs to model large systems.

We foresee a rather broad potential for AOTs and MASs: agents might be considered at conceptual, descriptive or operational levels. AOTs can be used for the analysis and design of complex systems, for simulation and planning purposes, to enhance the autonomy of single systems components, and to realise new systems solutions. AOTs are expected to have a significant impact in problem domains where the problem is functionally or geographically distributed, where the subsystems exist in a dynamic environment, and where the subsystems need to interact more flexibly. To emphasise the point once more, the benefits that one gains from AOTs are the reduction of complexity in systems design due to the concise and natural modelling of the problem domain, and enhanced robustness and adaptivity due to self-organisation of the subsystems.

The domain of traffic and transportation is geographically and functionally distributed; subsystems have a high degree of autonomy, and typically we need to deal with settings with a variety of dynamics. Owing to these characteristics, many applications in this domain can be adequately modelled as multi-agent systems. As a result of this modelling we could also meet the growing interest in making traffic and transportation more efficient, resource-saving and ecological.

To profit from advances in information technology, various national and international programs have been established. Daimler-Benz Research is undertaking various projects to explore the possibilities and advantages of the MAS approach in improving the existing systems, as well as developing challenging new applications.

## 2 Potential of multi-agent systems in traffic and transportation

As mentioned above, many areas in the traffic domain exhibit the very characteristics for which AOTs have been developed: traffic is geographically and functionally distributed, subsystems have a high degree of autonomy, and typically traffic applications are highly dynamic. The remainder of this section presents an overview of the potential (and the existing) applications of AOTs in traffic management, traffic guidance and control, and capacity and resource management.

Note 1: Historically we refer to these protocols as cooperation protocols, and to keep consistent with our earlier works we use the same term throughout this article.

### *2.1 Traffic management*

Traffic management is largely the process of planning, implementation, testing and administration of traffic to optimise the assignment of traffic supplies to traffic demands under economic and environmental features. Of increasing importance in traffic management is the integration and interconnection of different traffic means (i.e. intermodal traffic systems) and their infrastructure.

There are a number of national, European and international projects aimed at applying the most recent information technology to govern traffic systems. The majority of these projects are specifically targeted at road traffic (e.g. PROMETHEUS, DRIVE and ITS). Dynamic park and ride information, travel information and reservation system, driver routing guide system, inter-connections information system, fleet management and emergency system are some examples of integrated traffic management systems that are studied in these projects.

Taking these examples as references, we briefly describe some of the areas where AOTs can have a great potential impact.

*2.1.1 Analysis and description of traffic systems:* To manage the scale and the complexity of traffic information, AOT approaches can be used to structure and appropriately combine the information into a comprehensible form. AOTs provide the necessary tools for analysis, modelling and design of the whole traffic system in terms of its subsystems, each with its own set of local tasks and capabilities. Integration can then be achieved by modelling the interactions among the subsystems.

*2.1.2 Increasing the autonomy of individual traffic components:* Currently most of the traffic centres provide suggestions and information unique to their site and/or specific domain. Though some of these centers are interconnected, in most cases the subsystems play a rather passive role. Using AOTs each subsystem can be modelled as an agent with a sufficient degree of autonomy that would enable the subsystem to have an active control on its decisions and activities. By devising appropriate interactions among the agents, the subsystems can cooperate with one another to increase their performance (e.g. better quality information, multi-modal routing and connections information, etc.), and coordinate their activities to collaboratively adapt to the dynamics of traffic demands (e.g. providing extra vehicles or dynamically increasing train frequencies).

*2.1.3 Integration framework for the existing and future systems:* An example of this is a system that is linked to an emergency rescue management centre. When an accident related to flammable fluids and chemicals occurs, on the site the system can transmit information on the content being transported to a decision support system that computes a prognosis and makes appropriate suggestions. Such a system could work with agent systems that combine the data on pollution concentration and composition, pollution spread and development and the meteorological data, to provide appropriate guidelines.

### 2.2 Traffic guidance and control

Independently of the particular medium of transport (road, rail, air, etc.) one could differentiate between the interaction: among various means of transport, among various components in different traffic infrastructures (e.g. traffic lights in road traffic, airports in air traffic), and between the means of transport and components in the traffic infrastructure. In almost all traffic forms, information and communication techniques are predominantly applied to the second type of interactions (i.e. to the components of traffic infrastructure). In their interaction with these components (the third type), the means of transport play rather a passive role in that the instructions come from a central traffic guidance system. With respect to the first type, even if there is any interaction between various means of transport, this interaction is mainly data exchange; there is no active co-ordination of activities among the means of transport. Agent-oriented approaches may improve existing road, air and rail traffic guidance and control systems by increasing the autonomy of single systems and enhancing their interaction capabilities.

*2.2.1 Road traffic:* Many system components used in road traffic guidance (e.g. changing traffic signs for the surveillance and control of traffic flow on highways, mobile and stationary meters; and traffic jam, fog, and speed warnings) can be made more autonomous and consequently more adaptive, if modelled as agents.

Particularly unsatisfactory is the degree of robustness and flexibility of the existing systems, which is basically due to the fact that the number of possible states that the network may be in is immensely large, and therefore it makes little sense to specify central operation plans. The occurrence of new events leads to consequences that would rapidly change the whole state of the network, and as a result the central traffic regulating plans never execute into completion. Off-line decision-support systems are now used to deal with this problem.

Use of AOTs would lead to a greater flexibility and robustness. The knowledge about the current traffic flow and general data (such as building sites, the road network, etc.) can be captured locally by regional/zone agents who control the traffic performance of each region/zone. If an event may have consequences for the areas under the control of the neighbouring zones, the corresponding agents can be informed so that they collaboratively adjust and coordinate their plans accordingly. Such a system has the advantage that the regional changes and interferences are considered at the regional level and are dealt with immediately. Should these events have consequences on any other regions, only the agents in charge of the affected regions must replan their activities. Consequently, the system can much more rapidly react and adapt to the new situations. Instead of an off-line decision support system, using agents this support can be provided on-line.

With the increasing awareness of the importance and urgency of environmental considerations in traffic, more ecological traffic routing systems can be imagined. These systems may consist of a network of interconnected stationary and mobile measuring stations, vehicle electricity charging stations, prognosis systems and traffic routing centres. Such a system can be modelled as a multi-agent system where agents communicate and interact to reduce environmental damage.

*2.2.2 Air traffic:* Compared with the other modes of traffic, information techniques have had wider application in air traffic to attain more efficiency.

The tasks in Air Traffic Control (ATC) are geographically and functionally distributed. The airspace above the airport areas where take-off, landing and land manoeuvres take place is distinguished from the area close to the airport where planes descend and ascend, and from the air terminals for different type of flights (e.g. domestic and international flights). The terminals are in turn divided into geographical sectors that are controlled and regulated by an autonomous controller. The airspace above an airport used for take-off, landing and land manoeuvres is also controlled by a separate system.

The geographical and functional distribution and the highly dynamic nature of ATC makes it an ideal candidate, with many potential applications that can be modelled with MAS. In addition to the controllers, there are other components that can be modelled as agents; for example, aircraft, radar equipment, flight security systems, airport information systems, and luggage handling systems. For example, an agent-based taxonomy has been described in terms of roles and capabilities (e.g. aircraft controller and security equipment) [5], and in [6] the author further makes a task analysis of ATC regarding the information exchange necessary and the desired cooperation between the agents.

In ATC, aircraft can also be modelled as agents who coordinate their activities. This could be a significant improvement over the present systems, where the air traffic is largely centrally controlled from the ground. Active cooperation would also enable the automation of many currently uncontrolled areas of flight manoeuvres (e.g. distance maintenance). Examples of different applications of MAS to air traffic control can be found in [7, 8].

*2.2.3 Rail traffic:* The majority of the present rail transporters run on a rigid schedule that often does not accurately reflect the transport demands. As a result, in many occasions trains run empty over long routes, whereas at peak times they are overloaded/overcrowded. With agent-oriented approaches one could realise a demand-oriented schedule, where the vehicles have a higher degree of autonomy and dynamically react to the transport requirements accordingly. Such an approach may first be simulated to investigate and test the necessary measures for an adequate infrastructure, and in a later phase investigate and devise an appropriate model of vehicle autonomy.

With the increasing tendency to hand over various rail traffic services to the private sector (e.g. various privately owned companies offering rail transportation services over different networks, or parts of the network over specific time periods) it is seen as necessary to shift from a largely centralised control of railway, goods traffic and commuter transport, into a decentralised control. For example, the control in the rail infrastructure (e.g. rail junctions) can be removed from the central systems that control the infrastructure, to the individual trains that operate on those networks. Such a distribution can be conceptually supported by applying agent-oriented techniques.

### 2.3 Capacity and resource management

Capacity and resource management largely concerns the provision of transport means based on demands, under the legal, economic, business and technical restrictions and criteria. Such tasks can be seen in various organisations, such as shipping and freight logistics, taxi firms, public transportations and airlines.

Capacity and resource management typically involves: route planning; planning the schedule for unloading, waiting, loading and dispatching; and allocating the crew. Each of these tasks is in principle mathematically solvable, and there are already many algorithms in operations research (OR) that efficiently optimise individual problems similar to those mentioned above. However, these algorithms are successful when only one or a very few criteria need to be considered (i.e. when they are aimed at optimising specific problems). They prove to be very unsatisfactory when solving problems with dynamic restrictions, and when a combination of problems needs to be optimised.

This section will give some examples of possible and existing applications of MASs in this domain.

*2.3.1 Fleet management:* The existing fleet management systems for goods transportation are largely restricted to controlling on the level of representation and evaluation of costs and performance data in relation to single vehicles, by assigning and allocating routes, customer orders, drivers and the workshops. Active planning, however, is not part of the controlling process in these systems. With a multi-agent architecture where some of these tasks are delegated and carried out by agents as an integral part of the controlling process, agents could also carry out active planning. That is, by choosing and combining a fleet based on the incoming orders, such a system could serve dynamically to compute adequate fleet configurations.

Freight logistics is one part of fleet management that stresses tour planning. Working through freight orders for shipment and logistics involves estimating if the fulfillment of the order is possible, scheduling the orders, allocating and provisioning transport and transfer capacity, collecting, loading and distributing goods, and finally storage allocation and store keeping.

Multi-agent systems can serve to improve the current techniques by better use of storage capacity, reducing the frequency of transporters riding unloaded over long routes, and still being able to optimise routes within the short planning time. Examples of systems that demonstrate the applicability of MAS to this area have been described in [9–11].

*2.3.2 Air traffic:* Fleet management has also come to play an important role in business management for airlines. Airlines all over the world have large departments that deal only with fleet management problems. Currently this is done by two types of system: aircraft management systems which plan the type of aircraft to operate over various routes for various transportation requirements and purposes, and the amount of time required by each aircraft on the ground (for refuelling, inspections, etc.); and flight scheduling systems which assign crews to aircraft and schedule the flights. The current systems produce a sequential plan to connect flights, but are not capable of coupling the consequences of unexpected changes. In this respect, MAS approaches can help to improve the existing systems. Each agent can be responsible for part of the planning and scheduling and to try to optimise its local plans according to the given constraints. Furthermore, because of the ability of agents to adapt dynamically their activities to unpredictable circumstances, on unexpected changes in one schedule, those affected can collaboratively reschedule and adjust their part of the schedule. Consequently, an MAS approach can lead to a more adaptive flight planning and scheduling system. Some of the problems involved in this application are discussed in [12], and a multi-agent systems approach is proposed to solve some of these problems.

*2.3.3 Shipment and transfer at airports and harbours:* In the recent years increasing effort has been put into optimising such airport capacities as aircraft parking positions, gates, information rooms and luggage collection stands. For this purpose, the airport management must develop a long-term scenario for capacity planning, must construct seasonal pre-assignment models, or must quickly react to plan changes in the business.

An additional problem here is the efficiency of the organisation of the airport forefield traffic, with about a dozen types of staff: police, customs, catering, fire brigades and so on. MAS offers an appropriate modelling and a basis for a simulation and planning tool. In the ESPRIT project IMAGINE a demonstration example for the air-catering domain was developed, in which the agents in the scenario represented airlines, aircraft, warehouses of catering firms, transport vehicles, a dispatcher and a ground traffic controller [13].

There is an increasing shift towards transportation by ship containers, and consequently a new type of infrastructure is required. The allocation of containers to storage units, the movement of transporters and scheduling of the storage operations are highly interrelated. For instance, a knowledge-based system is described in [14] that will be installed in Singapore. Although there are some potential applications in this domain, we are not aware of any reported MAS activities in this area.

In this Section we briefly reviewed some of the potential areas in the traffic and transportation domain where the application of AOTs is very promising. The remainder of this paper will describe the approaches and the results of two research projects at Daimler-Benz Research. These projects are parts of activities aiming at exploring the possibilities and advantages of the MAS approach in improving the existing systems, as well as developing challenging new applications.

## 3 The Belief, desire, intention (BDI) agent architecture

The BDI architecture is a meta-level control mechanism for practical reasoning. Practical reasoning is the process of reasoning about actions in dynamic, unpredictable environments. The BDI architectural models an agent as having beliefs (a database of facts), desires (a set of tasks that an agent is in principle capable of accomplishing), and plans (procedures that if executed successfully would each achieve a certain desired effect). We technically differentiate goals and intentions from desires and plans in that goals and intentions, respectively, represent the desires that an agent has chosen and committed to accomplish and the plans the agent has chosen and committed to execute to accomplish those goals, both under what the agent believes to be possible in a given situation. In other words, although desires and plans denote some static or permanent attitudes, goals and intentions denote the dynamic attitudes of an agent.

The underlying control mechanism is both event-driven and goal-driven, and can in general described as follows: at any given time, an agent is executing a plan to achieve a certain goal. Plans post new (sub)goals for which new (sub)plans have to be selected and intended for execution. Occurrence of events denotes changing circumstances that may violate the conditions necessary for executing the existing intentions or achieving the existing goals. Therefore upon the occurrence of new events, an agent may reconsider its current goals and intentions and if necessary adopt new goals in response to the changes denoted by the events. In this case, the execution of the existing intended plans will be interrupted and either suspended for later execution or terminated.

The reason for adopting a BDI architecture was that many of our applications demanded a model for better management in developing large and complex distributed systems, in addition to the crucial need for principles that enabled the system components to gracefully recover from the consequences of unpredictable states that result from the dynamics inherent in many such systems. In our experience, the BDI model provides a simple and general, yet very powerful mechanism to meet these requirements. There is a considerable amount of theoretical foundation that describes and analyses the interplay of these attitudes (mainly, beliefs, plans, desires, goals and intentions), and their role in practical reasoning is by now well-understood. Unlike many other agent architectures that have been developed based on some *ad hoc* principle, BDI agent architectures are based on the general concepts and principles extracted from the corresponding theoretical and experimental studies and analyses.

In our experience, this general model offers by far the best and the most powerful solution to interleave deliberative and reactive behaviours in practice (i.e. practical reasoning). This model may not be appropriate for agents with little need for practical reasoning; however, as multi-agent systems are inherently dynamic, this model can be used as a meta-level control mechanism to introduce greater flexibility in monitoring and administering interactions with other agents, and this is basically how agents can adapt to the dynamics of the system (for a detailed discussion see [3]).

The existing development and execution environments for BDI agents that will be referred to below, immensely reduce the task of managing and developing large and complex systems. Agent components can be described at a high level of abstraction (basically when using these tools, only the plans of agents need to be described/encoded), yet the resulting systems are very efficient, and this is a very important criterion in the acceptability of AOTs in practice.

## 4 Agent-oriented traffic simulation

### 4.1 Overview

One domain in which AOTs can improve existing techniques is in microscopic traffic simulation. Increasing traffic volume urgently requires efficient solutions. Broadening roads or expanding the road network neither offer a reasonable long-term solution, nor are always possible. Hence the capacity of a road network must be increased by intelligent traffic flow and operational control systems. In this context traffic modelling and simulation techniques can be used to investigate and evaluate the alternative controlling strategies.

Traffic simulation models are classified into macroscopic and microscopic models. The hydrodynamic approach to model traffic flow is typical for macroscopic modelling. With this kind of approach one can only make statements about the global qualities of traffic flow. To observe the behaviour of an individual vehicle a microscopic simulation is necessary. Because traffic cannot be seen as a purely mechanical system, a microscopic traffic simulation should also take into consideration the capabilities of human drivers (e.g. perception, intention, driving attitudes, etc.). A driver and a vehicle are jointly modelled as a driver–vehicle element. Each driver–vehicle element may be seen as an autonomous and intelligent subsystem (i.e. an agent).

Enhancing the existing systems by agent capabilities leads to a more adequate simulation approach. From a technical point of view, the advantages of the agent-oriented approach are a high-level description of driver–vehicle behavior, which is easier to edit, change, modify and maintain: and the possibility to model explicit communication and co-ordination among vehicles.

The agent-oriented traffic simulation system presented here is based on a microscopic traffic simulation model of the Daimler-Benz Traffic Research [15]. This system has been implemented using the COSY modular agent architecture [16] and the multi-agent development environment DASEDIS [17], developed by the MAS group of the Daimler-Benz knowledge-based systems department.

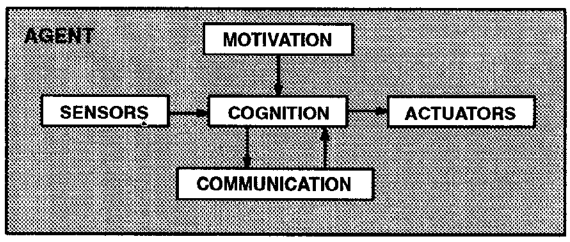


**Fig. 1** *COSY agent architecture*

As shown in Fig. 1, the COSY agent architecture consists of five top-level modules. ACTUATORS carry out the so-called effectoric behaviour, that is behaviours that has visible effects on the environment. Through SENSORS, the agent perceives the environment and other agents.

The COMMUNICATION module is responsible for explicit communication with other agents, (i.e. sending and receiving messages). The MOTIVATION module models the agent's long-term goals, preferences, roles and in general elements that characterise an agent and influence the way an agent makes decisions.

The COGNITION module controls and monitors an agent's individual, communicative and co-operative activities. It has an extended knowledge-based architecture, with the BDI architecture as its underlying principle [18].

The architecture of the DASEDIS development environment has been derived from the modular agent

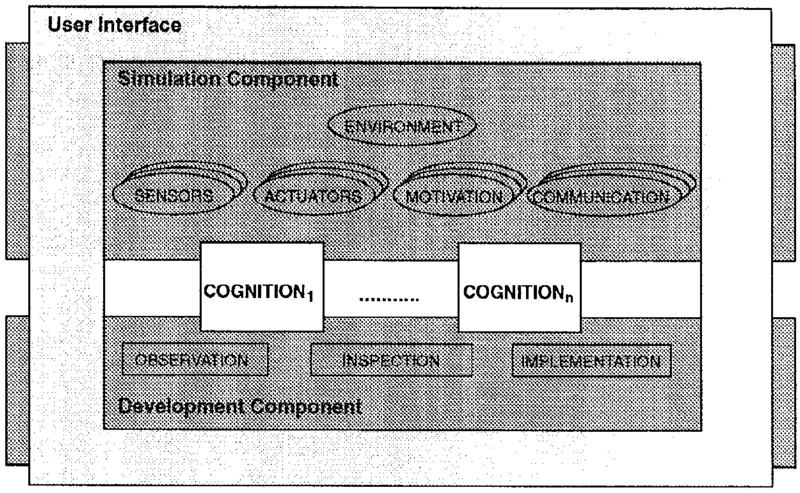


**Fig. 2** *DASEDIS*

architecture described above. The main components of DASEDIS (Fig. 2) are a simulation component and a development component.

The simulation component simulates the MOTIVATION, SENSORS, ACTUATORS, COMMUNICATION, and ENVIRONMENT by models specific for each application. The development component provides:

(*a*) a set of tools to implement the application-specific knowledge base of each agent. These tools enable an application designer to implement various data structures such as scripts, and cooperation protocols which together constitute the set of plans of an agent, and the knowledge required to execute these plans

(*b*) a set of tools to monitor the problem-solving and communication among agents during the execution of the system

(*c*) a set of tools to incrementally inspect agents' knowledge bases. For instance, inspecting what type of general knowledge an agent has and what plans constitute its plan library.

Both the simulation and development component function under a common graphical user interface, which also provides tools for visualisation.

## 4.2 Agent modelling

To implement the microscopic traffic simulation model in the DASEDIS environment, the driver–vehicle elements were modelled as agents. The modules of the agent architecture ACTUATORS, SENSORS and MOTIVATION are specific to the traffic domain (see below). At the current stage of the project, the module COMMUNICATION is dormant.

As described above, the COGNITION module controls and monitors the individual agent's behaviour by relating its perception to the execution of appropriate reactions, based on the operational semantics of the BDI approach.

An agent holds beliefs about itself (position, velocity, acceleration lane number, etc.), about other agents (position of the surrounding vehicles, their speed, etc.) and about the environment (e.g. road length, road width). The information about the surrounding vehicles, that is, the vehicles immediately in the front, immediately behind and at the neighbouring positions, are given by the SENSORS. The number of surrounding vehicles depends on the number of lanes being simulated.

At each interval the SENSORS compute in what relation the surrounding vehicles stand with the vehicle. Different states (e.g. 'free-driving', 'following', 'closing-in', 'danger') are characterised based on calculating the difference between a surrounding vehicle's distance and speed to the distance and speed of the vehicle itself. The data provided by the SENSORS are evaluated in the COGNITION module, so that when a state change (i.e. an event) occurs, an appropriate script (i.e. a plan) is chosen and intended for execution.

Scripts describe the stereotypical course of actions to achieve certain desired ends or respond to certain important events in the environment. One desire (goal) is to obtain a certain desired speed. This goal is dedicated by the MOTIVATION module. Scripts contain calls to other scripts or primitive actions such as 'braking' and 'driving'. Primitive actions are executed by the ACTUATORS.

The simplest action simulated is 'driving', that at each interval computes the vehicle's next ($x$ and $y$) coordinates and its velocity in that position. This is a default action that is executed if nothing important has changed in the environment. Whenever a state change occurs, depending on the change, a script is chosen as a response, as will be demonstrated by the following examples.

If the state changes from 'following' to 'closing-in', the script 'reacting-to-closing-in' is chosen. Fig. 3 shows a graphical notation used to specify the script. The first instruction in this script is to attempt to overtake, or if this is not possible, adapt the acceleration to the new situation.

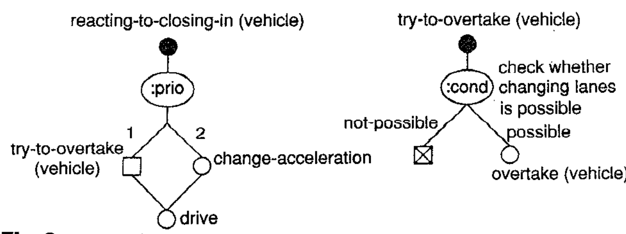


**Fig. 3** *script for reacting to state change to closing-in*

The preconditions for overtaking are that a left lane must exist, and changing to the left lane must be possible with respect to the vehicle behind, driving in the left lane.

A decision to overtake is also based on the driver's goal to reach or hold a desired speed. The driver will only change the lane if he could improve his 'following' situation. The last step is simply the default 'driving' action.

When the driver is in a 'free-driving' state either the behaviour of the vehicle in front is not important (because the distance is significantly large) or there is no vehicle in front. According to the traffic laws (at least in Germany), a vehicle must drive on the right-most lane possible, if the conditions would allow it. In this situation, a vehicle would change to the right lane, if it would not cause 'danger' to the vehicle behind on the right lane, and it would not have to decelerate as a result of the speed of the vehicle in front on the right lane. The corresponding script is shown in Fig. 4.

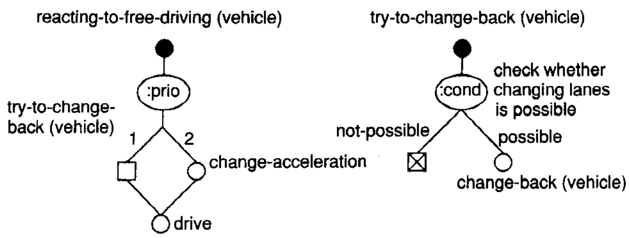


**Fig. 4** *Scripts for reacting to state change free-driving*

Finally, 'following' describes a situation in which the driver perceives a vehicle in front and there is a possibility that at some later stage it would 'close-in' to that vehicle. The driver chooses the last possible moment to change to the left lane. The latest possibility to change to the left lane is when it would not hinder or endanger the vehicle behind on the left lane and can smoothly overtake the vehicle in front. The script is shown in Fig. 5.

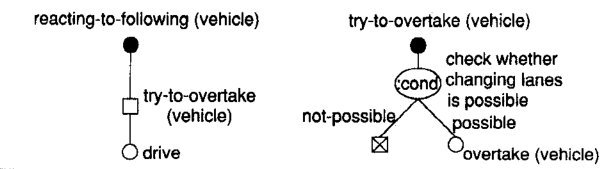


**Fig. 5** *Scripts for reacting to state change 'following'*

A more detailed description of the simulation model and the system described here can be found in [19].

### 4.3 The future

With the recent advances in telecommunications and the provision of vehicles equipped with diverse 'telematics devices', the ability for vehicles to communicate and cooperate promises the modelling of radically new concepts in traffic. As a follow-up to the scenario described above, we are currently working on simulating a priority junction. For this purpose the SENSORS must additionally evaluate the vehicle's position in relation to a traffic intersection. As there are no traffic lights to regulate the traffic flow, this simulation involves modelling how drivers perceive and choose the right moment to cross a junction. To resolve conflicting situations that may cause collisions or bring the traffic into a standstill, the agents communicate to coordinate their actions. Such a situation may arise when the driving directions of two vehicles are perpendicular to one another and both vehicles are equally in a position to cross the junction. By this we will realise a local and distributed traffic control strategy without using traffic light control hardware.

Cooperation between vehicles can improve certain measures in road traffic and consequently increase the throughput of vehicles on the road. Such measures are, for instance, building vehicle platoons by electronic coupling, the regulation of vehicle distances to optimise route capacity, and the co-ordination of velocities and accelerations of vehicles on crossing routes. The high complexity of these scenarios requires tools for modelling and simulation that are flexible to use and easy to enhance and maintain. As the human driving model is an important facet in microscopic simulations, our experiment has shown that the BDI model of the COGNITION module considerably reduces the task of simulating and experimenting with various models of perception, actions, plans and reactions of driver–vehicle elements. Furthermore, by enabling vehicles to communicate and cooperate with one another, AOTs provide the techniques to develop and experiment with radically new scenarios in this domain.

## 5 Simulation of car pooling stations

### 5.1 Overview

Discouraging unnecessary trips or facilitating means by which trips can be more evenly distributed about the day are some alternative suggestions for managing traffic volume. As part of our activities in developing applications for planning and decision making in the area of traffic organisation, we are developing a car pooling simulation platform. Shared use of cars by several customers (commonly referred to as carsharing or carpooling) will become more and more attractive especially in city areas. The idea is to encourage car sharing by convincing users of the benefits they may gain by saving on the maintenance costs incurred when owning a car. Furthermore, customers will demand high-quality customer support and service. Currently the existing carsharing organisations provide cars that are not specially equipped for the particular requirements of a shared use [20].

Among the issues of interest, the car pooling simulation plafform is aimed at facilitating experimentation with different organisational structures, composition of fleet, and technical equipment of cars of car pooling stations. The ultimate goal of these experiments is, on the one hand, to optimise the costs of running the stations, and, on the other hand, to provide an agreeable system in terms of user benefits and contentment.

The scenario that will be described here has been modelled as a multi-agent system and developed using the dMARS multi-agent development environment [21].

In our first experimental simulation model two types of agent are characterised: customer and station (see Fig. 6). The system models and simulates the booking of cars by customers, and the management of cars at stations. Every pooling station has a given number of different car types (car types are distinguished by their type of engine and their dimensions). The simulation system makes use of statistical investigations on customer commuting choices in different areas and their commuting patterns over some time periods (e.g. a week). These provide information such as the frequency of using cars, commuting time intervals and the choice of cars used by different types of customers. Based on this data, customers are divided into different customer groups and further into different customer types. Depending on the structure and the size of these

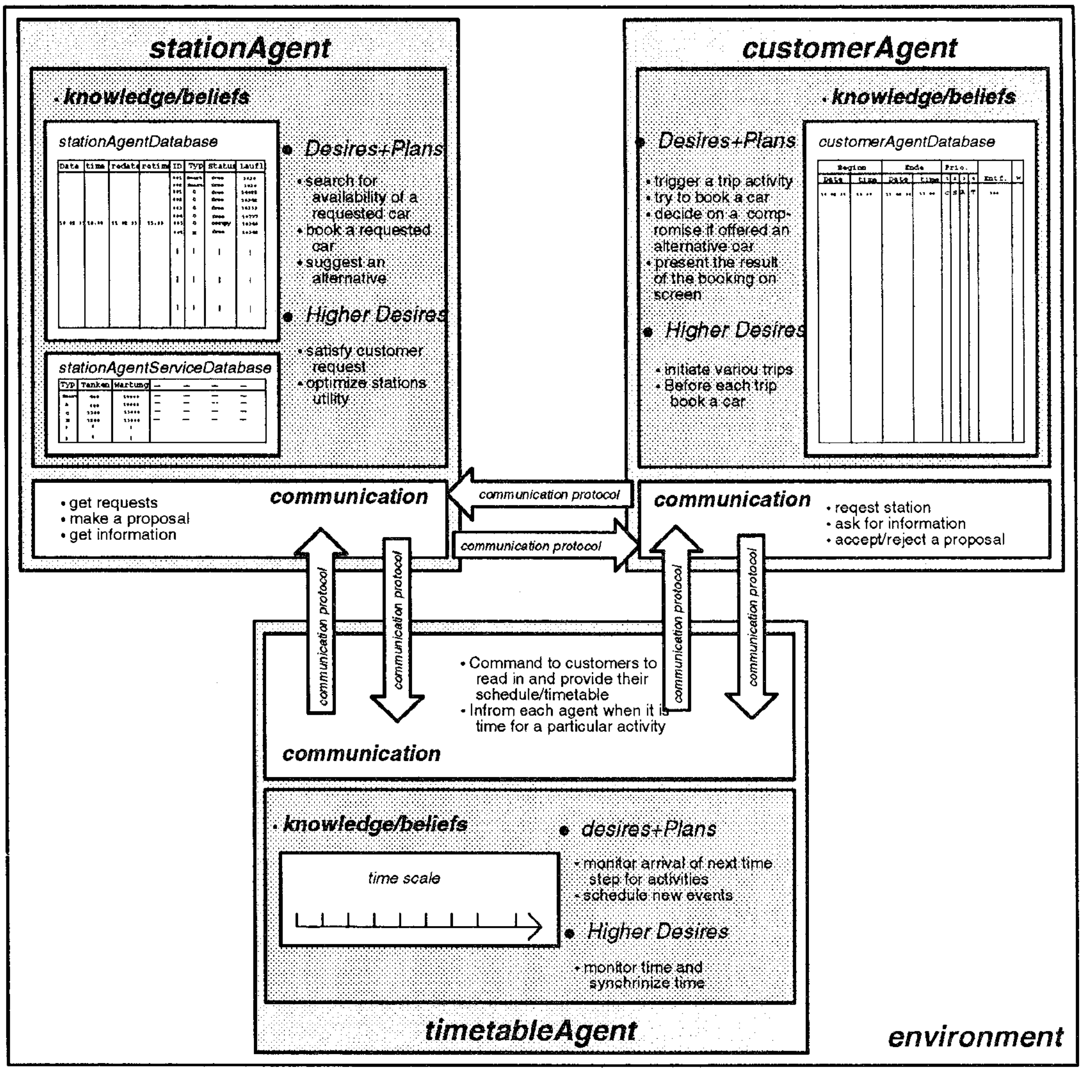


**Fig. 6** *Agents in the carsharing scenario*

groups, we can estimate the type and the number of cars appropriate for a car pooling station in a given area.

The pattern of commuting is structured into a timetable (as shown in Fig. 6). When the simulation system is initiated, each customer agent provides its timetable to the additional timetable agent. The timetable agent is modelled to simulate the motivational events that would cause a customer to adopt a goal to commute at certain time to a certain destination with a certain car. It continuously monitors each customer's timetable against the simulation clock, and according to the time stamp in each table sends a message to the corresponding customer. This message then would cause the customer to adopt a goal in an attempt to book a (desired) car for a desired time interval. On receiving a message from the timetable agent, a customer agent calls up the station and makes a request for a particular type of car and for a particular time and period.

The car pooling station agent maintains a record of cars and their schedule. Cars are considered as occupied not only for the duration they are booked for, but also at service, cleaning and inspection intervals. On receiving a request, the pooling station agent searches its database, and if the customer's wish can be fulfilled the requested car is booked for the customer. Otherwise, if there is a similar car available for the same period, the station agent would propose the booking of that car to the customer. Only if the station and the customer could not arrive at a mutually satisfactory condition is the booking request considered as failed.

## 5.2 Using generic cooperation protocols for car booking dialogues

The booking negotiation between customers and the station requires an expressive representation and a control mechanism for successful dialogues. The technical execution of the dialogue for this negotiation was realised by using generic cooperation protocols [4]. These protocols represent communicative actions of both dialogue participants (see Figs. 7–10). The root node of a protocol represents the sending of the first message in the protocol by the sender and its processing by the receiver. Other nodes are typically calls to other protocols. The colour of a node determines the active agent (sender) on that node, a white node representing the sender and the grey nodes representing the receiver of the very first message in the protocol. Branches represent the permitted alternative paths of dialogue.

The customer initiates the booking dialogue by calling the requesting protocol as follows:

*Requesting ($station_address (Booking $time_interval, $car_type))*

The first message in the protocol is created by the procedure call denoted at the root node. This message is a

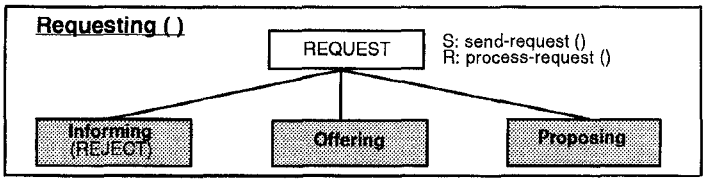


**Fig. 7** *Requesting protocol*

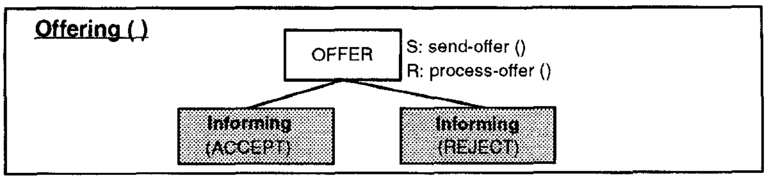


**Fig. 8** *Offering protocol*

request from the customer to the station to book certain desired *$car_type*, at sometime within a certain *$time_interval*. A requested *$time_interval* is a list that consists of a start and an end time, which denotes a time period within which the customer would later (after negotiation) choose an exact booking time period, when accepting an offer (see below).

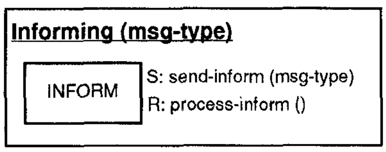


**Fig. 9** *Informing protocol*

By using the parameters of the message content, the pooling station will look into the vehicle schedule, for an appropriate vehicle which is 'free' within the requested *$time_interval*. Based on this result, the station will make one of three responses (see Fig. 7):

(*a*) if the desired *$car_type* is free within the requested *$time_interval*, the order will be considered as possible and offered to the customer (using an offering protocol, Fig. 8

(*b*) if all the cars of the desired type are occupied within the requested *$time_interval* but a similar type of car is free within that period, booking of this car type will be proposed (using the proposing protocol, Fig. 10)

(*c*) if neither the *$car_type* requested nor a similar one can be offered for the requested *$time_interval* the order will be rejected (using an informing protocol, Fig. 9).

When a customer agent receives an offer as a response to its request, it could either accept the offer (using an informing protocol, with ACCEPT message type, including its desired booking period), or reject the offer (using an informing protocol, with REJECT message type). (An informing protocol is used to provide information to the receiver, and it can be specialised by specific message types to indicate the nature of the information. An informing protocol is considered as a leaf node, as it does not require a response.) This would allow the customer to either make a booking now or simply gather information about availability of cars, and if needed, only later make a request to actually book the car.

As with an offer, if the customer receives a proposal to book a similar type of car, the customer may either accept the proposal including the exact desired booking time, or reject it (see Fig. 10). If the customer would want to insist on the desired car type, it may additionally choose to make a second request to the effect that the station later proposes a time interval within which a car of that type would be available. On receiving this request the pooling station would calculate the time difference between the next scheduled return of any car of that type and its availability thereafter, and accordingly make a proposal. The difference between the proposed start time and the customer's original desired start time is acceptable for a customer if it is small in relationship to the duration of booking. If a proposal of the station fails twice, that is, its proposed car type and its proposed time interval were not accepted by the customer, the station will simply reject the third request. This would enforce the dialogue to terminate.

Customers choose other means of transport if unsatisfied with the pooling station. Thus we need to estimate an appropriate car pooling service which, within a reasonable range, provides a satisfactory service. Several runs of each experiment on a particular form of service, produce data for our assessment and analysis. The system keeps a record of unfulfilled requests at each run and this data is used to evaluate the success rate of the service experimented with.

At first the system presented above was conceived as a simulation system, and presently it is used to model different organisational forms of car pooling stations, diverse customer requirements, and vehicle equipments. By introducing some minor modifications, we believe that the system would also be suitable to actually automate the management and control of the operation of a

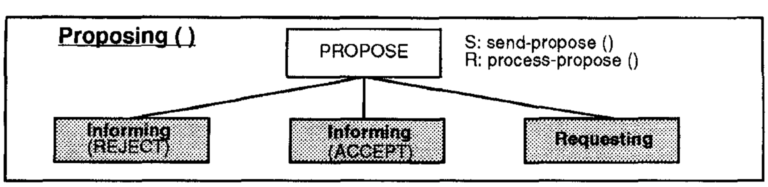


**Fig. 10** *Proposing protocol*

car pooling organisation. This is an important direction in our future work.

## 6 Conclusion and outlook

Agent-oriented techniques provide a new but improved approach to software development. They provide a set of tools and techniques particularly suited to domains that are functionally or geographically distributed into autonomous subsystems, where the subsystems exist in a dynamic environment, and the subsystems have to interact more flexibly. AOTs reduce the complexity in systems design by making available abstraction levels that lend themselves to a more natural way of modelling the problem domain. They enhance the robustness and adaptivity of systems by virtue of increasing the autonomy of subsystems and their self-organisation. The range of applications that can be adequately modelled by AOTs and MASs is perceivably broad.

This paper has promoted the fact that many areas in traffic and transportation domain exhibit the sort of characteristics that AOTs are aimed at: subsystems are geographically and functionally distributed, traffic and transportation is by nature highly dynamic, and much more flexibility is demanded of the existing systems. Agent-oriented techniques as a new advance in information technology provide appropriate set of tools and techniques for making traffic and transportation more efficient, resource-saving and ecological. In this paper we have presented a diverse range of possible applications of MASs in this domain; in particular, integrated intelligent traffic management, enhanced traffic guidance and control systems and improved capacity and resource management.

The MAS applications that we are currently developing in this domain are mainly used for simulation and experimentation. Agent-oriented tools and development environments like DASEDIS [17] and dMARS [21] facilitate rapid prototyping of systems. They provide a natural modularisation of the system in terms of agents that can be executed concurrently or even distributed among different machines. In terms of implementation, they provide abstraction levels such as scripts (or plans) and cooperation protocols that can be conveniently developed, modified and enhanced. In fact the major part of implementation of a system is only a matter of devising appropriate plans and protocols, taking the underlying execution of agents and their components for granted.

The applications described here are still in the state of research. We are convinced that these simulative applications are a prerequisite and a first step in the direction of sophisticated distributed real-world applications in this domain. We see a growing interest and an increasing technological support to use MAS in such applications. There is a large amount of activity in equipping vehicles (being cars, trucks, trains, etc.) with ever more powerful hardware, with mobile communication devices and with a diverse range of intelligent systems. The design and programming of these systems can benefit from the techniques developed in the agent-oriented paradigm, not only to make the systems more autonomous, but also to enable them to interact and cooperate with one another to increase their capabilities and enhance their performance.

## 7 References


1 SHOHAM, Y.: 'Agent-oriented-programming', *Artif. Intell.*, 1993, **60**
2 HADDADI, A., and SUNDERMEYER, K.: 'Belief-desire-intention agent architechtures' *in* O'HARE, G.M.P., and JENNINGS, N. (Eds.): 'Foundations of distributed artificial intelligence' (Wiley, 1996)
3 HADDADI, A.: 'Communication and cooperation in agent systems: a pragmatic theory' (Springer, 1996), Lecture notes in artificial intelligence 1056
4 BURMEISTER, B., HADDADI, A., and SUNDERMEYER, K.: 'Configurable cooperation protocols for multi-agent systems'. *in* CASTELFRANCHI, C., and MÜLLER, J.-P. (Eds.): 'From reaction to cognition'. Fifth European Workshop on *Modelling autonomous agents in a multi-agent world* (MAAMAW-93), (Springer1996), Lecture Notes in Artificial Intelligence 957
5 WALSH, M., and NDOVIE, B.: 'Types of agents, a general taxonomy'. Report DAKE/–/TR-92021.1, University of Keele, 1992
6 NDOVIE, B.: 'Analysis of cooperation in air traffic control (ATC)'. DAKE Centre research report TR-92019.0, Univeristy of Keele, 1992
7 LO, R., and FINDLER, N.V.: 'Empirical studies on distributed planning for air traffic control, Part I: a dynamic hierarchical structure for concurrent distributed control; Part II: the location-centered, cooperative planning system; Part III: experimental results'. TR-88-007, Computer Science Department, Arizona State Univeristy, 1988
8 LJUNGBERG, M., and LUCAS, A.: 'The OASIS air traffic management system'. Technical note, Australian Artificial Intelligence Institute, Melbourne, 1992
9 FALK, J., BORKOWSKI, V., NEUMANN, R., and MERTENS, P.: 'TRAMPAS: a knowledge-based agent system for distributed freight allotment'. in BALAGURUSAMY, E. (Ed.): Proceedings of Fourth International Computing Congress (ICC'93), Hyderabad, India, 1993
10 SANDHOLM, T.: 'An implementation of the contract net protocol based on marginal cost calculations'. Proceedings of the Eleventh National Conference on *Artificial intelligence* (AAAI-93), Washington, DC, 1993
11 FISCHER, K., MÜLLER, J.P., PISCHEL, M., and SCHIER, D.: 'A model for cooperative transportation scheduling'. Proceedings of the First International Conference on *Multi-agent systems*, IC-MAS'95, San Francisco, 1995
12 SUHL, L.: 'Systems for computer-aided production planning in airlines'. Post doctoral thesis (Habilitationsschrift), Technical University of Berlin, 1994
13 HAUGENEDER, H. (Ed.): 'IMAGINE: final project report'. IMAGINE technical report series, IMAGINE consortium, 1994
14 CHEW, T., GILL, A., and LIM, J.: 'Planning the discharging and loading of container ships: a knowledge-based approach' *in* SCHORR and RAPPAPORT, (Eds.): 'Innovative applications of artificial intelligence' (AAAI Press, 1989)
15 FRITZSCHE, H.-TH.: 'A model for traffic simulation', *Traffic Eng. Control*, 1994, **5**
16 BURMEISTER, B., and SUNDERMEYER, K.: 'Cooperative problem-solving guided by intentions and perception' *in* WERNER, E., and DEMAZEAU, Y., (Eds.): Decentralized A.I.3, Third European Workshop on *Modelling autonomous agents in a multi-agent world*, MAAMAW-91, (North Holland, 1992)
17 BURMEISTER, B.: 'DASEDIS – Eine Entwicklungsumgebung zum Agenten-orientierten Programmieren (A development environment for agent-oriented programming)' *in* MÜLLER, H.J. (Ed.): 'Verteilte Künstliche Intelligenz' (BI-Wissenschaftsverlag, 1993)
18 HADDADI, A.: 'A hybrid architecture for multi-agent systems'. Proceedings of the Special Interest Group on Cooperating knowledge based systems, CKBS 93, DAKE Centre, Keele, 1993,
19 BURMEISTER, B., DOORMANN, J., and MATYLIS, G.: 'Agent-oriented traffic simulation', *SCS Trans. Simulation*, 1997
20 'Delta-report, supplement 1995/96'. Internal research report, Daimler-Benz AG Research and Technology, 1995, (in German)
21 KINNY, D.: 'The distributed multi-agent reasoning system architecture and language specification'. Australian Artificial Intelligence Institute, Melbourne, 1993