UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

ROSALDO JOSÉ FERNANDES ROSSETTI

A BDI–based approach for the assessment of drivers' decision–making in commuter scenarios

Thesis presented in partial fulfillment
of the requirements for the degree of
Doutor em Ciência da Computação

Prof. Dr. Sergio Bampi
Advisor

Prof. Dr. Helena Beatriz Bettella Cybis
Coadvisor

Porto Alegre, November 2002

CIP – CATALOGING-IN-PUBLICATION

Rossetti, Rosaldo José Fernandes

A BDI–based approach for the assessment of drivers' decision–making in commuter scenarios / Rosaldo José Fernandes Rossetti. – Porto Alegre: PPGC da UFRGS, 2002.

187 f.: il.

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2002. Advisor: Sergio Bampi; Coadvisor: Helena Beatriz Bettella Cybis.

1. Multi-agent systems. 2. BDI architecture. 3. Decision-making. 4. Intelligent transportation systems. 5. Traffic modelling. 6. Microscopic traffic simulation. I. Bampi, Sergio. II. Cybis, Helena Beatriz Bettella. III. Título.

ACKNOWLEDGMENTS

This work is the result of a great effort and a paramount investment on a career project. However, bringing it into reality was only possible with the contribution, support, and friendship of many, many people.

To my supervisor, Dr. Sergio Bampi, a grateful and especial thanks. He has been a key presence from my earliest days at UFRGS. I have definitely learnt a lot from his research experience and professional character. Being always accessible and providing his students with a paternal friendship, constant support, encouragement, and patience are remarkable characteristics of him. Dr. Helena Cybis, my co-supervisor, has also been a very important friend. Her expertise helped me to better understand the complex nature of the application domain and fed interesting discussions on the course of my work.

During the doctoral programme, I had the opportunity to be a research student at the University of Leeds and to interact with the Network Modelling Group of the Institute for Transport Studies. Dr. Dirck Van Vliet and Dr. Honghui Liu supervised my work there and helped me to build the roadmap to my thesis proposal. That was a pleasant experience I will always remember. To Dr. Van Vliet and Dr. Liu all my gratitude for their interest in my work and encouragement that made my research internship in Leeds a successful and enriching experience of life. Also, my office-mates Fang, Clare, and Il-ho made me feel at home from the beginning, providing me with an excellent work atmosphere.

My sincere thanks to Alexandre Martins, for his friendship and pleasant discussions on all the subjects related to data modelling, and also for his invaluable tips on software development with Java. Dr. Rafael Bordini and Dr. Ana Bazzan have also contributed with invaluable comments and fruitful discussions on the soundness of the approach proposed. I gratefully thank Rodrigo Machado, for his help on the implementation of the first simulation framework prototype. He presents wonderful programming skills and great potential to be explored.

I could not forget Eliana Senna, my aunt Neca, who gave me all the support that I needed while settling myself in Leeds. She became a very special and important friend since then. I would also like to express my gratitude to Isidório, Jutta and Rafael Gamba for their kind assistance, at all times, that turned Porto Alegre into my second home.

In special, I dedicate this work to my family, my parents Adroaldo and Rosália, my brother Júnior, and my sister Lialda, whose emotional support, encouragement, and pa-

tience made this work possible. My father, a remarkable man, was the first scientist that has inspired my research career. And also a special dedication goes to Susana, my sweetheart, for her love, support, and enthusiasm.

I gratefully acknowledge the financial support from CNPq and CAPES, the Brazilian agencies for R&D that partially provided the funds to cover the expenses of this work, both in Porto Alegre and in Leeds. The financial supports from FAPERGS, Insituto de Informática of UFRGS, and the Institute for Transport Studies of the University of Leeds were also important and helped me with the publication of partial results of this thesis at workshops and conferences.

I believe this is one of the most difficult parts to write after finishing such a hard and extensive work. Even if I wanted to mention everyone's name here, I would certainly fail in doing so. Therefore, I will not even try it, but I would like to acknowledge the support and encouragement I received from so many people, either directly or indirectly, from the beginning of my research life to the final results of my doctoral thesis. Thank God, for the gift of life! And, to all of you, my grateful thanks.

The question of whether computers can think is like the question of whether submarines can swim

— Edsger Wybe Dijkstra (1930-2002)

CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS

3APL	An Abstract Agent Programming Language
AI	Artificial Intelligence
AICC	Autonomous Intelligent Cruise Control
API	Application Programming Interface
APTS	Advanced Public Transportation Systems
ARTS	Advanced Rural Transportation Systems
ATIS	Advanced Traveller Information Systems
ATIS	Advanced Traveller Information Systems
ATMS	Advanced Traffic Management Systems
AVCS	Advanced Vehicle Control Systems
BDI	Beliefs, Desires, and Intentions
BRF	Belief Revision Function
Bee-gent	Bonding and Encapsulation Enhancement Agent
CA	Cellular Automata
CAPES	*Coordenação de Aperfeiçoamento de Pessoal de Nível Superior*
CE	Cognitive Emergence
CNPq	*Conselho Nacional de Desenvolvimento Científico e Tecnológico*
CPU	Central Processing Unit
CTL*	Computational Tree Logic
CVO	Commercial Vehicle Operations
DAI	Distributed Artificial Intelligence
DIS	Driver Information Systems
dMARS	Distributed Multi-Agent Reasoning System
DRACULA	Dynamic Route Assignment Combining User Learning and microsimulAtion
DRG	Dynamic Route Guidance

DRGS	Dynamic Route Guidance Systems
DTC	Design-To-Criteria
DVO	driver-vehicle objects
ELP	Logic Programming extended with explicit negation
FAPERGS	*Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul*
FLOWSIM	Fuzzy LOgic enhanced motorWay traffic SImulation Model
GPSS	General Purpose Simulation System
IT	Intelligent Transportation
ITS	Institute for Transport Studies
ITS	Intelligent Transportation Systems
IVHS	Intelligent Vehicles and Highway Systems
JAL	JACK Agent Language
MADAM	Multi-Agent DemAnd Model
MAS	Multi-Agent Systems
OD	Origin–Destination
PPGC	*Programa de Pós-Graduação em Computação*
PRS	Procedural Reasoning System
R&D	Research and Development
SeSAm	Shell for Simulated Agent Systems
SITRAS	Simulation of Intelligent TRAnsport Systems
TCS	Traffic Control System
TIS	Traveller Information Systems
TRYSA2	TRYS Autonomous Agents
UFRGS	*Universidade Federal do Rio Grande do Sul*
UMPRS	University of Michigan’s implementation of PRS
UTC	Urban Traffic Control
VIM	Visual Interactive Modelling
VIS	Visual Interactive Simulation
VMS	Variable Message Signs
WFSX	Well-Founded Semantics extended for explicit negation

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The rapid growth of urban areas has a significant impact on traffic and transportation systems. New management policies and planning strategies are clearly necessary to cope with the more than ever limited capacity of existing road networks. The concept of Intelligent Transportation System (ITS) arises in this scenario; rather than attempting to increase road capacity by means of physical modifications to the infrastructure, the premise of ITS relies on the use of advanced communication and computer technologies to handle today's traffic and transportation facilities. Influencing users' behaviour patterns is a challenge that has stimulated much research in the ITS field, where human factors start gaining great importance to modelling, simulating, and assessing such an innovative approach.

This work is aimed at using Multi-agent Systems (MAS) to represent the traffic and transportation systems in the light of the new performance measures brought about by ITS technologies. Agent features have good potentialities to represent those components of a system that are geographically and functionally distributed, such as most components in traffic and transportation. A BDI (beliefs, desires, and intentions) architecture is presented as an alternative to traditional models used to represent the driver behaviour within microscopic simulation allowing for an explicit representation of users' mental states.

Basic concepts of ITS and MAS are presented, as well as some application examples related to the subject. This has motivated the extension of an existing microscopic simulation framework to incorporate MAS features to enhance the representation of drivers. This way demand is generated from a population of agents as the result of their decisions on route and departure time, on a daily basis. The extended simulation model that now supports the interaction of BDI driver agents was effectively implemented, and different experiments were performed to test this approach in commuter scenarios.

MAS provides a process-driven approach that fosters the easy construction of modular, robust, and scalable models, characteristics that lack in former result-driven approaches. Its abstraction premises allow for a closer association between the model and its practical implementation. Uncertainty and variability are addressed in a straightforward manner, as an easier representation of humanlike behaviours within the driver structure is provided by cognitive architectures, such as the BDI approach used in this work. This way MAS extends microscopic simulation of traffic to better address the complexity inherent in ITS technologies.

Keywords: Multi-agent systems, BDI architecture, decision-making, intelligent transportation systems, traffic modelling, microscopic traffic simulation.

Uma abordagem baseada em modelos BDI para avaliação do processo de decisão de motoristas no tráfego urbano

RESUMO

O rápido crescimento das regiões urbanas tem impacto significativo nos sistemas de tráfego e transportes. Políticas de gerenciamento e estratégias de planejamento alternativas são claramente necessárias para o tratamento da capacidade limitada, e cada vez mais deficitária, das redes viárias. O conceito de Sistemas Inteligentes de Transportes (ITS) surge neste cenário; mais do que procurar aumentar a capacidade por meio de modificações físicas na infraestrutura, sua premissa baseia-se na utilização de tecnologias avançadas de comunicação e computação para melhor gerir os recursos de tráfego e transportes atuais. Influenciar o padrão do comportamento dos usuários é um desafio que tem estimulado muita pesquisa na área de ITS, onde fatores humanos passam a ter grande importância na modelagem, simulação e avaliação dessa abordagem inovadora.

Este trabalho tem como foco a utilização de Sistemas Multiagentes (MAS) na representação dos sistemas de tráfego e transporte, com base nas novas medidas de desempenho impostas pelas tecnologias ITS. As características de agentes têm grande potencial para representar componentes geográfica e funcionalmente distribuídos, como a maioria dos elementos no domínio da aplicação. Uma arquitetura BDI (*beliefs*, *desires*, *intentions*) é apresentada como alternativa a modelos tradicionais, usados para representar o comportamento do motorista em simulação microscópica, considerando-se a representação explícita dos estados mentais dos usuários.

Os conceitos básicos de ITS e MAS são apresentados, assim como exemplos de aplicações relacionados com o tema do trabalho. Esta foi a motivação para a extensão de um simulador microscópico existente, no sentido de incorporar as características dos MAS para melhorar a representação dos motoristas. Assim, a demanda é gerada a partir de uma população de agentes, resultando da decisão sobre a rota e o tempo de partida ao longo de vários dias. O modelo estendido, que passa a suportar a interação de motoristas BDI, foi efetivamente implementado e foram executados diferentes experimentos para testar a abordagem em cenários de tráfego urbano.

MAS permite uma abordagem direcionada a processos que facilita a construção de representações modulares, robustas, e extensíveis, características pouco presentes em abordagens voltadas ao resultado. Suas premissas de abstração permitem uma associação direta entre modelo e implementação. Incerteza e variabilidade são assim tratadas de maneira mais intuitiva, uma vez que arquiteturas cognitivas permitem uma fácil representação do comportamento humano na estrutura do motorista. Desta forma, MAS estende a simulação microscópica de tráfego no sentido de melhor representar a complexidade inerente às tecnologias ITS.

Palavras-chave: sistemas multiagentes, arquitetura BDI, processo de decisão, sistemas inteligentes de transporte, modelagem de tráfego, simulação microscópica de tráfego.

1 INTRODUCTION

1.1 Overview

The rapid growth of urban areas has deserved special attention from scientific and technical communities over the last years. New management policies and planning strategies are necessary to tackle the problems that arise from today's urban scenarios. Not surprisingly, transportation and traffic systems are objects of concern in many of these problems as they play an important and indispensable role in contemporary society. However, road infrastructures are no longer sufficient to meet the increasing demand and traffic congestion is frequently encountered in most commuters' journeys. This implies considerable economic, social, and environmental losses, which should be minimised. Physical modification to the road infrastructure is no longer the best alternative to tackle such a problem. Besides the high cost of implementation, they cause disruptions and can damage the environment. Alternatively, some efforts have been identified in order to increase road capacity by improving the efficiency of traffic control systems. Although such efforts have addressed some of the problems arising from traffic jams, they are not considered to be a lasting solution.

Nonetheless, researchers still seek alternative means to cope with traffic and transportation specificities. The first attempts at improving road capacity have relied on dealing with the static part of the system, namely the road infrastructure and control systems. However, another approach has been experienced, which, on the other hand, relies on maximising the use of the actual road capacity through directly influencing users' behaviour patterns. The concept of Intelligent Transportation System (ITS) arises in this scenario. The growing advances in communication, as well as in computer technologies have encouraged the use of such systems to tackle problems in the field of traffic and transportation engineering.

The underlying concept of ITS is to ensure productivity and efficiency by making better use of existing systems. It is mainly concerned with the application of distributed solutions; each of which deals with specific issues of users' needs on an individual basis. Modifying travel patterns through directly influencing user behaviour can be seen as the main premise within these technologies. Autonomy and intelligence are two concepts that should be present in such systems. Furthermore, integrating all factors, both dynamic and static, which can somehow effect the traffic flow is also central to ITS. So, computer and communication technologies become key ingredients in the process of implementing these systems. In this way, all components are expected to work together in a co-operative environment to maximise the overall efficiency of traffic and transportation.

Models used to represent traffic systems need to better handle characteristics such as the human behaviour and the temporal–dependent nature, which are brought to the user decision level. Such a need forces the use of a lower-level abstraction to describe traffic systems that, on the other hand, leads to an increasing complexity. Thus, practitioners will need systems capable of dealing with the new performance measures brought about by the deployment of such intelligent and adaptable technologies. Human behaviour becomes a variable of huge importance to be coped with, as it plays the central role in assessing public acceptance for ITS. Owing the use of simplified approaches, some traditional models fail in representing these complex scenarios. Therefore, many efforts have been identified in order either to elaborate new models from scratch or to adapt traditional ones to meet the need for representing ITS requirements.

Since representing human behaviour in models for simulation purpose becomes imperative, agent-based techniques could be considered as very appropriate approaches to represent transportation and traffic systems. Multi-agent System (MAS) is a sub-field of Distributed Artificial Intelligence (DAI), which has deserved an increasing interest in the last decade. The rapid evolution in the available computational resources, both in hardware and in software, which support a widely physically distributed computing environment, has contributed to that. Additionally, the increasing demand for suitable tools to represent the complexity inherent in some application domains has motivated much research on MAS.

The concept of MAS can be basically seen as a modelling approach devised to represent systems whose entities, coined agents, exhibit intelligence, autonomy, and some degree of interaction, both with one another and with the environment. The abstraction approach of MAS consists of representing a system by multiple agents that exist in a common environment and interact in order to achieve specific goals. Thus, an agent can be any entity capable of perceiving facts through sensors and acting upon the environment through effectors. Some degree of interactions will also imply the presence of communication capabilities. Furthermore, some agent-based approaches present powerful tools for representing mental attitudes, such as beliefs, desires, intentions, emotions, and others, which are intrinsic in human beings. Agent-based models are ideal to deal with entities that are geographically and functionally distributed, and a good ability of representing entity ontology ensures agent-based models' scalability and robustness, which are desirable characteristics for ITS models.

1.2 Motivation

In traffic modelling it is possible to distinguish between two approaches, namely the macroscopic and the microscopic point of views. Macroscopic models rely on flow theory rather than representing individual particles and are usually based on assignment algorithms. A rigid structure, an origin–destination (OD) matrix, is used to represent trip distributions between all possible origins and destinations. The assignment is an iterative process that seeks a steady state; in other words, a state such that the average travel time for each link cannot be minimised by assigning trips to other routes between certain origin and destination. To the contrary, microscopic models rely on an individual representation of each driver–vehicle particle, which moves throughout the network. OD matrices are equally used in microscopic models to represent travel patterns between different zones within the traffic system.

In this scope of traffic representation, DRACULA is a tool that implements a microscopic model and will deserve a special attention in this thesis. Drivers are extracted from a population of potential drivers, which is estimated from an OD matrix and will compose the demand for travel on a certain day. The demand of a day, for instance day k, is formed of drivers from the population that have effectively decided to make a trip on that day. Therefore, in a hundred-day simulation, for instance, different configurations for the demand can be set on each day. Contrary to models based on a fixed trip matrix, DRACULA is based on a variable demand, where uncertainty and variability are the underlying concepts. So, the steady state can be identified within a distribution of probability rather than being identified by a single value of flow. Drivers make their route choices accounting for past experiences, which are stored in driver's memory in terms of travel time for each link; it is then a quantitative assessment. However, such an approach does not correspond to reality as in most of the cases decisions are made with regard to qualitative aspects that drivers believe to be held within the system.

In order to illustrate the motivations for this work, one may imagine a situation in a daily life of a traffic network user; in other words, certain driver named Joe. One of Joe's friends, say Mary, invites him to come over for a happy hour on day k, for instance. However, Joe is not so familiar with the streets in Mary's neighbourhood and does not know for sure how he could get there. To cope with this possible limitation, Joe decides to ask for some help. He logs onto the Internet and accesses a traffic information system application.

Knowing where to get to and estimating the time he will need to perform the journey, Joe can now plan his trip. Thus, he selects a course of actions that will result in his objective. He chooses a time to leave and a route to follow so that he can arrive about the time he has committed to Mary. Once he has planned his journey, he can execute it. While Joe has not found any obstacle within it, he can keep executing his original plan. However, he just finds that certain road on his itinerary is interrupted. As Joe is not able to drive through that road any more, he has to reconsider his options and find another alternative route to get to Mary's house. Therefore, Joe abandons his original plan and starts executing the new one. In another point in time, Joe just realises that he has not got enough petrol to get to his destination. Albeit he knows he will probably be late, he prioritises changing his plans once again in order to stop somewhere to get more petrol. After doing so, he is finally able to arrive at Mary's home.

Analysing the simple story above, it is possible to identify some characteristics in Joe's behaviour that are very interesting to be featured in the driver representation of microscopic simulation models.

- the driver presents *autonomy* as he can identify on his own what his objectives are and which actions he needs to perform to yield the expecting results;

- through his *social ability*, as in the example above, a driver can ask for some help in order to ease the execution of his actions, for instance, by contacting a service provider such as a traveller information centre;

- responding to traffic signals and breaking in order to avoid colliding with others are some well known examples where *reactivity* is realised. Nonetheless, perceiving an interruption and adopting another route can also be modelled as a reaction by the driver;

- the driver presents *adaptability* in the sense that he must be able to reconsider his options and to adopt another strategy in order to accomplish his goals in the case the original plan becomes inadequate;

- a driver also must be able to prioritise the execution of an action to the detriment of his original plans, for instance, arriving later after adopting another route that is more convenient owing some other reason. In this sense, a driver presents *proactivity*.

Considering these very human characteristics could be extremely useful in tools aimed at assessing drivers' behaviour. However, it is not easy to represent such features within existing models; their structures are so rigid that scalability is a feature they lack. Bearing this possible limitation in mind, modelling drivers as entities fully endowed with reasoning abilities is the main motivation for this work.

1.3 Goals

This work is aimed at relevant aspects of the potentials of agent-based techniques in modelling human behaviour within Intelligent Transportation Systems. Coping with drivers' decision-making in simulation frameworks is the subject of main concern. Devising a computational tool that allows for modelling, implementing, and analysing this decision-making is the challenge for this work. This effort seeks to ease the assessment of new measures brought about by the deployment of ITS technologies.

However, the number of ITS scenarios are too vast and there are so many situations drivers can be involved with and which demands their making of a variety of decisions. Thus, the focus of this thesis is given to commuter scenarios that involve decisions such as which route to take and what time to depart. A BDI-based architecture is proposed as the underlying structure of the driver agent, which may serve to a wide range of purposes within microscopic simulations. In order to test and demonstrate the feasibility of the approach, a framework is proposed on the basis of an extension to the DRACULA model, where demand can be characterised, evaluated, and analysed on the basis of a population of driver agents.

1.4 Methodology

The methodology adopted in order to reach the aim of the thesis and to accomplish its goals is basically composed of the following parts.

- to describe the application domain by means of agents and their features. In order to accomplish this, the basics of ITS and MAS are briefly presented and potential connections between these two fields are discussed. Examples are gathered from the literature, which illustrate the research efforts toward using agent-based techniques as a designing tool for simulation and evaluation environments. Although a number of possible applications have been identified, this step is focused on representing demand as a result of decision-making in two levels: route and departure time;

- to devise a cognitive agent model to serve as the underlying structure for modelling the decision-making process. The ease for straightforwardly describing the concepts of mental attitudes, such as beliefs, desires, and intentions, as well as their relation within the cognitive process motivated the adoption of a BDI-based approach;

- to choose a BDI theory capable of supporting the practical implementation of the cognitive driver agent. Practical implementation of BDI architectures has defied researchers within the MAS research community. A great challenge with this concern has been how to overcome the difficulties of applying such an approach to societies composed of a huge number of cognitive participants, mainly those sorts with a stringent time-dependent nature. Some scientific works have been focused on addressing such drawbacks and relatively recent commercial agent development tools are featured with BDI models to support practical implementation;

- to design and to implement the cognitive driver agent architecture with the necessary mechanisms to allow interaction with others and with the environment. The agent architecture herein proposed, though designed as the framework to base the BDI reasoning, is also intended to support integration within different levels for different purposes in ITS;

- to design and to implement a microscopic simulation environment aimed at testing the feasibility of the approach proposed. Rather than developing it from scratch, the environment is implemented on the basis of an extension to an existing microscopic simulation model;

- to specify and to carry out simulation experiments within the framework implemented. Commuter scenarios are suggested to test the generation of demand as a result of decisions rationally made by the BDI agents that compose the population of drivers.

1.5 Challenging issues

One main idea herein presented is to see Intelligent Transportation Systems as a ground where theories from Computer Science can be not only applied but also generated and tested on an integrated basis. The different ways in which information can be handled and processed within ITS scenarios inspire the development and deployment of a huge range of technologies. This suggests that ITS can be seen from different perspectives, for example, from software to hardware, from local solutions to the use of distributed computer systems, and so forth. Modelling approaches also differ from one another and are very likely to be affected by different levels of granularity, for instance, dealing with the system as entities in an agent society or modelling the details of an embedded system on its own rights. Moreover, ITS can also be classified as both soft and hard real-time systems, depending on the several types of time constraints that are identified. And endowing their components with autonomy, adaptability, reactivity, and pro-activity, so that they can be dealt altogether with as Intelligent Systems has turned ITS into an interesting ground for Computer Science.

Although the use of agents seems to be very adequate to cope with representing specificities of today's traffic and transportation scenarios, linking ITS and MAS together poses challenging issues.

- how to handle a large number of interacting heterogeneous elements, with different goals, skills, reasoning capabilities, and degree of autonomy;
- how to address different levels of complexity, both with respect to inter- and to intra-element interactions;
- as the driver becomes an important actor within ITS, how to cope with the complex nature of humanlike behaviour, especially with that concerning reasoning and decision-making. This very feature of the driver is also affected by whom it is in-

teracting with, which implies it needs to dynamically switch between reactive and cognitive behaviours, in different levels;

- as there may potentially be a huge number of entities, both co-operation and competition may arise from such a heterogeneous society. So, conflicting situations are very likely to happen and must be addressed in either relation. In this sense, communication and co-ordination need to be represented, as well;
- how to represent the information flow throughout the system and how such information is interpreted by heterogeneous entities;
- the environment is very likely to be affected by the behaviour of several of the traffic components, and being capable of observing and interpreting its instant state is one main source of information. How to represent and parameterise the traffic environment so that involving components are able to perceive from and act upon it.

The modular structure of a framework relying on agent-based techniques seems to suit the demands for testing and assessing different rational and social behaviours of ITS components. The ease to represent communication, interaction, reasoning, decision-making, planning, and learning, for instance, has motivated and reinforced the thought of MAS as the starting ground where practitioners, engineers, and scientists will devise, test, and apply today's traffic and transportation technologies.

1.6 Structure of the thesis

The remaining of this thesis is organised as follows. An overview of Intelligent Transportation Systems is presented in Chapter 2, which gives special attention to Advanced Traveller Information Systems (ATIS) as an important exogenous source of information. In Chapter 3, the basic concepts of Multi-agent Systems (MAS) are introduced, as well as some examples of agent-based techniques applied to the field of traffic and transportation engineering. Mental attitudes such as beliefs, desires, and intentions, are the subjects of main interest in Chapter 4, and a model to represent intentional drivers within microscopic simulation is presented in Chapter 5. An extension to an existing microscopic simulation model, DRACULA, is proposed in Chapter 6. Such an extended framework aims at incorporating agent features in order to address the lack of suitable representations for decision-making as an important instrument for demand formation. It also presents and discusses the results obtained by the implementation and simulation of the MAS model devised. Conclusions are drawn in Chapter 7, as are presented further developments and some proposals for future works. The OD matrix and the network description files used in the experimental framework are made available in Appendix A and Appendix B, respectively, whereas Appendix C is used to provide a brief explanation on how incidents are modelled in DRACULA. An extended abstract of the thesis, in Portuguese, is presented in Appendix D.

2 INTELLIGENT TRANSPORTATION SYSTEMS

2.1 Overview

The increasing demand in urban areas has awarded the attention of important research groups around the world. Notably, physically modifying traffic infrastructures is no longer the best way to improve services. Such an initiative leads to serious economics, social, and environmental problems. Other alternatives have been investigated over the years in order to optimise the limited capacity of traffic networks, for instance, through improving control systems. Another way relies on providing travellers with relevant information in order to aid their decisions and, consequently, to influence their travel patterns. Intelligent Transportation Systems (ITS) arise then from the growing advances in communication and in computing to overcome the more than ever limited capacity of traffic and transportation resources. Instead of intervening physically, these systems are expected to ensure productivity and efficiency by making better use of existing resources. In general, ITS has the potential to provide travellers with up-to-date information suited to their particular requirements through a wide variety of mechanisms and technologies (CHATTERJEE; MCDONALD, 1999).

So, creating intelligent systems that are able to communicate and to co-operate with one another toward the improvement of capacity usage is at the heart of ITS (ADLER; BLUE, 1998). This has motivated the development and widespread use of such technologies. They require the collection of large amounts of data, turn them into 'intelligence' and then convey relevant and timely information to managers and users (BARFIELD et al., 1989). It is also important to bear in mind that the term coined to describe the application of these systems is not restricted to ground traffic and transport. ITS involves all transport modes, all transport users, and every kind of vehicle, as well as their management and control. It reflects the recognition of all transportation components, both passive and active, namely the environment, users, and service providers.

It is important to remember that more than being a topic of research, ITS is of paramount importance and has great influence on peoples' lives. Therefore, the development of this applied knowledge field has attracted the interest of both the scientific research community and different sectors of the society (BOCHNER, 1998).

2.2 Brief history

Although the term has just been coined, ITS applications can be dated to the early 1970s. The first aim was at addressing control system issues through developing more responsive traffic signals. This approach relied on responsiveness to meet the highly varying demand in urban areas. Between the late 1980s and mid 1990s, in the United States, such applied technologies were designated by Intelligent Vehicle and Highway Systems (IVHS), as earliest examples in that country were mostly related to highway and expressway scenarios. Notably, in many parts of the world, namely the United States, United Kingdom, Australia, and Japan, the road and vehicle industries have been taking a leading role both in developing and promoting ITS (GARRETT, 1998).

From the late 1980s on, efforts to tackle the growing problem stemming from urban traffic congestion have been devoted to providing travellers with timely and suited information. The main assumption of this approach is concerned with increasing traffic efficiency by means of influencing drivers' behaviour patterns (BARFIELD et al., 1989). Hulse et al. (1998) give a broad view of how such technologies can be implemented and used toward improving traffic systems.

Intelligent technologies applied to the field of traffic and transportation engineering have experienced a great evolution in the last decade. Although much has been done, ITS is still in its earliest ages and there are still great potentialities to be exploited (BOCHNER, 1998).

2.3 Advantages of ITS

The Intelligent Transportation Systems have great potential to provide both users and service providers with a wide range of benefits (BOCHNER, 1998). Some of them can be already perceived, whereas others are expected to become reality in the near future (BOCHNER, 1998; GARRETT, 1998).

- *safety* is one of the most important issues that concern ITS. The increasing number of vehicles moving throughout a network has contributed to hazardous and dangerous scenarios, which must be forecasted and prevented. Automatic control of vehicles could promote a high level of safety, as could computer-aided driving to help individuals to take the right decisions at hazardous situations, especially for inexperienced or less skilled drivers;

- the *environment* is expected to benefit from ITS by means of two main factors, namely the decreasing demand and more efficient vehicles. The former can be achieved by pricing policies, both on road usage and on vehicles' occupancy, boosting public transport modes. On the other hand, non-pollutant fuels and alternative works engines would contribute to minimise CO_2 emissions. Also, some other onboard facilities are being investigated toward improving interactions both with other vehicles and with network resources. This way, road capacity might be enhanced as platoons are formed by coupling mechanisms and electronic control of travel speed, in response to roads' velocity limit;

- Although today's ITS technologies are not accessible to the whole population, as yet, *equity* is to be achieved as they prove to enable high quality, cost effective, reliable and efficient information both to private and to public transport users, on either individual or collective basis. It is also important to bear in mind that there are efforts by government and industry toward consensually achieving ITS standards;

- in a broad perspective safe and easy *mobility*, for both people and goods, is all ITS is concerned with. Optimising transport modes, routes, and travel time, for instance, are the ways provided by ITS to overcome lack of mobility. Quite related to the issue of equity, above, mobility must contemplate groups of disabled and elderly people, as well;

- ITS technologies are likely to improve the *efficiency* of road networks. They seek to avoid, or at least to postpone, the need for physical modifications to infrastructures. The specificities of such intelligent systems will also impose specific operational and maintenance requirements, which should be taken into account, as well;

- *business opportunities* are likely to emerge, as ITS originates new trends for consumers and service providers. Delivery, tourism, restaurants, hotels, information hosting and provision, and others will profit from efficient communication and data processing granted by ITS.

2.4 Basic architecture for ITS

The Intelligent Transportation Systems basically rely on the application of distributed solutions, which address specific issues of traffic and transportation. And by working together on a co-operative basis, they seek to maximise the overall efficiency of the system (BARFIELD et al., 1989). According to Mast (1998), a basic structure of an ITS would comprise the following modules, whose interactions are illustrated in Figure 2.1.

- the *Advanced Traveller Information Systems* (ATIS) have emerged as a key component of ITS. They include a number of facilities to provide drivers with real-time and in-vehicle information, suited to their needs on either individual or collective basis. Such information can be related to navigation and route guidance, motorist services, road signing, and hazard warnings, for instance;

- the concept of *Advanced Vehicle Control Systems* (AVCS) refers to the mechanisms that aid individuals in driving tasks, particularly in either emergency or hazardous situations. In more audacious approaches, such systems could even take over some or all of the driver functions;

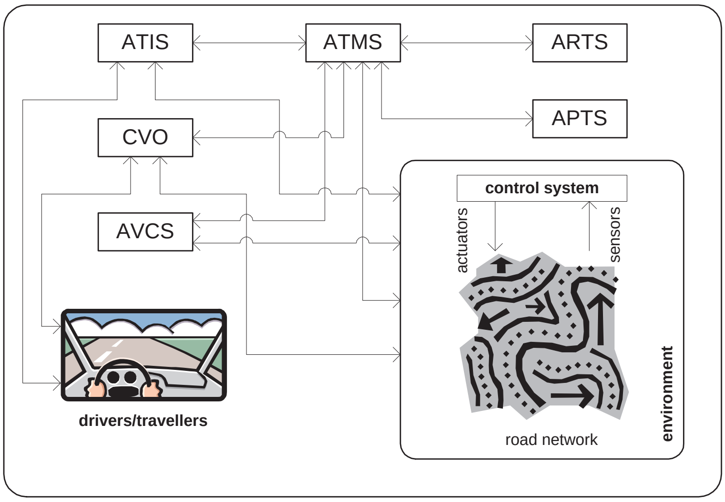

Figure 2.1: Intelligent Transportation Systems.

- *Commercial Vehicle Operations* (CVO) comprise the ITS mechanisms to address special needs of commercial roadway transportation, including vehicle identification, location tracking, weight-in-motion, clearance sensing, and record keeping, to mention some. This way, ITS represents a great deal in turning the high costs involved in such technologies into worthwhile outcomes, both to transportation companies and to drivers;

- the *Advanced Traffic Management Systems* (ATMS) are responsible for monitoring, controlling, and managing traffic on streets and higher order roads. They play a central role within the architecture. In general terms their aim is to reduce congestion, which is basically accomplished trough controlling and constraining vehicles' route diversion. Some technologies encompassed in this module include route guidance, automated traffic signal timing, variable message signs (VMS), and priority control systems;

- *Advanced Rural Transportation Systems* (ARTS) are specially suited to the specificities of traffic and transport in rural areas. In large countries, these technologies are of invaluable help to the countryside communities. Emergency notification and response, vehicle location via GPS, and traveller information are some facilities provided;

- on the other hand, *Advanced Public Transportation Systems* (APTS) are concerned with public transportation within large urban areas. APTS enhances the effectiveness, attractiveness, and the economics of public transportation and includes, among others, fleet management, automated fare collection, and real-time information systems.

Although each of these modules can be defined on its own right, they must rely on a common and integrated framework. Only in this way will it be possible to profit from all the advantages promised by an ITS architecture.

2.5 Some examples of ITS

Applications of Intelligent Transportation Systems have already been turned into practice. Some of them can be considered as pioneering experiences, as their appearance dates to prior the term ITS had been formally coined. Undoubtedly they served to motivate much advance in this field. Examples that are already part of people's daily lives are reported in (CHATTERJEE; MCDONALD, 1999).

- area traffic control;
- electronic tolling;
- driver information;
- trip planning systems;
- automatic vehicle control.

Bochner (1998) also lists some practical examples of ITS in use. Most of them have been focused on higher order roads as freeway systems in the United States and the motorway systems in the United Kingdom, for instance.

- detection and surveillance systems to help transportation agencies to identify incidents;
- ramp metering systems consisting of mechanisms for prioritising and expediting through movements on freeways;
- electronic toll collection to improve payment by electronic means;
- variable message signs to warn drivers about incidents within their routes and to post other useful information;
- responsive traffic control systems to meet variable demand and to prioritise special services, such as ambulances and public transport;
- multi-jurisdictional transportation centres aimed at monitoring and managing transportation operations in a highly efficient manner;
- information systems on the Internet to allow information on prevalent conditions of urban networks to be accessed from service providers' web sites (LYONS; MCDONALD, 1998).

Although much has already been put in practice, researchers still work to identify ITS potentialities. Chatterjee and McDonald (1999) point out some areas where ITS-based solutions present a great deal of applicability.

- traffic management and control systems are mostly elected as an application field where novel technologies are tested. Incident detection, area traffic control, and electronic tolling, for instance, are aimed at improving operational efficiency and are very likely to exert influence on drivers' behaviour as they perceive improvements in travel costs;

- in-vehicle devices, such as automatic speed control and anti-collision systems, seek to improve safety and comfort, as well as operational efficiency. They are also found to have the potential of altering the perceived attractiveness of a particular mode;

- traveller information systems are intended to improve knowledge of travel alternatives and network conditions, which is expected to play a central role in decision-making.

Thinking about all the benefits of ITS architectures feeds the imagination and makes one envisage the future traffic and transportation systems. And evidence shows it is definitely not far in the future. This way, Garrett (1998) suggests some interesting forthcoming scenarios for ITS.

- fully automated vehicle control;

- higher order roads as freeways, expressways, motorways, and principal arteries, operating under fully automatic control, which means hands-off driving;

- some roads operating under driving-assisted mode, allowing for hands-on driving aided by sophisticated warning systems;

- full integration of all transport modes in order to provide the optimum mix of public and private services;

- transport user charges will apply across all modes and services;

- vehicles will be 'smarter' and safer. They will be equipped with information and navigation systems, collision avoidance mechanisms, security and Mayday alarms, drowsiness detection, and black boxes to record and report errant behaviour.

2.6 ITS related issues

Practical applications of ITS have proven to be of paramount importance and indispensable to society. However, there remain boundaries to be overcome. According to Garrett (1998), researchers still have to tackle challenges posed by technical, cultural, psychological, and economic issues, even though potential benefits of intelligent technologies are undoubtable. This has motivated much work, which turned ITS into a multidisciplinary field to which not only researchers but the whole society have been devoting special interest.

- *assessment* is related to how benefits and impacts are to be evaluated. It plays a central role in the process of designing and implementing ITS. Currently, researchers strive to devise adequate computational tools, both hardware and software, for assessing the very complex nature of ITS applications;

- *standards* have also become a technical matter of special concern. Finding adequate means to ensure compatibility, inter-operability, and easy upgrading of systems, as well as avoiding conflicting communication protocols and transmission media is imperative. Enabling the interaction of legacy models, systems, and services of different providers is also very important to grant interaction and to the success of ITS applications. Finding standards has been subject of works carried out by research groups, technical organisations, and competent authorities around the world.

Besides technical matters, Bochner (1998) also emphasises the need for drawing government and public attention. The author suggests some strategic approaches to attract such support.

- combining resources of different jurisdiction regions, which is necessary to efficiently manage transportation in an inter-jurisdiction fashion. This has been initially identified as a problem of countries with autonomous jurisdiction units. However, globalisation has brought that to inter-country level, as in the European Community;

- contemplating local street systems, as most attention has been given to traffic control systems, which is not so evident for travelling public, mainly public transport users. These technologies become more evident when providing users with individual benefits;

- prioritising critical areas within the city, which would avoid implementing complete systems in unusable segments. This might make benefits more evident in a very short-term;

- publishing successful experiences, especially those that benefit average travellers. Publicity is always a good way for calling attention of public.

Yet, Garrett (1998) presents an issue of a more social concern, which is privacy. It is naturally a polemical subject. However, this topic is also important and the author suggests it should be considered in a sensitive and non-emotive way. Surveillance and monitoring systems should work in a way such that individual's privacy and safeguard are ensured. And, in the era of telecommunication and the Internet, which have been the great boosters for the integration of information systems, this brings about the *security* matters. They have revealed to be a challenging technical subject for researchers, as data get more and more exposed in such a world-wide web.

2.7 Advanced Traveller Information Systems

Grouping travellers in accordance with their common preferences and characteristics, identifying effective and potential behaviours, and understanding humanlike decision-making are the elements toward the development of Advanced Traveller Information Systems (ATIS). These applications rely on their ability to influence behaviour to tackle the ever increasing congestions in urban traffic systems. Together with ATMS, these technologies have proven to be very effective and to produce very short-term improvements (ADLER; BLUE, 1998).

The main premise behind ATIS assumes that by providing users with timely and appropriately designed information it is possible to modify decision-making and to affect behaviour patterns. These likely changes are expected to enhance the efficiency of transportation facilities. Adler and Blue (1998) highlight some of the primary goals of providing travellers with information.

- better managing traffic flow;
- enhancing driving functionalities;
- improving traveller safety.

Accordingly to Barfield et al. (1989), the design of ATIS would have two main purposes, namely to inform and to aid travellers.

- *informing* is achieved by collecting, designing, and delivering real-time traffic information;
- on the other hand, *aiding* is performed on the basis of storing, displaying, and delivering dynamic route guidance (DRG) and vehicle navigation information.

Whether it is designed to inform or to aid, ATIS is always expected to yield short- and long-term outcomes, both individually and collectively. For example, improving motorist response to incidents and peak hour congestion is likely to be achieved on a short-term basis. However, modifying commuter behaviour patterns towards a more efficient use of existing transportation resources is expected to be a long-term outcome.

Although this specific technology has recently gained special interest from scientific community, applying information systems to the field of traffic and transportation dates back to the 1950s. Since then, much work has been identified on the development of surveillance and real-time information systems to overcome traffic difficulties. About two decades ago, in-vehicle route guidance facilities were first envisaged, and centrally controlled variable message signs are now becoming more common within many metropolitan areas (ADLER; BLUE, 1998). Over the history, studying and understanding human factors commence to be preponderant, as people interaction gets to be much more facilitated. And it is also true for ITS, where such an interaction is likely to be very affected by the information provided through exogenous sources. So, assessing its applicability, public acceptability, and its effects, becomes imperative.

It is important to bear in mind that the term ATIS is used to identify systems that are aimed at conveying information both to drivers and to general travellers. Some authors prefer to distinguish between Driver Information Systems (DIS) and Traveller Information Systems. In this text the term Traveller Information Systems (TIS) will be used indifferently.

2.7.1 Advantages of using TIS

Information systems are the relying technologies that endow ITS with the ability to change traffic patterns by directly exerting influence on the moving elements, which are the travellers. Such a perspective seems to generalise the traditional approach of seeing as moving element just drivers and vehicles, yet indistinguishably. In fact, it now considers the occupancy factor encompassing collective transport modes, as well. The deployment of information systems promises a wide range of attractive benefits, which are identified in a number of works such as in (BARFIELD et al., 1989) and (ADLER; BLUE, 1998).

- in economic terms, if compared with traditional approaches based on physical modifications to road infrastructures, information systems are considerably much cheaper;

- collective benefits arise from the fact that potential negative impacts on major social issues such as land use, environment impact, and air pollution are to be reduced. However, yet related to equity as highlighted in (GARRETT, 1998), accessibility to new technologies remains to be a difficulty to overcome;

- by providing real-time information on individual basis, motorists can enhance their knowledge about the network, which will likely impact future decision-makings. TIS has a great deal in assisting personal needs, not only during the journey but specially within the whole process of travel-planning. Despite of being an individual advantage, the effects of such an information contribute for the more efficient distribution of travellers' routes, which may be perceived on a collective basis, as well;

- the political benefits are quite related to the economic ones, as public authorities are more likely to opt for inexpensive solutions;

- public health is also expected to profit from TIS. These novel technologies can help reducing anxiety and stress associated with travel-planning, way-finding, and navigating throughout the network;

- improvement of the overall system performance is another important contribution of TIS. Reductions in travel time, delays, fuel consumption, and emissions are expected if a significant number of users adhere these technologies.

2.7.2 Categories of TIS

Adler and Blue (1998) grouped traveller information systems (TIS) into three categories, according to the kind of technologies applied.

- *first-generation TIS*, dating to the 1960s and 1970s, represent a first attempt at using communication technologies for information dissemination;

- *second-generation TIS* refers to today's ATIS and encompasses a wide range of new technologies. They have been designed to different purposes, such as dynamic route guidance, informing current traffic condition on a real-time basis, and conveying other useful information as off-road traveller services;

- *third-generation TIS* will allow systems to more effectively respond to travellers' within-day travel-planning needs and to easily adapt their functioning to users' travel preferences, ability to acquire spatial knowledge, and attitudes over time.

The authors defend that, in spite of presenting some degree of autonomy, second-generation TIS cannot be considered really intelligent. As for third-generation systems, their ability to behave both autonomously and proactively toward assisting users' needs seems to constitute a domain to which AI-based solutions can certainly be applied.

2.7.3 Types of information

Surveillance facilities of all sorts gather an enormous amount of data, which need to be filtered and tailored so that travellers can use it. However, providing travellers with suitable knowledge certainly poses the question: *what is the important information, when should it be delivered to users, and how should it be presented?* Chatterjee and McDonald (1999) group information into two basic types, according to the time users are to receive it. They also mention some ways through which information can be presented.

- *pre-trip information* is acquired before starting the trip. It is said to be *historical* if realised from travellers' knowledge, which evolves over time. When the information is acquired just moments prior to the journey, it is said to be *contemporary*. It is also possible to access *predictive* information, which is derived from the two others. Pre-trip information has a direct effect on decision-making and drives departure time, route, transport mode, and service choices;

- *in-trip information*, to the contrary, is acquired during the course of a journey. The *static* information outside the vehicle is available to every motorist, such as traffic signs. With advances in telecommunication, *dynamic* information is becoming common through VMS or DRG, for instance. However, in-trip information is not accessible to all travellers yet.

Much work is also reckoned on adequate channels through which information is to be conveyed to travellers. They will mostly depend on the addressed public and on the way knowledge must be presented. Then, it is possible to identify a number of means to provide travellers with information (CHATTERJEE; MCDONALD, 1999).

- *mass media*, such as newspaper, radio and television, and nowadays the Internet can provide advance warning or information about the current state of transportation resources. It is a collective information rather than oriented for individual purposes;

- through *motorist in-trip* channels drivers can receive any kind of advise or warning during the course of a journey. They can be basically presented by means of in-vehicle and outside-the-vehicle facilities, for example DRG, navigation systems, and VMS;

- *public transport in-trip* channels are placed on strategic spots at stations, stops, and interchange points to provide in-trip travellers with real-time information. They are intended to reduce the frustration and uncertainty experienced by waiting trip-makers;

- *trip planning systems*, including public terminals, telephone enquiry services, and home- and office-based systems, as cable television and the Internet are particularly suited to provide public transport information. Trip planning systems, specially when available at origin, can be used both to plan the trip in advance and to check conditions before setting off.

Information systems have always been a challenging field for Computer Science, and so has TIS that, on its on right, offers a number of interesting paths for multidisciplinary investigation. Considering the very presence of human beings in this scenario, it is important to bear in mind that there are also pitfalls armed by misinterpretation of information contents. Yet, excess of elements and the frequency advises and warnings are posted to users may confuse one's mind. Another function of TIS is to filter targeted users from the population altogether. What if everyone follows an alternative route to avoid a congested itinerary? The alternative would also become congested very certainly. So, reliability is an imperative quality for information, and it seems to be very affected by the interrogatives *what*, *how*, and *when*, applied to its contents. Inaccurate information could yield very hazardous situations.

2.7.4 Types of sources

As to sources of contents, Adler and Blue (1998) identify three basic types where information is generated and from which it is delivered.

- *historical experiences* of previous trips, gathered over time and inferred through learning;
- *current perceptions* of the network conditions;
- *exogenous sources*, which are all the facilities provided by TIS.

According to Chatterjee and McDonald (1999), it is useful to consider the essential characteristics of the information rather than its source. Nevertheless, most issues related to the contents also apply to the sources.

- in that related to availability, whether it is pre-trip or in-trip information;
- according to the sort of transport, whether it is mainly aimed at public or private transport users;
- as to the currency of information content, whether it is static or dynamic;
- according to the targeted public, whether it is customised to individuals or made generic;
- concerning interactivity, whether it is passive or the user can interact with it;
- according to the typical nature information content, whether it is homogeneous or heterogeneous.

In fact, what source to apply is much dependent on the features of the information it gets to handle. So, this is also the subject for many research works, which again involve a number of different disciplines. Owing the existence of different systems, different providers, and different manufacturers, it is necessary to standardise the way information is dealt with to make it better understood, to ease access, and to grant it reliability. Trustworthiness of sources can affect acceptance of users. In addition to the concerns above, communication, social abilities, human-device interfaces are equally stimulating topics.

2.7.5 Types of behaviour

Experience shows that traveller behaviour is likely to be affected by both prevalent conditions of the environment and characteristics of the information. Finding a pattern in a very complex, stochastic, and uncertain domain is really difficult, and so is influencing and modifying it on a controlled basis. This is all researchers have been looking for. Chatterjee and McDonald (1999) assert that it is quite acceptable that some situations will induce travellers to make specific decisions and behave accordingly.

- congestion and incidents;
- information systems;
- prevalent situations;
- control systems;
- other travellers' behaviour;
- planning the trip.

In recent years there has been a significant increase in the research as to understanding routing, way-finding, and navigation facilities. The aim has been basically to figure out the role that TIS could play in providing travellers with the necessary information (ADLER; BLUE, 1998). It is quite reckoned that decision-making is affected, in any manner, by source, content, availability and currency, and reliability of information.

Besides, the spatial knowledge of the network seems also to contribute to the way people commutes (ADLER; BLUE, 1998). It poses other interesting thoughts.

- a representation of the spatial orientation can be built in one's mind through either observing maps or learning from repeated trips. *To what extent would a driver accept to be 'blindly' guided by an electronic device throughout a network it does not know?*
- the networks are of a very stochastic nature as traffic may behave widely differently for many reasons. It seems that making repeated trips contributes to the learning of such variations. *How can network dynamics be represented and contribute to one's individual model? How can it affect decision-making?*
- people present a natural tendency to mature and have their behaviour changed over time owing to past experiences. *How can this behavioural evolution be influenced as the environment gets vastly populated with autonomous facilities?* travellers will have to be able to detect, understand, get used to, and learn with them. On the other hand, network dynamics is strongly associated to the way people commute. *How can such an iteration be handled to yield a sustained optimal state?*

Understanding spatial knowledge and network dynamics is certainly a step toward realising travellers' behaviour patterns. This is of invaluable help to figuring out a way to alter it to some extent and, most important, on a controlled basis.

Population must also be segmented as to users and non-users of information resources. Even among users it is necessary to consider those unwilling to use the content provided, either because they trust more on their own knowledge about the network or because they consider the information is not reliable enough (CHATTERJEE; MCDONALD, 1999). Barfield et al. (1989) grouped commuters into four classes according to their willingness to adjust behaviour in response to the information supplied.

- *route changers* are always willing to change routes either before or during their commute;
- *non-changers* are reluctant to change time, route, or transportation mode at any time within the journey;
- *route and time changers* are likely to change either route or departure time. They can even change both;
- *pre-trip changers* are not likely to make en-route changes, but willing to change time, route, or even mode prior to leaving their origins.

In this way, TIS-based solutions are commonly designed to affect one or more of four aspects of travellers' habitual practice, as reported in (MAST, 1998).

- *departure time*, by influencing the time commuters live their origins, such as home, work place, and others;
- *means of transport* is basically associated to users of public transport as bus and train services. The private sector of car pooling is also experiencing some advantages of TIS technologies;
- in a commuter scenario the *pre-trip route* choice is less likely to be changed. Nonetheless, information specifically tailored to this end can make much difference. Seasonal journeys, as on holidays and for entertaining purposes in general, seem to profit a lot from pre-trip advices;
- *en-route diversion* seldom happens for commuters unless either an incident blocks their usual itinerary or it gets to be adjusted for the sake of an unexpected purpose elsewhere than the common destination. In-vehicle information facilities may play an important role in such situations.

There are at least two reasons that make people alter their departure time, as suggested in (BARFIELD et al., 1989). If arrival time is flexible, travellers can delay their departure until after congestion has died out. Otherwise, they can judge if there are unusual delays due to any incident or on-site maintenance works and leave earlier so as to meet their arrival deadline.

The way people make travel choices, including destination, mode, departure time, and route, depends on their needs, personal preferences, and the information available. The ideal scenario would imply that travellers had a perfect model of the system so that they could make the optimum decision. It is definitely fictitious as the environment is not completely accessible, which is prohibitive to individuals having such a perfect knowledge (ADLER; BLUE, 1998). As to the strong association between cognition and behaviour, three issues of primary concern are highlighted in (JACKSON, 1994; ADLER; BLUE, 1998).

- *identifying likely effects of cognition on behaviour*. It is necessary to better understand the processes by which a driver seeks to acquire and use special knowledge under normal conditions, which means without the aid of any exogenous information. Thus, it is critical that TIS mimic drivers' cognition and reflect their behaviour;
- *understanding cognitive representation of the environment*. One implicit goal of TIS is to provide drivers with adequate means by which they can build a perfect model of the world. This is expected to aid drivers in trip-planning;
- *assessing acceptability for exogenous information*. It is important to identify the factors that may have the largest influence on deciding whether to accept and rely on exogenous information. Thus, TIS should work in such a way its services are perceived to be personalised, timely, and relevant.

However, in order to allow for information-driven behaviours it is imperative that users effectively use the information provided. This way, the design of an efficient TIS will also depend much on the understanding of which and how human factors contribute to the effective use of such technologies, as suggested in (BARFIELD; MANNERING, 1993; ADLER; BLUE, 1998). This again poses another series of relevant concerns.

- whether travellers use such technologies;
- the reasons that would make travellers use the information provided;
- how and when travellers are more likely to use the information provided;
- the way travellers perceive sources of information;
- how travellers perceive the likely consequences of using such systems.

Additionally, it is important to have a good comprehension on how the content is understood by drivers. It is quite acceptable that different drivers may have different interpretation of the same information. Adler and Blue (1998) also point out cost, compliance, oversaturation, long-term effects, and driver comfort as complementary matters to be addressed. Lots of research efforts have then been carried out in order to study traveller behaviour according to a number of different factors; some of which are discussed in (CHATTERJEE; MCDONALD, 1999). However, albeit their findings have shown that there is a great variability in responses, travellers tend to rely on their own knowledge about the network conditions and dynamics.

In commuting scenarios, it has been shown that patterns are largely habitual in nature. However, in spite of being a collective standard, it seems that each person tends to have more than one 'typical' daily set of preferences (CHERRETT, 1998; CHATTERJEE; MCDONALD, 1999). As to departure time choices, commuters have been found to possess route 'strategies' based on a series of home departure times, which allow them to arrive at their destinations within an acceptable delay at each destination. Such a perception of lateness can be associated to some interpretation of cost. Also, changes to the itinerary chosen are mostly dependent on the conditions encountered. This makes diversions to be more likely to happen at specific 'decision' points within the journey.

2.7.6 Requirements for TIS

TIS has proven to encompass much complexity on its own rights, and to turn it into reality it is imperative to deal with a wide range of parameters and their relations. Barfield et al. (1989) summarise some premises, which are mostly regarded to commuting specificities. They must be taken into account for the development of effective driver information systems.

- as to the heterogeneous nature of audience it is important to bear in mind that commuters cannot be treated as a single and homogeneous group of traffic information users. This is concerned with *whom to target for a particular type of information*;

- specificities of information should be oriented in order to have an impact on drivers' behaviour. The system must be capable of delivering the content tailored to the particular driving decision faced by individuals during the journey. The issue is then *how to tailor the information* to impact the targeted group;

- sources of information must provide for a regular delivery of accurately, timely, and appropriately designed traffic information in order to produce a long-term positive modification on commuters' behaviour. That means *how to deliver such information* at appropriate decision points within the journey.

The authors assert that a single successful information system is capable of meeting the needs of a wide range of different drivers under varying conditions and stages of travel. With this aim, a single integrated driver information system should consist of carefully designed information modules oriented to address particular commuting decisions of well defined subgroups of receptive commuters (BARFIELD et al., 1989). Few examples are already in use all over the world, mainly oriented to collective use. Facilities for individual needs have not yet reached most of the population and their effects are expected to be noticed on a long-term basis. Nonetheless, scientific community, as well as the industry are striving to enhance TIS technologies as some requirements are still to be met toward making their use popular (ADLER; BLUE, 1998).

- such systems must be affordable. Such technologies, mainly those aimed at individual use, are quite expensive at the moment;
- they must provide understandable and reliable information. Confusing and excessive contents are proven to yield misunderstanding by drivers;
- the systems must contemplate within-day and day-to-day travellers' preferences. Behaviour cannot be generalised as some preferences are found to be seasonal and apply weekly, monthly, and even annually, as for vacations for instance;
- TIS must be exceptionally user-friendly. It is proven that easy-to-use interfaces are partially responsible for the popular acceptance of new technologies;
- TIS must be capable of providing customised travel assistance, and people must perceive to be provided with personalised valuable service.

The ability of TIS to broadcast valuable information has been widely recognised as significant advances are incorporated to vehicular technologies and made available. Then, users can already profit from facilities and services such as interactive user interfaces, vehicle location and intelligent mapping, path search, yellow page directory, multi-modal information, and dynamic route guidance (ADLER; BLUE, 1998).

2.7.7 Framework to assess ITS technologies

In order to submit these issues to expert appraisal, conceptual models of driver behaviour under information have been proposed. It is definitely imperative to devise adequate tools to simulate and assess the performance measures of such solutions. Up to a while ago, they were based on traditional techniques, what is easily explained by the constraints imposed by the computational resources available then. With the increasing capacity of today's hardware architectures and the availability of new computing methodologies, these models can now be implemented through the use of a wide range of different, robust, reliable, and intelligent ways. They have been proposed for many purposes, but some areas have awarded special interest (ADLER; BLUE, 1998).

- *determining traveller preferences for information systems*, which is mainly dictated by the type of information, the way information is received and identified by users, and the presentation media;

- *modelling route choice and switching*, which serves for researches as a means of understanding the influence that information can have on decision-makings;
- *representing and modelling cognition*, as the spatial representation in one's mind may affect routing behaviour and the need for information;
- *assessing and evaluating effects* is probably the most important aims for evaluation tools, as information systems are expected to exert great impact on the network performance. Using traffic simulation environments is a frequent practice;
- *conceptualising dynamic models* as a means to handle and to understand the very stochastic nature of traffic scenarios and the influence that ITS technologies certainly have on system's dynamics.

According to Chatterjee and McDonald (1999), the way sources of information are modelled is more or less of no importance. The authors also suggest that it is clearly impractical to expect to be able to model all effects and scale of drivers' responses. The modelling efforts should concentrate on flagging up those responses that are important and likely to be measurable under some conditions. Route and departure time are likely to be very affected by the quality of information, for instance. Whether the contents are made available pre-trip or in-trip and whether they are accessible from inside or outside the vehicle, what decisions are intended to be affected, and channels through which the information can be conveyed to drivers are some factors that should be parameterised.

After gathering the necessary requirements to better describe the relations between behaviour and the information supplied, Chatterjee and McDonald (1999) proposed a general framework to aid the design and assessment of ITS architectures. According to the authors, existing modelling procedures do not explicitly take into account trip makers' knowledge and have limited capabilities for assessing impacts of information systems. Besides that, there remains a lack of evidence of the impact that these systems have on behaviour and travel patterns against traditional approaches, for instance the enhancement of physical infrastructures. Such a modelling framework is supposed to meet two major needs, namely incorporating support to assess responses to ITS technologies and providing transport scientists and practitioners with a computer-aided tool. Its components and relations are depicted in Figure 2.2 and briefly commented as follows.

- *household and person generator* synthesises the population within the study area, such as demographic data, vehicle ownership, work place, and so forth. It could use information from demographic simulation models or from transport and land-use relations, for instance;
- *week activity plan* generates a plan for the forthcoming week, which is then assigned to each individual of the population. It relies on the assumption that people tend to arrange their lives on a weekly basis. Activity purpose, destination, starting time, and duration should be represented;

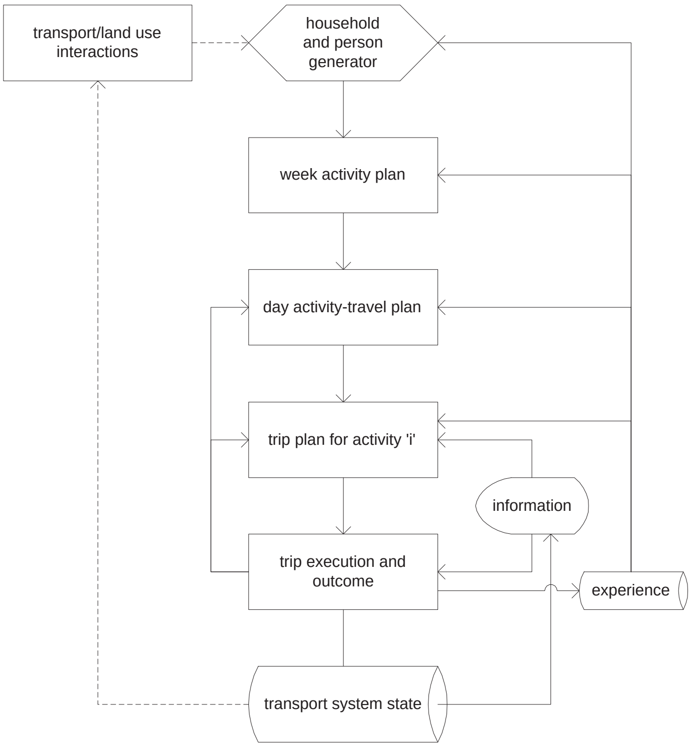
transport/land use interactions
household and person generator
week activity plan
day activity-travel plan
trip plan for activity 'i'
information
trip execution and outcome
experience
transport system state

Figure 2.2: Framework for assessing ITS technologies, adapted from (CHATTERJEE; MCDONALD, 1999).

- *day activity-travel plan* generates plans on daily basis, including the itinerary to the sites where each activity is to be performed. Purpose, destination, starting time, departure time and duration, mode, route and parking facilities are some characteristics to be explicitly represented. Daily planning can be affected by outcomes of trips performed on the same day, which may force the reconsideration of the original options;

- *trip plan for activity 'i'* selects the activity from the day plan to be performed next. The trip plan for the subject activity can be affected by pre-trip information, experiences of performing the trip on previous days, and even journeys made earlier on the same day;

- *trip execution and outcome* supports the execution of the trip and outputs its performance measures. This can be given in terms of travel time, route, parking location, and delays, for instance. The *experience* model is then updated accordingly to the travel conditions and degree of satisfaction perceived. Also, the course of a journey can be affected by in-trip information, as well as the prevalent conditions of the network during the period the trip is being executed;

- *transport system state* is the data representing the actual state of the transport network as trips are performed. It feeds the previous module as drivers perceive the environment conditions during their journeys, and also bases the *information* systems so that up-to-date contents can be provided to users both pre-trip and in-trip;

- *information* simulates the operation of the information systems that are present in the study network. Surveillance facilities ensures that TIS are able to provide reliable contents. Pre-trip information is delivered through the module *trip plan for activity 'i'*, whereas in-trip is enabled through the *trip execution and outcome*;

- *experience* is the data representing the modifications, after each trip, in the state of people's satisfaction, knowledge, and perceptions of travel alternatives. It has great influence in travellers' preferences, learning, and decision-making in all levels.

2.8 Summary

Major urban areas, as well as their suburbs and accesses have notoriously experienced an increase of the recurrent traffic flow. This has frequently yielded traffic congestions that in turn contribute for waste of energy, for air pollution, and for excessive delays. In general terms it has brought about economic and environmental issues that need to be addressed through effective policies. Moreover, increasing capacity by means of physical modifications to the road infrastructure is even less feasible as space lacks.

Resulting from the efforts to tackle traffic and transportation problems, today's systems have been considerably transformed by novel mechanisms and strategies of traffic management. With respect to the ground traffic and transportation, the Intelligent Transportation Systems basically rely on the integration of autonomous processes aimed at optimising the usage of limited capacity road networks. Communication and computing techniques serve as the framework for such technologies. Rather than intervening in the static entities, namely roads and control systems, one premise of ITS-based technologies is to act directly upon the moving particle, which can be seen as a vehicle-driver unity.

Besides considerable advances in on-board devices to aid driving tasks toward safety, influencing drivers' behaviour patterns is another key objective of ITS. As drivers have only a local access to the network conditions during a journey, exogenous information sources seem to be of valuable help. They provide drivers with knowledge and advices that can be used in building a model of the system as a whole. Such an internal model is expected to improve reasoning and decision-making locally, but also it is expected to enhance the overall system performance. Variable message signs, route guidance systems, the Internet, radio broadcast, and now mobile technologies are already part of citizens' daily lives.

Traffic network models and traffic theory will be, and certainly already are affected by the Intelligent Transportation Systems. Representing new performance measures in modelling and simulating today's traffic scenarios has revealed to be a tricky task, though. The abstraction process has been brought to very detailed levels, which makes the microscopic approach suitable to this end. However, traditional models are mostly result-driven, are not so consonant with their implementation, and their rigidity makes scalability hard. On the other hand, owing the very complex nature of certain applications it is not possible to dissociate the domain model from the data structures and algorithms that base its implementation. Process-driven approaches usually rely on such an assumption, which overcomes difficulties of traditional models by easing scalability and enhancing robustness, for instance.

Modelling and simulating ITS-based technologies, as well as assessing their impacts to the overall performance of traffic systems demand for robust methodologies to cope with increasing levels of complexity. And this is specially the case of models that involve humanlike reasoning and decision-making.

3 MULTI–AGENT SYSTEMS

3.1 Overview

In general terms, Artificial Intelligence (AI) is especially concerned with the development of computational models that mimic human intelligent and rational behaviour, and encompasses a wide range of multidisciplinary knowledge areas. Multi–agent systems (MAS) is a sub-field of the Distributed Artificial Intelligence (DAI) whose abstraction approach basically consists of representing the application domain by means of multiple entities, coined agents. Contrary to the approach adopt in the traditional AI, scientists have been motivated by functional and spatial distribution of tasks and components of some complex systems. In this scenario, MAS constitutes a central research and application area, which can be seen, at a first glance, as computational systems composed of several software entities capable of mutual and environmental interaction (WEISS, 1995). Although the main concern of MAS has relied on the concept of intelligent and autonomous behaviour, defining what an agent is has been the focus of much controversy. Nonetheless, MAS has gained especial attention from both Computer Science and other knowledge fields.

This Chapter is aimed at giving a broad view on multi–agent systems and autonomous agents, as well as their potential application to the domain of traffic and transportation engineering.

3.2 Desired features in intelligent agents

Russell and Norvig (1995) define agents as any entity capable of perceiving facts through sensors and acting upon the environment through effectors. Rationality allows an intelligent agent to act toward making the right decision, which should lead to successfully achieving a goal. So, a key feature in multi-agent systems is autonomy. Autonomous agents exist in the environment independently of the problem to be solved for the whole society (HÜBNER, 1995; FROZZA, 1997). According to Weiss (1996), what really makes an object like a software program or an industrial robot to be an agent are some properties like perceptual and cognitive skills, communicative and social abilities, affection and emotions, and autonomy, in other words, to have self-control (WEISS, 1995). An intelligent agent can possess a wide range of characteristics, which will clearly depend on the application the agent is designed for. Frozza (1997) points out some desirable features that an agent should present in some extent.

- an agent may represent either a real or a virtual entity;
- it must be inserted in the context of an environment;
- it must be capable to perceive the environment and other surrounding agents;
- an agent is capable to perform some actions in the environment, which can change the state of the environment, the agent's internal state, or the state of other agents;
- an agent must present communication capabilities;
- it must present social ability;
- an agent possesses goals to be achieved, and should be autonomous to carry out its tasks toward accomplishing them;
- an agent is capable of reacting to changes in the environment;
- it must present some degree of initiative in order to seek to accomplish its goals;
- it must present adaptability, in other words, it must be capable to adapt its behaviour as the environment evolves;
- some degree of mobility is also desirable, as an agent would need to change its physical location in the environment;
- an agent must have knowledge about itself, about the environment, and about other agents. Also, an agent may have an initial knowledge, which can be extended as it interacts with the environment and with other agents;
- the reasoning feature is also intended to give an agent the capability of making inferences about the behaviour, tasks, and plans of other agents.

Whatever the features one may design for an agent, rationality and autonomy may be considered two major properties that deserve special attention. According to Russell and Norvig (1995), being rational at any given time depends basically on four aspects, namely the performance measures that define degrees of success, the sequence of perceptions at certain instant, which consists of everything the agent has perceived, the knowledge about the environment, and the actions the agent is able to perform. On the other hand, autonomy would be concerned with situations in which individuals did not need to rely on any perceptions, as actions would be based solely on built-in knowledge. Russell and Norvig (1995) also define autonomous systems as systems whose behaviour is determined by previous experiences, as well. Such a definition suggests that autonomy should be achieved through some sort of learning mechanism.

3.3 Structure of an intelligent agent

With regard to their autonomous nature, intelligent agents should possess a structure that allows them to perform their tasks in order to achieve particular goals. In this sense, an agent perceives the environment and other agents through sensors. Effectors, on the other hand, are the structures an agent uses in order to act upon the environment and to interact with others, as suggested in (RUSSELL; NORVIG, 1995). The authors identify three major elements that bounds the agent design, namely the agent behaviour, the agent program, and the agent architecture.

- the agent behaviour can be understood as the action that is performed after any given sequence of perceptions. Such a behaviour can be based on either its own experience or some sort of built-in knowledge;
- the agent program is the function that implements the mapping from perceptions to actions such that the agent can play its role onto the environment;
- the agent architecture can be seen as the structure that runs the agent program. It should present a means to receive information from the environment, as well as to properly effectuate agent's actions.

Pragmatically, the structure of an agent comprises both the program and the architecture. It is provided with internal data structures that are updated as the agent perceives new information from the environment. Such data structures can be understood as the knowledge of an intelligent agent. They are operated on by decision-making procedures to generate an action choice, which is executed through the agent's architecture. Agent programs are functions, which implement the mapping from a perception, or a sequence of perceptions, to actions.

As suggested in (WERNER, 1991; FROZZA, 1997), the structure of an agent consists of two major parts, one that is static and the other that is dynamic.

- the static part is the agent architecture, which defines the representation of the knowledge the agent is capable to keep and the way such a knowledge is represented and handled. The way the knowledge is represented can be influenced by several factors, for instance: how the environment is represented; capability of representing what an agent can describe to other agents; problems and goals an agent needs to solve or achieve; plans to be followed by the agent; and likely choices and decisions to make;
- the dynamic part corresponds to the processing methods that effectively allow an agent to behave in the environment. They can be grouped into reasoning capabilities, used by the agent to make inferences about its knowledge, and decision-making mechanisms, which allow the agent to make decisions in order to accomplish its goals.

Although agent structures are presented in (RUSSELL; NORVIG, 1995; WERNER, 1991; FROZZA, 1997) in different ways, there is a consensus that the knowledge representation and the functions, which can in any form transform its content, are central to AI and to agent designing. These two features of an agent structure are also important to define in which extent an agent is reactive or cognitive.

3.3.1 Reactive agents

Systems composed by reactive agents are generally simpler than those composed by cognitive ones. This can be easily explained as reactive agents do not present mental states. Also, planning and reasoning capabilities are not strong characteristics and the major idea behind such an approach is based on an emergent behaviour. The underlying idea of emergent behaviour is to achieve complex, intelligent, efficient, and more organised behaviour through the combination of several simpler structures, as exemplified in (DROGOUL, 1993) through a model devised to represent a colony of ants. The overall behaviour of a system rises from the interaction among agents and the individual performance of their tasks. Some features presented in (FROZZA, 1997) are suggested to be present in most of the systems formed of reactive agents.

- reactive systems are mostly inspired in ecological organisations;
- knowledge about the environment and about others is implicitly represented through the reactive behaviour of the agents;
- such systems follow a behaviour-based approach, that means behaviour is expressed on the basis of the state of the environment. Any variation in this state triggers changes in the agents' behaviour;
- agents behave on a stimuli–action basis. Actions are carried out as response to predefined stimuli from the environment.
- agents present perception and communication capabilities, albeit these capabilities are limited;
- they do not present complex reasoning or inference capabilities, neither memory of past experiences and of results of previous actions;
- a reactive agent society is generally composed by several members.

According to Frozza (1997), some considerations should be made while modelling a domain by means of reactive agents. The phenomenon should be decomposed into a set of as simple and autonomous entities as possible, which will interact in order to reproduce the system behaviour. Each entity is seen as an agent with defined knowledge, capabilities, and behaviour that will interact with others and with the environment. So, and accounting for the behaviour-based approach mentioned above, the environment should be well designed. It is equally important to take into account that there may be passive agents, which do not possess neither action nor communication capabilities. With respect to the architecture of a reactive agent, albeit it is mostly intended to be conceived as a simple structure, it may be designed to cope with more complex tasks. The subsumption model, proposed by Brooks (1991a), is a traditional architecture to represent reactive agents that prioritises the execution of tasks distributed into different levels of complexity.

3.3.2 Cognitive agents

Cognitive agents are intended to acquire knowledge about the environment and about the others, and are able to interact with each other and with the environment, as well. Perception and communication play an important role as the behaviour of a cognitive agent is susceptible to be modified by exogenous information. This approach is based on the notion of mental states, such as intentions, beliefs, desires, compromises, choices, goals, and aptness, which are analogue or similar to those found in human beings (SHOHAM, 1990). Understanding the relation among them is of huge importance to better model cognitive and decision-making mechanisms, which is the focus of interest in the field of cognitive systems. As with reactive agents, Frozza (1997) suggests the following features can be found in cognitive agents.

- they are based on models of social organisations;
- they have an explicit representation of knowledge about the environment and about other agents;
- they are capable to plan their actions;
- they may present accurate perception and communication capabilities;
- they present mental states and can memorise past experiences, which are taken into account for future decisions;
- society composed by few members.

It is worth mentioning the features presented above and the ones presented for the reactive agents are complementary to each other in many ways. Thus, the combination of both approaches makes reactive and cognitive agents suitable to handle most specificities of a huge range of different application domains.

A general-purpose structure for a cognitive agent is presented in (DEMAZEAU, 1991; FROZZA, 1997), as depicted in Figure 3.1. The act of perceiving both the environment and other agents alone does not yield any modification, neither to the environment nor to the state of the agent itself. Nonetheless, such a capability is important to gather relevant information, which may enhance the agent's knowledge. It is this updating of the agent's internal model that may influence future behaviour. The knowledge of a cognitive agent can be viewed as a composition of three kinds of information.

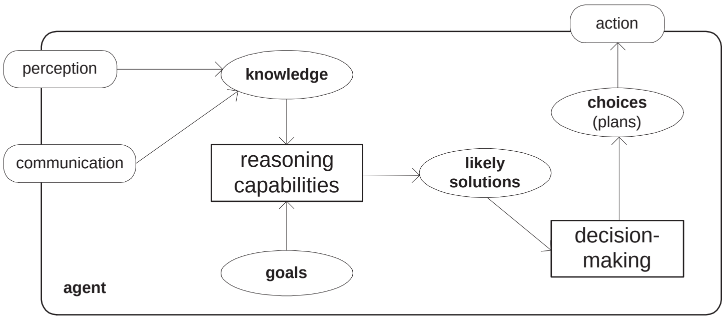

Figure 3.1: General structure for cognitive agents, adapted from (FROZZA, 1997).

- initial knowledge (also referred to as the built-in knowledge of an agent), which is defined in the modelling time;
- information gathered as the environment evolves;
- information gathered through the communication with other agents.

The initial knowledge and the information got from perceiving the environment can be seen as the accurate knowledge. On the other hand, information gathered through communication with other agents is considered to be uncertain knowledge. This sort of classification seems to be related to the confidence the agent has on the information sources. The former refers to the information the agent gets by itself. However, it cannot state anything about the latter.

Executing an action, however, does cause changes either to the state of the agent or to the environment, including other agents. Specifying a problem implies defining a goal or a set of goals, a set of actions, and a description of the initial state of the system. A task-planning procedure finds a suitable sequence of actions that better results in the changes that brings about intended states of affairs. A chosen plan can be delayed or even reformulated for the sake of unexpected conditions that does not favour its execution, for instance. This could lead the agent to find a contingency plan. In such situations, some mechanisms to ensure that plans can be constantly revised during its execution are also desired, so that unexpected results can be prevented or at least minimised.

3.4 Basic architectures for intelligent agents

Choosing between reactive and cognitive architectures for an agent will depend enormously on the application the agent is being designed for. Since the earliest ages of MAS research several models have been proposed as an attempt at fitting the requirements of each application domain. An agent model can reach a wide range of complexity both in structure and in functionality (WEISS, 1995). Russell and Norvig (1995) suggest four basic types of agents, which can range from a simple reactive to a more complex cognitive structure. Such basic models, namely simple reflex agent, agent that keeps track of the world, goal-based agent, and utility-based agent can be seen as start points for modelling application-dependent structures. A traffic scenario where a driver attempts to reach a destination from an origin is regarded here as a means to illustrate some conceptual ideas behind these models. What time to depart and which route to take are some examples of decisions a driver needs to bother with in that scenario.

The simple reflex agent (Figure 3.2) relies on the concept of condition-action rule. Instead of building up a table that contains every possible combination of perceptions and actions, this type of structure should contain only certain commonly occurring input-output associations. Some processing is done on the input from the agent's perception in order to establish a condition, which triggers some established connection to specific action in the agent program. This connection is well known as condition-action rule and would be written as an expression of the form **if** ⟨generated condition⟩ **then** ⟨mapped action⟩.

Humans also make such connections between perceptions and actions, some of which are learned responses and others are innate reflexes. A simple reflex agent uses condition–action rules in order to make the connection between perception and action. Although such agents can be implemented very efficiently, they pursue limited applicability.

- *sensors* allow an agent to perceive its surrounding environment. However, such an information is not enough for a driver to make a picture of the system as a whole, as cost of travelling through roads are not known *a priori*, for instance. Nevertheless, some information could also be gathered through exogenous sources, such as Traveller Information Systems;

- *what the world is like now* represents the actual state perceived by the driver. However, drivers are not likely to know about the exact state of traffic at a certain instant prior to the journey start. Nevertheless, individuals could either forecast the system conditions from previous experiences or realise it from the information supplied by exogenous sources. In the former case drivers cannot forecast the system state as simple reflex agent does not keep track of previous experiences;

- the decision on *what action should be done now* is made as a function of available established condition–action rules. For instance, *if* the cost is beyond a certain prefixed value *then* leave origin 15 minutes later;

- *effectors* will perform the agent behaviour, which is to start the trip at the chosen time through the chosen route.

The simple reflex agent mentioned above will work well if the correct decision can be made on the basis of the current perception. In other words, the current perception should produce a condition such that the agent can find some action associated to it.

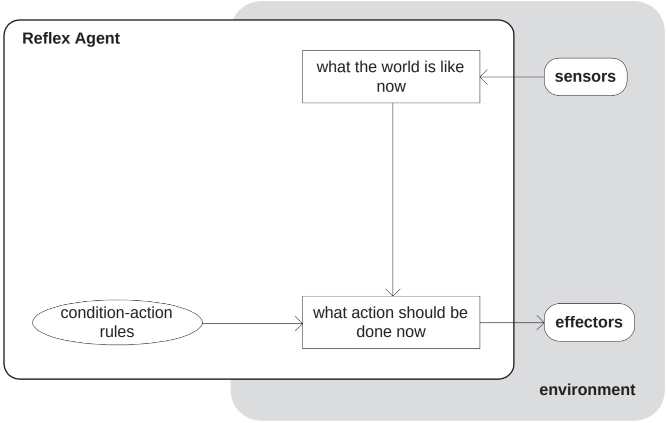

Figure 3.2: The simple reflex agent,
adapted from (RUSSELL; NORVIG, 1995).

The concept of internal state is concerned with how the system evolves. Depending on the complexity of the environment, sensors do not provide access to the complete state variables. Thus, the agent may need to maintain a model of the system internally so that it can distinguish between world states that generate the same perceptual input but that are significantly different. In other situations this internal state can be used as a means to draw conclusions about trends in the world dynamics. The objective of keeping track of the world is to improve the quality of the agent's knowledge basis. This is done combining the old internal state with the current perception to generate the updated description of the current state (Figure 3.3).

- information gathered through *sensors* is memorised so that the driver can remember previous experiences. Thus, the driver sensibility for experienced costs can now be considered a source of information about the environment state, even if regarded just as past observations. Information from exogenous sources can also be perceived;

- with a proper representation of *what the world is like now* the driver can forecast the actual system conditions and have qualitative notion about the environment state variables (road volumes, path travel time, waiting time at junctions, and so on). Individuals also have a clear idea about how the world evolves and about what effects their actions may provoke;

- as for the simple reflex agent the decision on *what action should be done now* is a function of available established condition–action rules, though.

Knowing the current state of the environment may not be enough to decide on how to behave, specially when the agent is facing several possible actions. The right decision may depend not only on the actual state of the environment. Some sort of goal describing desirable states of affairs to be brought about is also required. Some times, an agent must

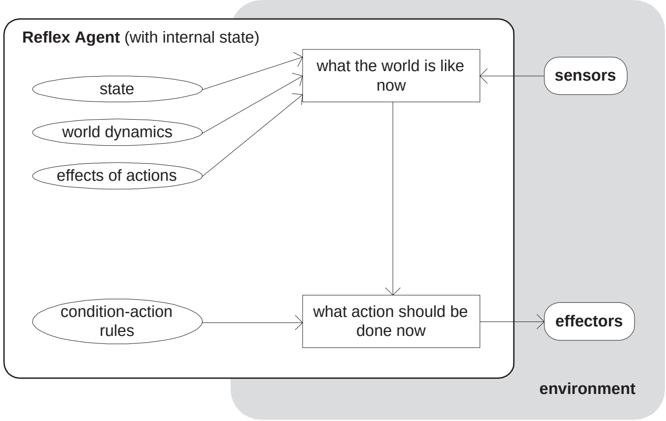
Reflex Agent (with internal state)
state
world dynamics
effects of actions
what the world is like now
sensors
condition-action rules
what action should be done now
effectors
environment

Figure 3.3: A reflex agent with internal state, adapted from (RUSSELL; NORVIG, 1995).

search for possible actions and plan action sequences in order to achieve its goal. This kind of structure is fundamentally different from the condition–action rules mentioned above. It involves some considerations about the future, both related to effectiveness and to satisfaction (Figure 3.4). In reflex agents, this information is not explicitly used, because the designer has pre-established the correct action for various cases.

- the interpretation for *what the world is like now* is the same as in the previous example. People will forecast the actual state from an internal model, from the knowledge about how the world evolves, and from experiences of previous actions;

- Evaluating *what it will be like if action A is chosen* results from weighting their actions as they have a specific goal to achieve. For example, drivers seek after a minimum journey cost. Such an evaluation is a function of knowledge about how the world evolves and experiences on the effects of previous journeys, for instance;

- *what action should be done now* is a function of the agent's goal, differently from the previous structures, whereby the action usually is a function of condition action–rules. The need for arriving at work earlier on certain day may influence, and certainly does, the decision for using one route instead of using another.

Yet in some situations, goals alone may not be enough to drive decision-making to the best behaviour. An agent may be faced with the problem of having to decide among several ways of reaching its desired state of affairs. The concept of utility represents the degree of satisfaction an agent would have with respect to choosing any of them. This could be implemented as a function that mapped a state to a real number, which would be associated to the degree of happiness, for instance. So, utility can be used as a means to trade off between different ways of achieving the same goal. In the case of having several goals, utility would provide a way in which the likelihood of success can be weighted up against the importance of each of them (Figure 3.5).

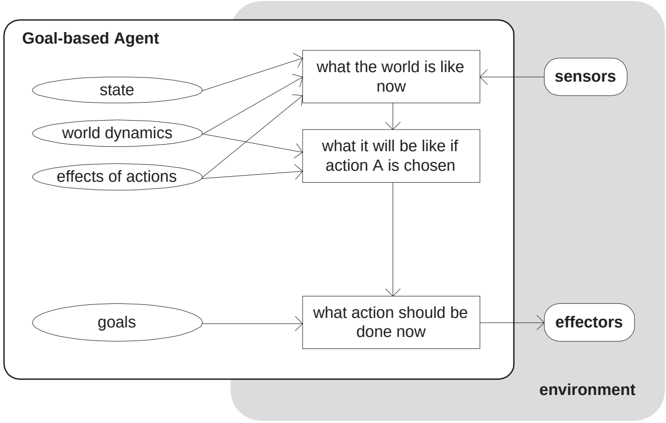
Goal-based Agent
state
world dynamics
effects of actions
goals
what the world is like now
what it will be like if action A is chosen
what action should be done now
sensors
effectors
environment

Figure 3.4: The goal-based agent,
adapted from (RUSSELL; NORVIG, 1995).

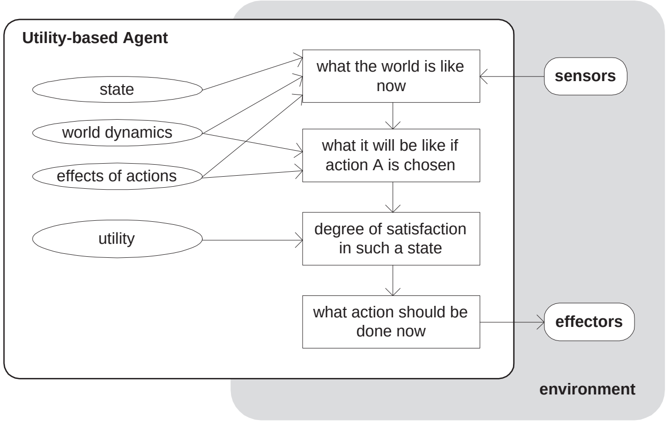
Utility-based Agent
state
world dynamics
effects of actions
utility
what the world is like now
what it will be like if action A is chosen
degree of satisfaction in such a state
what action should be done now
sensors
effectors
environment

Figure 3.5: The utility-based agent,
adapted from (RUSSELL; NORVIG, 1995).

- differently from the goal–based agent, *what action should be done now* results from an evaluation of the degree of satisfaction for executing certain action chosen out of different alternatives. For example, the driver could trade off between changing departure time and changing route in order to minimise travel time. Depending on the number of different possibilities, it can be an arduous task;

- *what action should be done now* is that which presents the highest degree of utility, thus providing the agent with most satisfaction.

3.5 Societies of agents: the multi-agent systems

A society of agents can be seen as a collection of autonomous entities that behave accordingly in order to achieve the goals they are designed to pursue. Also, they are able to interact both with one another and with its surrounding environment. Communication and co-operation are key concepts in such a scenario, and co-ordination of their behaviours allows for sharing knowledge, goals, aptness, and plans among them toward the solution of problems. In MAS each individual's behaviour results from the knowledge represented in its internal state, from its perception of the environment, and from its interaction with others. Although this perspective leads to thinking of a society of agents as a co-operative community, they may also have conflicting goals involving competitive behaviours. In either case, the following elements are of huge concern for the MAS design.

- the agents that live in the same environment;
- the environment itself;
- interactions between agents and between an agent and the environment;
- the society organisational methods.

In the majority of real applications, each agent possesses incomplete, uncertain, and partial knowledge of the environment and of its neighbourhood. This very uncertain nature will demand carefully monitoring of the task execution and frequently updating its course.

As suggested in (STEELS, 1990; FROZZA, 1997), agents can collectively exhibit an *emergent behaviour*. This concept relies on the idea that each single agent albeit having a very simple structure can contribute to more complex and efficient behaviour of the system as a whole. This approach has been used in a range of different applications and is mainly proposed to those with a huge number of interacting components (DROGOUL; FERBER, 1994). It has also been coupled with the Cellular Automata (CA) theory (CODD, 1968; SMITH III, 1969) as the relying approach for other applications (HALPERN, 2002).

Self-organisation is also an important concept related to society of agents. Creating a society that is capable of evolving dynamically and autonomously demands the implementation of efficient self-organisation mechanisms. Such a concept could be useful when a complex problem is to be solved by way of grouping individuals with different expertise. As reported in (FROZZA, 1997), self-organisation takes place through changing the topology of agents in an autonomous way both with respect to the environment, to one another, and to the internal model of each one. This allows for adapting themselves to the prevalent conditions of the system. Self-organisation can also be understood within the same framework as of the evolution theory (BAR-YAM, 1997).

3.5.1 Classifications and taxonomies for MAS

Much effort has been devoted to proposing classifications and taxonomies as there is a wide range of characteristics and applications that could be modelled and implemented by means of agent-based techniques. Some authors, specially in the earlier ages of the MAS society, have suggested different criteria for classifying and categorising such systems. This effort has been of paramount importance to better organise concepts, to aid identifying requirements for the whole designing process, and to support the creation of ground theories.

Among many ways of classifying autonomous agents, there is one found in (FROZZA, 1997) that groups agents into two categories according to the role played in the system.

- an agent can be seen as a *component* when it behaves as part of a system toward the completion of specific tasks. The system goal is divided into smaller and simpler sub-goals that are assigned to each component. So, the overall system performance is expected to be achieved on a co-operative basis;

- on the other hand, an agent can be understood as the *system* itself when it behaves on behalf of the user to achieve its own goal. There is no need either for co-operation or for interaction. In this case, the agent is fully autonomous and its behaviour may be not dependent on an external coordinator.

The aptness to solve problems and the architecture of an agent are also features that define two major categories of agents.

- the *cognitive agents*;

- the *reactive agents*.

Theoretically there exist well-defined boundaries separating these two types. However, in practical terms it is possible to develop systems coupling features of both types. In this way a single agent will present either cognitive or reactive behaviour according to the prevalent conditions of the environment.

As suggested in (OLIVEIRA, 1996; FROZZA, 1997) societies of agents can be classified according to many criteria such as the type of agents, the nature of the environment, and the behaviour of agents.

- according to the *type of agents* a system can be grouped into *homogeneous* and *heterogeneous*. The former characterises a society composed of entities of the same type, which means agents present the same architecture. The latter consists of agents of different types;

- the *nature of the environment* serves as a means to identify between *close* and *open societies*. Agents are fixed in the environment in a close society, whereas they are allowed to migrate throughout different environments in open societies. The latter is the abstraction approach used within mobile agent frameworks (KENDALL et al., 1998);

- the *behaviour of agents* is another criterion that identifies two possible societies. A *rule-based society* explicitly defines behaviour rules to be followed by all of its components. In a *society without rules* agents are allowed to follow their own behaviour. In the former case and given certain circumstances an agent is expected to behave in the same way, whereas it is not always true in the latter.

The social behaviour of an agent can also be classified with respect to the tasks it is expected to perform. This is presented in (SICHMAN; DEMAZEAU; BOISSIER, 1992) on the basis of two criteria, both in the perspective of the agent and in the perspective of the task.

- regarding their *capacity of performing a task*, an *autonomous agent* can easily adapt its behaviour to any kind of task. A *task-oriented agent*, on the other hand, is capable only of performing the specific task it is designed for;

- with respect to the *locality of a task* it is called *local* when a single agent is capable of performing it by itself alone. Otherwise, the task is partitioned and performed by multiple agents, which interact one another to achieve the desired results. In this case, it is called a *distributed task*.

Four possible social behaviours in MAS are also identified in (DEMAZEAU, 1991; FROZZA, 1997), which account for the fact that an agent behaviour can be changed from completely autonomous to specialised and tasks can range from local to distributed.

- *co-habitation* combines autonomous agents and local tasks. The agent performs the task individually albeit in the presence of other agents;

- *co-operation* combines autonomous agents and distributed tasks. Such behaviour can be necessary either when an agent cannot perform the whole task by itself alone or for the sake of efficiency. In this case, agents perform part of the overall task;

- *collaboration* combines task-oriented agents and local tasks. It concerns global goals that can involve all of the agents in the system and can be individually achieved. An issue brought about in such a society is how to choose the agent to perform a specific task;

- *distribution* combines task-oriented agents and distributed tasks. It concerns global goals that can only be collectively achieved by multiple agents. In this case, an important issue relies on dividing and distributing the task among agents.

According to Weiss (1996), multi-agent systems can differ widely in three key aspects namely the environment, the agent-agent and agent-environment interactions, and the agents themselves. For each of these important aspects, the author identify several dimensions by which MAS can be classified.

- some features of the *environment* could be used to qualify MAS such as *availability of resources* ranging from *restricted* to *ample*, *diversity* ranging from *poor* to *rich*, *uncertainty* and *predictability* ranging from *predictable* to *unpredictable*, and *dynamic nature* and *status*, which could range from *fixed* to *variable*;

- the *agent-agent* and *agent-environment interactions* is used to classify MAS with respect to the *frequency*, ranging from *low* to *high*, to the *persistence* ranging from *short-term* to *long-term*, to the *level of interaction* ranging from *signal-passing* to *knowledge exchange*, to the *patterns* such as *unstructured* and *structured*, to the *variability* ranging from *fixed* to *changeable*, to the *type of interaction* namely whether it is *competitive* or *co-operative*, and to the *purpose* involved, such as *random* and *goal-oriented*;

- regarding the *agents* within the system some criteria are proposed on the basis of features such as the *number of agents* or *granularity*, the *number of goals* assigned to each agent, the *compatibility between goals*, which can be either *contradicting* or *complimentary*, the *uniformity*, that means whether the agents are *homogenous* or *heterogeneous*, and *individual properties*, as well.

The environment definitely plays an important role in MAS. In this extent, Russell and Norvig (1995) suggest some features that must be considered in order to define agent-environment relations. The underlying idea is to identify whether the environment is *accessible* or *inaccessible*, *deterministic* or *non-deterministic*, *static* or *dynamic*, and *discrete* or *continuous*. Environments that are inaccessible, non-deterministic, dynamic, and continuous are the most challenging to be implemented and frequently encountered in real applications.

The next criteria are concerned with the learning mechanism in MAS, as presented in (WEISS, 1995). These are closely related to the information an agent is able to gather, which is expected to influence the decision-making process in many ways.

- a learning mechanism may have two basic *purposes*. It may be aimed at the improvement of a single agent, its skills and abilities. It may also be aimed at improving the interaction of all agents within the system, their coherence, and co-ordination;

- *locality* is concerned with the degree of distribution and parallelism of a learning mechanism. When only one of the available agents gets involved in the process the learning steps are neither distributed nor parallel. Learning can be ‘maximally’ distributed and paralleled when all agents within the system participate in the process;

- the *involvement* of an agent in the learning process is also presented as an important criterion. The involvement of certain agent may be not a necessary condition for achieving the pursued learning goal. However, in an extreme situation, the learning goal cannot be achieved without the involvement of such a specific agent. Refinement of this criterion is also possible if one considers other aspects such as duration and intensity of the involvement;

- as for the *interactions* required for the learning process, both agent-agent and agent-environment ones must be taken into account. Such a dimension can range from requiring only a minimal degree of interaction to being untractable without extensive interaction. This criterion could be further refined with respect to the frequency, persistence, level, pattern, and type of interactions.

It is important to bear in mind that this plenty of criteria represents the insights of researchers into the potentials of applying MAS in a wide range of different domains. Some classifications and taxonomies can overlap one another. Thus, it is reasonable to suggest that other criteria can derive from combination of those mentioned above, as well as from further refinement of ones on the basis of others. Nonetheless, understanding the relations between these dimensions would provide for valuable guidelines for deciding which type of multi-agent system is best or at least sufficiently well-suited to a given application. This problem is sometimes called the multi-agent system-application assignment problem, as stated in (WEISS, 1995).

3.5.2 Organisational structures

Identifying relations between agents is a crucial task in the field of multi-agent systems, and choosing an organisational structure is of paramount importance to modelling agent societies. As an attempt at aiding such processes, two possible organisational structures are proposed in (LABIDI; LEJOUAD, 1993; FROZZA, 1997) which might serve as excellent starting points in MAS designing.

- in a *horizontal structure*, all agents of a society are in the same level of involvement. For example, agents can be requested to execute different tasks in order to solve a common problem or to achieve a common goal;

- in a *vertical structure*, contrarily to the previous one, the agents are hierarchically disposed in different levels of involvement. This structure can be viewed as a hierarchy of horizontal structures. For example, the solution of a problem could be partitioned into simpler subproblems, which would be delegated to the agents at lower levels in the hierarchy.

3.6 Important issues in multi-agent systems

The increasing interest in applying agent-based technologies to a wide range of domains have stemmed a huge amount of research works with different focuses and aims. These efforts have given rise to discussions on a number of theoretic and practical implementation themes. Woodridge and Jennings (1995) group such issues related to the use of agents into three major areas of study, each of which concerning aspects that demand special care and attention.

- *theory of agents* is related to theoretical approaches that can be seen as the specification for an agent-based methodology; these serve to describe mathematical formalisms needed for modelling agents. An agent can be represented in terms of attitudes, such as belief and knowledge, and pro-attitudes, such as desires and intentions. The former representation is related to the information an agent possesses about the world, whereas the latter drives the agent's actions upon the environment;

- *architectures* represent models designed to support practical implementation. According to what has been stated in (WOOLDRIDGE; JENNINGS, 1995), they can be classified into three basic types, namely *deliberative*, *reactive*, and *hybrid*. Deliberative architectures are a classical approach to describe agents in a society. In such structures, an agent keeps a symbolic representation that explicitly represents every relevant fact within the world (JENNINGS, 1994). On the other hand, reactive architectures are based on a behavioural approach. Agents are described in a simpler way and do not possess neither a symbolic model of the world nor plans for their actions (BROOKS, 1991a,b). Finally, the third type is based on the previous ones. It seems to be more suitable to tailoring agent's behaviour for specificities of actions needed to achieve different objectives. The major idea behind an hybrid architecture is to enable agent's actions to result from two subsystems, namely a deliberative and a reactive, or from a combination of both. The deliberative subsystem keeps a symbolic representation of the world and makes use of elaborated task-planning and inference mechanisms. To the contrary, in the reactive subsystem the agent is able to choose a specific action from predefined events of the environment in a linear way. This avoids the use of complex reasoning mechanisms, as actions are taken from directly mapping events to possible reactions.

- *languages of agents* have moved researchers into the challenging task of creating agent-oriented programming languages, which are software systems designed to implement agents. Shoham (1990) proposes agent-oriented programming as a new paradigm for programming. The basis of this paradigm would be on the facilities offered to define agents as an entity constituted of mental attitudes, such as beliefs, desires, and intentions, for instance.

Besides the areas proposed in (WOOLDRIDGE; JENNINGS, 1995), Weiss (1996) also points out a number of challenging and open subfields concerning specification, implementation, handling, and assessment of MAS.

- communication languages;
- co-ordination mechanism;
- negotiation and co-operation strategies;
- organisation design;
- planning and diagnosis of tasks;
- problem decomposition and synthesis.

3.6.1 Learning

It is quite reasonable to say that learning plays vital role in MAS technology. This topic has been the subject of both theoretical and practical works as it is central in the reasoning process, and the technical community agrees it deserves special attention on its own right. Weiss (1996) identifies between two categories of learning.

- *single-agent* (or *isolated*) *learning* consists of a learning mechanism that does not rely on the presence of multiple agents. This is the basic approach adopted by the traditional AI;
- on the other hand, *multi-agent* (or *interactive*) *learning* requires the presence of multiple agents and their interactions in order to achieve an efficient learning. More specifically, it could be viewed as a process concerned only with situations in which several agents collectively pursue a common learning goal. It may also refer to situations in which an agent, albeit pursuing its own learning goal, is affected by other agents' knowledge, beliefs, intentions, and so forth.

The variety of possible forms learning can be thought of in MAS is certainly enormous. So, the effective implementation of such mechanisms demands for some considerations to be taken into account. Weiss (1996) also suggests the following criteria to better structure learning approaches.

- the *learning method* or *strategy* concerns the process itself, used by a learning entity. For example, rote learning, by repeatedly studying likely situations using memory rather than understanding, learning from instruction and by advice taking, learning from examples and by practice, learning from analogy, and learning by discovery. A major difference between all these methods relies on the amount of learning effort required;

- the *learning feedback* enables the assessment of the degree of performance achieved. Three basic kinds of learning are identified according to this criterion. The supervised learning specifies the desired activity and it is aimed at matching a desired action as closely as possible. The reinforcement learning specifies the utility of the actual activity and it is aimed at maximising such a utility. Finally, the unsupervised learning where no explicit feedback is provided. In this case, the learning entity needs to find out useful and desired activities on the basis of trial-and-error and self-organisation processes.

With respect to the criterion of learning feedback, it is assumed the performance level to be achieved is provided by either the environment or the agents themselves. With this respect, three possible types of feedback providers are identified in (WEISS, 1995).

- *teachers* are either the environment or agents that provide feedback in the supervised learning scenario;

- providing feedback in reinforcement learning is a responsibility assigned to *critics* that are, once more, either the environment or agents;

- in the case of unsupervised learning neither the environment nor other agents will provide the learner with any feedback. They act just as *passive observers*.

3.6.2 Communication

Integration within an agent society may be dependent on the communication abilities of its members. Thus, well-defined communication protocols are necessary to establish efficient and proper interaction between agents. Basically, communication can be implemented in two ways.

- *direct communication* happens when agents know each other allowing them to exchange data. The message-passing mechanisms are good examples of this kind of communication. The interaction happens on the basis of well-defined protocols, which specify the dialog process to be performed by the agents involved in the communication. Different protocols are defined to different kinds of interaction within the agent society. Contract net protocol, for instance, could be used to implement such a kind of communication mechanism (SMITH, 1988);

- to the contrary, *indirect communication* may be established between agents that have no previous knowledge about each other. In this case information is delivered to and collected from a common directory, which is accessed by every one within the system. For example, blackboard architectures are based on a data structure divided into different levels of information. Agents can write on or read from it. A well-defined scheduling mechanism is necessary to manage this processes, which should allow the consistency of information on the blackboard (ROTH, 1984).

Albeit perception is the basic mechanism through which an agent gathers information, communication can be used to improve this process. In some applications it is even more significant when the information is to be shared with others. There are three types of information, which are very likely to be exchanged among agents of a society.

- an agent's *knowledge* may depend much on its ability to sense the environment, on its expertise and skills, and on its communication facilities. Sharing this kind of information through communication may help others in many forms. For example, one could provide information about the environment to a 'blind' agent;
- in an agent society, *plans* or possible *solutions of specific problems* may be the privilege of a reduced number of individuals. Expertise and skills could be shared with others facing the same problem already solved by any member of the society, in a co-operative scenario for instance;
- in certain societies, either homogeneous or heterogeneous, agents may need to execute the same plan in order to speed up a system task or as a means to make the solution of a problem uniform, for instance. In such situations, the group of agents should share the same *plan choice*, which can happen through communication.

However, interaction is not only related to communication. It can also be identified with the purpose of controlling some sort of process. In a broad perspective, a control mechanism basically dictates and regulates data exchanging within the society. It can be either distributed among several agents or centralised in a single entity (FROZZA, 1997).

3.6.3 Co–operation and conflicts

Achieving a common goal is often one of the intentions for agent-based applications. When this is the case, conflict and co-operation are central concepts underlying agent interaction, and can limit the execution of simultaneous actions by different entities, as well. Conflicting situations is the subject of much research in the field of MAS (TEDESCO; SELF, 2000), and can basically be grouped into two types.

- the conflict is said to be *local* in the case it happens between only two conflicting agents;
- it is said to be *global* in the case several individuals are found to be in conflicting state.

In general terms, a conflict happens when one's interest can be conditioned on the behaviour of others. Béron et al. (1995) identifie some likely conflicting situations, which are easily understood on their own right, namely conflict of goals, conflict of results, and conflict of resources.

Coordinating activities and negotiating actions are important mechanics to cope with conflicts. They greatly influence task-planning as the order in which actions are executed or the instant at which execution starts may in any form avoid potential conflicting situations. As presented in (FROZZA, 1997), there are two basic approaches for planning that minimise conflicts, if not to eradicate them.

- when a conflict is diagnosed, the task of addressing it can be delegated to or be the duty of a single agent. Thus, it starts a *local planning* to autonomously sort the situation out;
- solving a conflict in the system may be also accomplished by means of ordering actions performed by multiple agents. In this case, a *global planning* is necessary.

On the other hand, negotiation aims to address conflicts between individuals through their consensual co-operation. It is particularly interesting for domains in which a group of interacting agents associate their efforts to achieve desired goals. Thus, conflicts may arise for the sake of differing aptness and skills among the group. A negotiation process can involve actions such as proposing, evaluating, changing, accepting, and rejecting a solution. In order to enhance efficiency and be successfully terminated, the negotiation should follow a protocol that facilitates the solution to converge (LABIDI; LEJOUAD, 1993; FROZZA, 1997).

3.7 Agent–Based Simulation

With the advent of computers, computer simulation has been widely used as an important tool for understanding and assessing systems of varying complexity and nature, and has become the bridge between theory and experiment. Educational, training, and entertainment purposes have also been among the subjects of much research in this field, which has gathered the attention and interest of scientists and practitioners of different disciplines.

The ability to represent system dynamics in a controlled virtual environment has encouraged the application of simulation techniques in many different areas, including natural sciences, engineering, industry, business and financial market, the government and the army, and social sciences. Albeit the computer simulation methodology has been relatively kept on its original basis, its widespread application and the improvement of computers in recent years, both in processing and in graphical interface, have stimulated the evolvement of different techniques, including agent-based simulation. One can refer to computational prototyping as a scientific discipline on itself, allowing a new way to bridge theory and experiments, and enabling to go beyond both.

Shannon (1975, 1992) presents in detail all the steps involved in a computer simulation study. Those can be basically summarised into three essential phases. First, the application domain is modelled through an abstraction process that identifies the relevant characteristics and relations of the problem components. Second, the model is translated to an executable form, and the experiments are executed to generate new information and simulation results. Finally, results are assessed and interpreted, and then can be used accordingly in the real system. It is important to bear in mind that these are parts of an iterative process where modelling plays an essential role, and should be addressed with special care. In such an iterative process, one can even formulate and test new models with simulation techniques.

Not surprisingly, modelling has been the main focus of most advances in computer simulation. Virtually, any programming language could be used to translate simulation models into executable code. Nonetheless, some languages such as Simula (BIRTWISTLE et al., 1975), GPSS (SCHRIBER, 1990), and Simscript (CACI PRODUCTS COMPANY, 1983), which are specifically devised for general simulation purposes, offer additional facilities for dealing with model specificities and enhance computational prototyping. More complex simulation frameworks have also been implemented as DOSE (MAK, 1991), VISE (LINDSTAEDT, 1995), and SIMOO (COPSTEIN, 1997), which incorporate the concepts of Visual Interactive Modelling (VIM) and Visual Interactive Simulation (VIS) (FREITAS, 1994). The use of object-oriented development gave a remarkable contribution in modelling techniques and, recently, the concept of autonomous agents has further boosted the development of the computer simulation field (CONTE; GILBERT; SICHMAN, 1998).

Besides the concepts of encapsulation of properties and behaviour of objects, agent-based modelling promotes an adequate means for representing autonomy and cognitive capabilities within the system entities. Two perspectives were initially assumed to base what has become the agent-based simulation field (KLÜGL, 2001). At a first point of view, agents are responsible for carrying out simulation capabilities (even though the model is devised on the basis of other techniques). The second approach is built up on the basis of agent-based modelling, which represents the application domain in terms of agents and their interactions. This has become the main stream adopted by the agent research community as the seed for cross-fertilisation between computer simulation and multi-agent systems, stimulating innovative research at the intersection between these two multidisciplinary areas (CONTE; GILBERT; SICHMAN, 1998). In this way, much research effort has also been devoted to the development of simulation environments that support the design of agent-based simulation models; a comprehensive description and comparison of some of such environments is presented in (ADAMATTI, 2001).

Two important issues arises in the relatively recent ages of the agent-based simulation field. First, MAS modelling is still seen with a certain skepticism when compared with traditional approaches, such as equation-based techniques. They both can ultimately be translated into an executable form and differ basically in terms of the abstraction used to build the model and the way they are executed (PARUNAK; SAVIT; RIOLO, 1998). In agent-based modelling, the model consists of a set of agents and execution consists of emulating their behaviours and interactions. In equation-based modelling, the model is a set of equations and the execution consists of evaluating them. Parunak, Savit, and Riolo (1998) then conclude that simulation is a general term that applies to both methods, which are distinguished as agent-based emulation and equation-based evaluation.

Whether to choose one or another simulation technique will very much depend on the purpose of the study and the nature of the problem. For example, time-dependent application domains may demand for efficient simulations in terms of execution time. In this way data can be generated and assessed on a timely basis and practitioners can intervene on the real system accordingly. On the other hand, when modellers and simulationists are mainly concerned with the application domain abstraction, the choice would certainly be better if headed to a more expressive approach (though inefficient in terms of execution time), such as agent-based modelling. The detailed description of the entities of a system and their interactions can contribute enormously for the understanding of its dynamics. The need for such a trade-off will be always present in any simulation methodology.

Another interesting issue is that there is still a controversial discussion as to whether adopting a more complex approach in the description of the agent reasoning capabilities is an adequate approach for agent-based simulation. Traditionally, emergent functions are modelled only among (simpler) reactive agents, as opposed to (more complex) cognitive agents able to deliberate about their joint goals and plans for all their collective activities. However, some works (CASTELFRANCHI; CONTE, 1992; CASTELFRANCHI, 1997) strongly question such an opposition and defend that collective intelligence and emerging functionalities must also be modelled among cognitive agents with limited knowledge and rationality, and are not able to understand, predict, and dominate all the global and compound effects of their actions at the collective level. The concept of Cognitive Emergence (CE) is then presented to designate the emerging dynamic processes of systems formed by cognitive agents (CASTELFRANCHI, 1998). Castelfranchi (1998) also argues that certain emerging phenomena in some complex domains, such as social organisations, cannot be explained without CE. In addition, the author claims that CE has critical importance in the process of immergence, that is how the resulting emergent structure of the system changes back the properties and then the behaviours of each individual element at a microscopic level.

3.8 Potential Applications of MAS in Traffic and Transportation

Examples of MAS applications are briefly presented in this section. The aim is making a survey on what has been done toward improving software tools for assessing traffic and transportation systems by way of agent-based approaches. Curiously, there is a number of examples reported in the literature, which are mainly concerned with traffic management and control systems, as well as with the microscopic representation of movement and driver behaviour. Nonetheless, there is also an increasing number of examples that address other aspects of traffic and transportation. Although special attention has been given to those related to modelling the traffic phenomenon, some other examples were found to be worth mentioning.

3.8.1 Traffic Management and Control Systems

Traffic management and control have been traditionally elected as an application ground where scientists can develop, test, and apply their approaches and theories. The need for certain degree of autonomy and intelligence in the growing area of Advanced Traffic Management Systems (ATMS) makes agent technologies suitable to model such a domain.

Haugeneder and Steiner (1994) applied agent-oriented techniques to address urban traffic control (UTC) issues. The authors grouped UTC systems into three different interacting levels, namely the traffic flow control, the traffic guidance, and the integrated traffic management. Focus was given to the second level, emphasising traffic guidance on individual basis. The MECCA/UTC system was presented as a domain-specific application built under the MECCA framework (HAUGENEDER; STEINER, 1993), where agents were implemented in the MAI2L language (STEINER et al., 1993). The application of MAS to traffic control was also explored in the work by Gabrié et al. (1994). Their aim was at testing and evaluating the adequacy of the approach used in dMARS (Distributed Multi-Agent Reasoning System) to real world problems.

A similar concern underlies the example reported in (PIRES; DIAS; BELO, 1997). Their model was devised as a means for investigating and verifying the applicability of multi-agent systems to the field of traffic control systems (TCS). Important issues such as control distribution, co-ordination and co-operation protocols, as well as system modularity were approached. Different kinds of agents with specific skills were specified and geographically distributed within zones of a city. Performance measures included, for instance, duration periods, the current traffic signal state, average duration of green time, and the messages associated to co-operation processes among simulated junctions. The functional architecture of the traffic agent was organised into three layers with different purposes, namely communication, inference and knowledge representation, and interface. Linda (CARRIERO; GELERNTER, 1989) was used to support the implementation of the distributed environment.

In (BAZZAN, 1997), a game-theoretic approach was used to confront the drawbacks of co-ordination in traffic control systems. The author proposed a model to yield a co-operative environment where controller agents were able to co-ordinate their actions. While overcoming the disadvantages of traditional decentralised approaches, the traffic control co-ordination was achieved with reduced communication. Game theory was also the basis for the work reported in (CHAMPION; MANDIAU; ESPIÉ; KOLSKI, 2001; CHAMPION; ESPIÉ; MANDIAU; KOLSKI, 2001).

Van Katwijk (2000) presented two examples through which he analysed the potentials of autonomous agents for modelling traffic control and management instruments. The author asserted this approach allowed for tuning the action of each single device toward a collaborative traffic control system, where co-ordination on the level of devices might reduce the need for co-ordination on higher levels of control centres. His work was inspired by Bazzan's previous research (BAZZAN, 1997) and approached co-ordination issues on the basis of communication and negotiation among controller agents.

The focus underlying the TraMas model (FERNANDES; OLIVEIRA, 1999) was concerned with using agent-based strategies for controlling traffic signals in a distributed and co-operating fashion. The model was presented as an alternative to traditional centralised-based approaches. The system was represented in terms of roads, vehicles, traffic signals, and traffic controllers, all implemented by means of object-oriented programming. Each traffic controller was associated to the junction where it was situated and was on duty for locally controlling traffic signals. The authors adopted a reactive approach based on the Brooks' subsumption architecture (BROOKS, 1991b), with three hierarchical levels of behaviour (see Figure 3.6). Co-operation was designed to take place by means of a simple communication protocol between adjacent traffic controller agents. The microscopic simulation environment, Magoo (FERNANDES, 1998), followed the Cellular Automata approach and was implemented in the Java language. A distributed approach for agent simulation of traffic systems is also proposed and reported in (TANDAYYA; ZOBEL, 2000a,b).

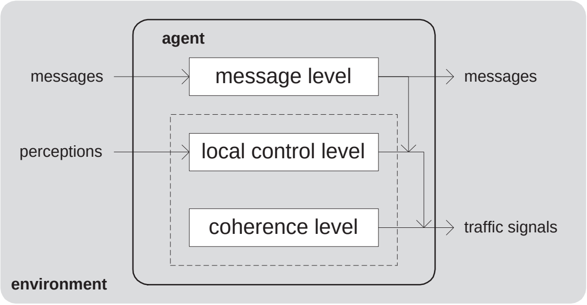

Figure 3.6: The basic agent architecture for the TraMas model.

Real applications of agent-based techniques in Intelligent Traffic Management Systems were reported in (CUENA; OSSOWSKI, 1999) and (HERNÁNDEZ; OSSOWSKI; GARGÍA-SERRANO, 2000). These examples were carried out within the European KITS and the Spanish TRYS projects (CUENA; OSSOWSKI, 1999). Two multi-agent systems with the purpose of performing decision support for real-time traffic management in urban highway networks, namely TRYS and TRYSA, were implemented on the basis of different approaches and compared. On the one hand, the centralised model used in TRYS relied on the knowledge approach presented in (CUENA; HERNÁNDEZ; MOLINA, 1996). Agents were endowed with different types of knowledge, organised within knowledge units that might lead behaviour in different perspectives. All control plan proposals resulted from individual reasoning of each controller and a coordinator was designed to mediate the process and to build global and coherent signal recommendations for the whole network. On the other hand, TRYSA implemented a decentralised model where agents coordinated their tasks through the structural co-operation approach presented in (OSSOWSKI; GARCÍA-SERRANO, 1998). Basically, the central coordinator of the former model was replaced by the socially bounded autonomy that agents enjoyed within the structural co-operation of the latter. This methodology was applied to the urban motorway networks around Barcelona, as reported in (HERNÁNDEZ; OSSOWSKI; GARGÍA-SERRANO, 2000), and around Madrid, as reported in (CUENA; OSSOWSKI, 1999). The centralised approach of TRYS was found to promote efficiency for real-time operation, whereas the decentralised approach used in TRYSA promoted scalability.

3.8.2 Traffic Microscopic Simulation and Driver Behaviour

The growing complexity of abstraction levels used to represent the traffic domain has encouraged the development of microscopic simulation models. Burmeinster, Doormann, and Matyls (1997) claim that existing microscopic traffic simulation models can be enhanced with the agent concept. According to the authors, the major advantage of such an approach relies on a better commitment to the system ontology, mainly with regard to the driver representation. In their example, the domain was represented as a society of cognitive agents featuring the BDI model proposed in (HADDADI, 1993). Such a structure was based on the COSY architecture (BURMEISTER; SUNDERMEYER, 1991) as depicted in Figure 3.7, and the simulation framework was implemented under the DASEDIS environment (BURMEISTER, 1993).

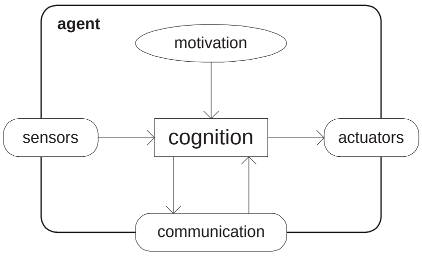

Figure 3.7: The COSY agent architecture.

SITRAS (**S**imulation of **I**ntelligent **TRA**nsport **S**ystems) as presented in (HIDAS, 2000), is a microscopic transport simulation model that has been developed since 1995. It was devised with the objective of providing engineers and practitioners with a general evaluation tool for ITS applications, such as congestion and incident management, public transport priority, and dynamic route guidance. The microscopic simulation model was designed in terms of DVO units, which are driver-vehicle objects. Although the DVO has not originally been implemented on the basis of agent concepts, the author defended that it can in fact be seen as an autonomous agent. Hidas (2000) based his analysis in the light of the definitions given in (TOKORO, 1994) and the agent model proposed in (CHAIB-DRAA; LEVELSQUE, 1994; RASMUSSEN, 1986). The author concluded that a DVO possessed most characteristics of autonomous agents. Basic behaviours such as lane changing, merging, and car following while travelling between an origin and a destination were implemented within each driver-vehicle unit. In order to allow the evaluation of the effects that DRGS (Dynamic Route Guidance Systems) might cause on the overall system performance, two main classes of driver-vehicle objects were identified, namely the unguided and the guided ones. This theme was further developed in (HIDAS, 2001).

The framework for anticipatory traffic forecast proposed in (KLÜGL et al., 2000) was aimed at providing drivers with information about the network future state (through VMS, for instance). In that work authors suggested that such an approach might enhance decision-making and improve the quality of the trip. The model required a more realistic representation of reasoning, as decisions influenced by exogenous information were expected to considerably effect the level of recurrent demand. A two-layered structure was proposed for the driver structure. Basic perception-reaction behaviours, such as car-following, lane-changing, and merging, were implemented in the tactical layer. The strategic one was designed to support more elaborated cognitive mechanisms. The authors suggested that BDI-based models would be ideal to represent such a strategic layer, as similarly discussed in (ROSSETTI; BAMPI; LIU; VAN VLIET; CYBIS, 2000a; ROSSETTI; LIU; VAN VLIET; BAMPI; CYBIS, 2000)

The microscopic simulation framework in (KLÜGL et al., 2000) was based on the Nagel and Schreckenberg's model inspired by the Cellular Automata (CA) theory (NAGEL; SCHRECKENBERG, 1992), which was extended and implemented as a multi-agent system in the SeSAm (**S**h**e**ll for **S**imulated **A**gent Syste**m**s) environment (KLÜGL; PUPPE, 1998; DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY WUERZBURG, 2002). Nonetheless, further work has been carried out to improve this research as reported in (BAZZAN et al., 2001). They presented a domain-specific framework to support the microscopic simulation of driver agents on the basis of an object-oriented approach. The environment representation has also followed the CA approach whereas the moving elements were designed to base more sophisticated reasoning mechanisms. Another example using CA was reported in (HERTKORN; WAGNER, 2000).

Dia and Purchase (1999) similarly envisaged the potential application of intelligent agents to modelling dynamic driver behaviour. A framework was conceptualised (DIA, 2000, 2001, 2002) to evaluate the effects of ATIS on the performance of transportation systems. This ongoing research started by a behavioural survey of congestion on a traffic commuting corridor. The data gathered from that survey served as the basis for modelling driver agent features, such as travel behaviour, personal preferences, and goals. The preliminary results were reported in (DIA, 2002). This approach consists of coupling the dynamic driver behaviour with the microscopic traffic simulation. The cognitive agent proposal was inspired by the works presented in (SHOHAM, 1993; THOMAS, 1993) albeit it has not, as yet, been actually implemented. PARAMICS (QUADSTONE LIMITED, 2002), which is a commercial microscopic traffic simulation framework, was elected to perform the traffic simulation.

In the research reported in (BRUGGEMANN; LEHMANN, 2000), each individual was endowed with action-planning and decision-making capabilities. The agent structure was designed so that drivers could behave rationally on the basis of a multi-criteria assessment model. Urban mobility was described in terms of supply-demand systems, where the execution of individuals' activities affects traffic generation. The simulation framework proposed basically comprises three stages, which include scheduling activities on week basis, performing scheduled activities, and evaluating performance measures. The behavioural model, inspired by the work presented in (COHEN; LEVESQUE, 1986, 1990), was proposed as an alternative to using traditional optimisation techniques.

Although FLOWSIM (**F**uzzy **LO**gic enhaced motor**W**ay traffic **SI**mulation **M**odel) (WU; MCDONALD; BRACKSTONE, 1998) did not explicitly use the concept either of agent or of multi-agent system, it relied on a microscopic representation of drivers endowed with reasoning capabilities. A fuzzy inference approach was used to support the decision-making on both car-following and lane-changing situations. The model was suited to assess various ITS measures, such as Autonomous Intelligent Cruise Control (AICC) and implemented within microscopic traffic simulation framework.

3.8.3 Other Applications

Apart from the two major areas mentioned above, the growth of the ITS field has encouraged much research. It has deserved an increasing interest from both AI and MAS communities owing its complexity and very dynamic nature. It has been used as a ground where theories and approaches have been tested.

After formally specifying a cognitive agent architecture, Haddadi (1996) presented a hypothetical scenario in the domain of transportation and shipping to demonstrate her theory. In order to meet costumers demand, a shipping company needs to optimise the service allocation for its fleet and sometimes to recur to private sub-contractors. The company coordinator and the sub-contractors were modelled as BDI agents on the basis of the theory proposed by the author. In this work, focus was given to communication as a means to achieve co-operation in multi-agent systems.

A multi-agent model was devised and presented in (BURMEISTER; HADDADI; MATYLIS, 1997) to analyse different organisational structures, fleet composition, and technical apparatus for vehicles within a car pooling station. The work was motivated by the increasing traffic volume throughout road networks and the need for optimising capacity usage, mainly in urban areas. The scenario was modelled and implemented under the dMARS environment (KINNY, 1993). Negotiation and control mechanisms for successful dialogs are some issues addressed in that work, which followed the generic protocol proposed in (BURMEISTER; HADDADI; SUNDERMEYER, 1995).

Rather than focusing on the representation of all physical elements found in the real traffic system, the model suggested by Adorni and Poggi (1996) were specially concerned with the guidance process itself. A route guidance system was designed in terms of distributed and interacting modules working on an on-board Road Map Knowledge Base. Each of such modules was modelled as an agent supporting different capabilities of the system, ranging from specifying the car position and planning routes to providing commands to actuators for autonomous car driving.

Garcia-Bulle (1990) applied the multi-agent concept to address the problem of network equilibrium. Instead of using aggregate functions to describe the behaviour of the flow as in traditional approaches, the author elaborated a framework of individual objective functions. In his model, agents were shippers of a good in a given network. Each agent was designed to determine the flow to be shipped in order to maximise an objective function accounting for the actions of the rest of the agents. Among the examples given by the author, the model was also used to tackle the traffic equilibrium problem.

The Pedestrian Crowds model proposed in (JIANG, 1998) was devised to serve as an aid to urban planning and designing. The interest in mentioning this work resides in its representation of the application domain by means of multiple agents moving from origins to destinations throughout a bounded-resource environment. This approach was inspired by the work reported in (DROGOUL; FERBER, 1994) and implemented under the StarLogo suit (MIT MEDIA LABORATORY, 2002). In a very similar work, Dijkstra and Timmermans (2000) modelled people in terms of the cognitive structure proposed in (FISCHER; MÜLLER; PISCHEL, 1998). In both examples CA was used to represent the environment.

Modelling driver behaviour within driving simulators is another application to which agent-based techniques has been found to have great potentials. Agents deployed in this kind of experiments are intended to interact with the subject driver in order to make its virtual environment more realistic and less predictable (AL-SHIHABI; MOURANT, 2001). Al-Shihabi and Mourant (2001) presented a framework for modelling different kinds of human driving behaviour to be used in autonomous vehicles within a driving simulator environment. The behaviour of the interacting virtual drivers was modelled through Fuzzy Logics. El Hadouaj, Espié, and Drogoul (2000) addressed conflicts between drivers within such environments. In their model, interaction decisions are made on the basis of an analysis of the traffic condition in the area around the vehicle, which the authors designated as the the driver's control field. This approach relies on the work presented in (SAAD; SCHNETZLER, 1994) and is focused on testing and refining behaviour models, testing man-machine interfaces, and testing driving aid equipments.

In the examples reported in (BOTELHO; RAMOS, 2000) and (BOTELHO, 2000), authors were specially interested in demonstrating theories on agent interaction and communication. The architecture was conceptualised in a way so that agents were endowed with the concept of emotions, which was central to the interaction behaviours. Their approach for interaction control and communication was applied within the Monitorix framework, a video-based multi-agent system aimed at traffic monitoring and surveillance.

3.9 Summary

By developing computer applications in either purely software or hardware-software design and development, one's goal is that those applications must be able to perform the tasks they were designed for in an autonomous way. Autonomous interaction has been even more desired, and turning it into practice in different levels of complexity has been the motivation of much research in the field of multi-agent systems.

In their relatively recent ages, the interest in the emerging agent-based techniques has presented a crescent growth and their practical utility to model and simulate a wide range of differing domains is reckoned to be evident. Since then, MAS community has worked toward formalising agent theories, classifying and organising their types and features, and devising ways for their practical implementation. Furthermore, what makes this field even more stimulating is its commitment to the ontology of systems. This very feature gifts MAS with the ability of representing different levels of complexity. From pure reactive to pure cognitive architectures, their suitability to interact with others and with the environment, to communicate, to learn and to plan turns agents into a powerful tool to model domains composed of geographically and functionally distributed entities.

Although advances in the processing power and memory of contemporary computer architectures have reached high-level standards, turning complex models into practical implementation is, as yet, a difficulty to overcome. From a result-driven perspective, contrary to the process-driven one, describing reasoning and decision-making in a detailed level may become much more complicated, though. Such a level of complexity has still been relegated to domains represented by means of a reduced number of agents, whereas the reactive approach has been preferred otherwise. Nonetheless, hybrid architectures have been suggested to address the cognitive approach drawbacks.

The complexity of today's transportation and traffic systems has definitely reached very concerning configurations, and representing uncertainty and variability within simulation and assessing models is even more imperative. As physical modifications become more and more unpractical, deploying ITS solutions brought about the need for autonomous mechanisms that can in any manner lead the usage of roads' limited capacity to converging optimum levels. This scenario becomes suitable for being represented by means of the abstraction premises of MAS, and such a practice has been verified in the literature.

There is already a considerable body of work presented in the literature aimed at practically applying multi-agent systems to the specific domain of traffic and transportation engineering. Traffic problems have always motivated researchers from a wide range of disciplines for the most varied interests. If it is not for the challenging issues this field poses, at least the relevance of its social and economic role may explain so. Two main groups of examples can be identified, namely those dealing with management and control systems and the ones devoted to representing movement in a microscopic way. The former group is traditionally elected as MAS has been already applied to other control systems scenarios, such as air traffic control, railway systems, industry, and process control theory in general. A real world's application is reported in (HERNÁNDEZ; OSSOWSKI; GARGÍA-SERRANO, 2000).

The latter group is specially concerned with the microscopic representation of movement. Vehicles and drivers have been traditionally dealt with indistinguishably as a vehicle-driver unity. Also in most agent-based models the environment is represented by means of the cellular automata approach (CODD, 1968; SMITH III, 1969), which seems to be much simplified when compared with vectorial models. As in (BAZZAN et al., 2001), for instance, such a simplification is desired and seeks to address the drawbacks of other approaches. It is aimed at integrating the agent-based model within an on-line simulation loop, where time dependence is a conditioning factor. Yet, in general terms, the goal has been the microscopic representation of movement. Nonetheless, ITS technologies have recently led some authors to envisaging proper representation for human behaviours, which brings decision-making to the driver level. This seems to be an adequate approach to represent different driver tasks, others than only driving. This may certainly increase complexity, though.

Another group not so specific encompasses diverse applications, which demonstrates the potential of MAS and its ability to represent ITS in different levels as already suggested in (BOUCHEFRA; REYNAUD; MAURIN, 1995; ROSSETTI; BAMPI, 1998a). From modelling interaction with service providers to on-board route guidance systems, as well as implementing personal assistants are some other examples found in the literature.

All of these examples show the potential of MAS to handle modern traffic and transportation scenarios. Proper data structures and algorithms provide for robustness and scalability, and constitute the natural abstraction to model, to simulate, and to assess the new performance measures. Contrary to former result-driven approaches, agent-based techniques are found to be very suitable process-driven methodologies to cope with the Intelligent Transportation Systems reality.

4 BDI: A COGNITIVE APPROACH FOR MAS

4.1 Overview

One great advantage of cognitive models is their ability to represent the reasoning mechanism. This provides for an ideal framework to understand more complex decision-makings, which are inherent in human beings. This very natural feature has motivated researchers from philosophy, psychology, social sciences, and anthropology to co-operate with computer scientists. Such a synergy has contributed to the design of adequate data structures and efficient algorithms that allow for the implementation and computation of their theories and formalisms. Definitely cognitive approaches privilege the representation of processes rather than focusing likely results.

As cognitive models cannot count on the simplicity of reactive architectures, they have been widely applied to societies of few agents only. Due to their complex nature and representation of knowledge, applications encompassing a larger number of reasoning entities are seen as interesting challenges. They are even more challenging when constraints, such as time, are to be overcome.

BDI (**b**eliefs, **d**esires, and **i**ntentions) is a cognitive approach that basically relies on mental states and their relations. As many other cognitive models, it has favoured an accurate representation of the reasoning process to the detriment of higher abstraction that eases implementation. This gave rise to the so-quoted ‘gap’ between the theory and its practical implementation. In this chapter the syntax and semantics for beliefs, desires, and intentions are presented as a means to model motorist reasoning. BDI is believed to be the ideal tool for simulating and understanding human behaviour and decision-making within today’s traffic and transportation scenarios.

4.2 Beliefs, desires and intentions

The BDI model used in this thesis relies on the formalism suggested by Rao and Georgeff (1991). The authors present their theory on the basis of the Bratman’s (1987) work, which deals with intentions as an important element for rational reasoning as beliefs and desires. Three important aspects are worth to be mentioned, as pointed out by Rao and Georgeff (1991), in order to base further discussions on their formalism.

- firstly, as stated by Bratman (1987), intentions are treated as first-class citizens on a par with beliefs and desires. Contrary to some reductionism theories, intentions are considered as important as beliefs and desires in the reasoning process;
- secondly, Rao and Georgeff (1991) distinguish between the choices an agent has over the actions it can perform and the possibilities of different outcomes of an action. In the former case, an agent can choose among outcomes of actions. To the contrary, in the latter case, it is the environment that determines the outcomes that a course of action will bring about. As a possible interpretation for this premise, one may consider that an agent chooses an action accounting for the possible outcomes it believes the action can bring about. However, it is the environment dynamics that dictates the outcomes that really result from executing that action. This allows for environment dynamics and nondeterminism;
- finally, an interrelation between beliefs, desires, and intentions is specified. This is aimed at avoiding many of the problems that are usually associated with possible-worlds formalisms, such as committing to unwanted side effects.

Beliefs, desires, and intentions constitute the main components in a BDI architecture. Rao and Georgeff (1991) present an alternative possible-worlds formalism for BDI models, which relies on these crucial components and the premises mentioned above. Such a formalism resembles the Computational Tree Logic (CTL*) (EMERSON; SRINIVASAN, 1988) to describe the concept of possible-words.

In the BDI formalism, the dynamics of the system is captured in a temporal structure, called a time tree. This structure is composed of both a single past and a branching time future. Basically, a time point can be seen as a specific moment in time that allows the agent to be characterised by its state (see Figure 4.1). Thus, a particular time point in a particular world is called a situation, which can be seen, for instance, as the scenario the agent is involved in at a particular moment in time.

The branches in a time tree can be viewed as representing the choices or options, which are available to the agent at each moment in time and that map to a possible future state. Each branch is associated to an event, which is, on the other hand, actions an agent can perform. Therefore, event types are responsible for transforming one time point into another. *Primitive events* are directly performed by the agent and uniquely determine the next time point in a time tree. *Non-primitive events*, contrary to the previous ones, refer to non-adjacent time points. They could be interpreted as a perspective of a far future, and have the potential for being decomposed into primitive events. In this way, they can be used to model hierarchical plan development. Nonetheless, it is important to bear in mind that the execution of an event can either be successful or not. Thus, the execution of an event does not mean necessarily its execution should be successful.

An agent, regarding its dynamic nature, has to act upon the environment in order to achieve its objectives. Thus, it is necessary to select appropriate actions or procedures to yield the effects it believes will result in the desired objectives. The design of a *selection function* allows the agent to choose an action, from the various options available, which will enable it to achieve its goals. Basically, this is the basis for the interrelation among beliefs, desires, and intentions.

Beliefs can be seen as the representation of what the agent effectively knows about the world, both dynamic and static aspects. As to selecting a course of action, there are two types of input data required by the selection function mentioned above. On the one hand, it is essential to acquire information about the state of the environment, which is basically done through sensing actions. However, such information may not be enough to capture the dynamic aspects of the environment, those related to how the environment is evolving over time, as well as which effects are produced by acting in a certain way, for example, as in inaccessible traffic systems. The beliefs, on the other hand, provide the agent with a cognitive representation of the world. Such an internal model is updated after every sensing action, and so can be used to deduce the state of the system in a broad perspective.

The motivational state of an agent is represented by its *desires*. It is quite intuitive that the existence of an agent within an environment has an end. Thus, the agent also needs information about its goals and about what is necessary to accomplish them. Rao and Georgeff (1991) distinguish between goals and desires in that while desires are inconsistent with one another, the goals must be consistent. Also, the agent should believe that the goal is achievable. Such a relation is referred to as the *property of realism* (COHEN; LEVESQUE, 1986). Therefore, goals are chosen desires of the agent that are consistent. For example, an agent might have both the desires of going to work and of going to the beach on a working day, which are inconsistent with respect to each other.

Considering that the objectives or priorities of an agent could be generated instantaneously or as a result of a function, there is no reason why they would require a state representation. However, some studies (RAO; GEORGEFF, 1995) have shown that the way this selection is implemented and the approaches that are assumed can bring about unwanted situations. For example, due to the high demand for accessing the selection function, an agent could be unable to accomplish a certain objective. Therefore, it is important to achieve a trade-off between too much reconsideration and not enough.

This way, *intentions* represent the state that an agent has committed to attempt to realise. In other words, to cope with the unwanted situations as already mentioned, intentions represent the currently chosen course of actions. Similarly to the requirement for belief-goal compatibility, the intentions of an agent must be compatible with its goals. In other words, the agent only can commit to some course of actions if it is one of the agent's goals. Regarding this process of choosing to accomplish a certain goal, one could identify many types of commitment strategies. Such a classification is important to characterise and analyse different reasoning behaviours of agents.

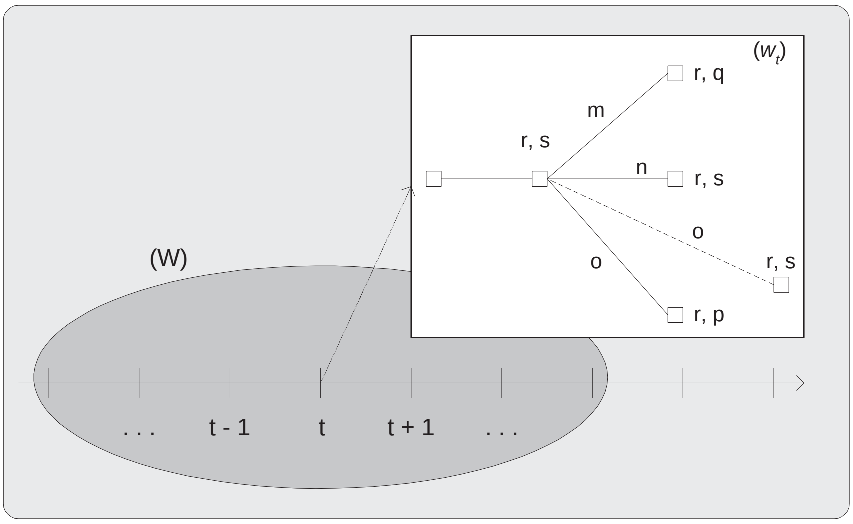

Figure 4.1: Possible world of an agent as a time–tree structure.

The conceptual architecture of a BDI agent is depicted in Figure 4.2, and briefly described by Wooldridge (WOOLDRIDGE, 1999) as follows.

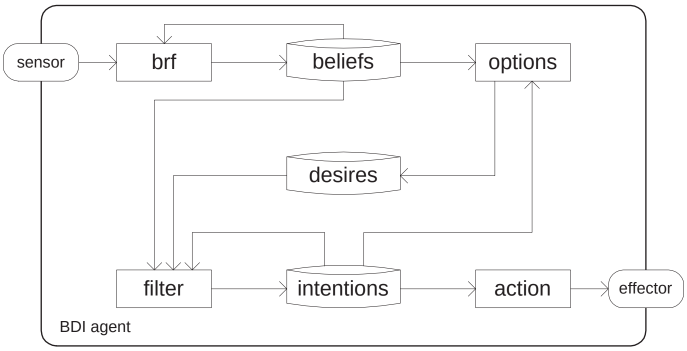

Figure 4.2: Conceptual architecture of a BDI agent, adapted from (WOOLDRIDGE, 1999).

At every perception from the environment, the agent's base beliefs set is updated. The new configuration of beliefs is performed by a *belief revision function* (BRF), which is responsible for preserving the consistence of the agent's beliefs. An options function determines the options available to the agent, which are its desires. This function receives as inputs the current configuration of the beliefs set, as well as the agent's current intentions. As further discussed in (GEORGEFF; LANSKY, 1987; GEORGEFF; RAO, 1996), an agent is equipped with a library of plans that are used to perform means-ends reasoning. Deliberation is achieved on the basis of instantiating meta-descriptions of plans, which generates the agent's options and are able to modify its intention structure dynamically at run time. The desires represent possible course of actions available to the agent, and a simplification is generally made in the sense that conflicting desires are discarded and only non-conflicting ones (the goals) are considered. A filter function representing the deliberation process determines new intentions on the basis of the agent's current beliefs, non-conflicting desires (goals), and the intentions currently being performed. The intentions represent those states of affairs that an agent has committed to trying to bring about. An action selection function then executes the next action the agent must perform on the basis of its current intention.

4.3 The BDI logics

Practically, the first step to model a multi-agent system is to choose a logic language to describe the agent's behaviour and interactions. The notation used to describe the various components of the language is borrowed from the ones presented in (RAO; GEORGEFF, 1991; HADDADI, 1996; RUSSELL; NORVIG, 1995). The syntax and the semantics are informally presented in this section.

Rao and Georgeff's (1991) formalism was presented as an extension of the CLT* logic as mentioned before. The authors suggested such an extension in two ways. First, a first-order variant was proposed. Second, the logic was extended to a possible-worlds framework by introducing modal operators for beliefs, goals, and intentions. It is important to note that a simplification was made, as only goals are used to the detriment of inconsistent desires. Thus, beliefs, goals, and intentions are represented as beliefs-, goals-, and intentions-accessible worlds. The authors also distinguish between two types of formulas, namely the state and path formulas. State formulas are evaluated at a specific time point, whereas path formulas are evaluated over a specific path in a given world. Practically, whether a formula is a state or a path one can be easily identified from its semantics.

Considering a given path formula ψ, it is said to be *optional* if, at a particular time point of a given world, ψ is true for at least one path emanating from that point. On the other hand, if ψ is true for all the possible paths, the formula is said to be *inevitable*. E and A are used to designate optional and inevitable paths, respectively. This representation obeys the convention of CTL*, as suggested in (RAO; GEORGEFF, 1991). The standard temporal operators, can be applied over both state and path formulas. A list of the components of the logical language is given next, following the same structure presented in (RAO; GEORGEFF, 1991; HADDADI, 1996).

- propositional connectives: $\Rightarrow$, $\Leftrightarrow$, $\wedge$, $\vee$;
- quantifiers: $\forall$, $\exists$;
- equality operator: $=$;
- negation operator: $\neg$;
- operator symbols: $\mathsf{succeeds}$, fails , does , $\mathsf{succeeded}$, failed , done ;
- modal operators: BEL , GOAL , INTEND ;
- path operators: E (optional) , A (inevitable) ;
- temporal operators: $\bigcirc$ (next), $\Diamond$ (eventually), $\Box$ (always) , $\mathscr{U}$ (until) ;

- action operators: $|$ (disjoint), ; (sequence) ;
- a set Constants of constant symbols: $Agent_1$, $Driver_2$, $Link_1$, $Event_1$, and so forth;
- a set Variables of variable symbols: $agent$, $driver$, $link$, $event$, and so forth;
- a set Predicates of predicate symbols: $Adjacent$, $Travel$, and so forth.

The possible relations among the various components of the language are listed next. It is important to bear in mind that both state and path formulas are evaluated in a given accessible-world. To build a state formula, the following rules must be observed.

- any first-order formula is a state formula;
- if ϕ_1 and ϕ_2 are state formulas and x is an individual or event variable, then $\neg\phi_1$, $\phi_1 \wedge \phi_2$, and $\exists\, x\ \phi_1(x)$ are state formulas;
- if e is an event type then $\mathsf{succeeds}(e)$, $\mathsf{fails}(e)$, $\mathsf{does}(e)$, $\mathsf{succeeded}(e)$, $\mathsf{failed}(e)$, and $\mathsf{done}(e)$ are state formulas;
- if ϕ is a state formula then $\mathsf{BEL}(\phi)$, $\mathsf{GOAL}(\phi)$, and $\mathsf{INTEND}(\phi)$ are state formulas;
- if ψ is a path formula, then $\mathsf{E}(\psi)$ and $\mathsf{A}(\psi)$ are state formulas.

Similarly, in order to build a path formula, the following rules must be observed:

- any state formula is also a path formula;
- if ϕ_1 and ϕ_2 are path formulas, expressions similar to $\neg\phi_1$, $(\phi_1 \vee \phi_2)$, $(\phi_1 \cup \phi_2)$, $\Diamond\phi$, and $\Box\phi$ are also path formulas.

4.4 Traffic system: the application domain

As mentioned before, the application domain of concern in this work is the traffic system of urban areas. More specifically, it is focused on the commuter scenario where decisions such as what time to depart and which route to take are important to meet certain constraints as of fixed arrival time at destinations. So, modelling driver reasoning is central, albeit TIS technologies are also to be represented in terms of agents.

Drivers are dealt with as rational and intentional entities. Hence, public transport users, as well as other transport modes are not taken into consideration at the current level of this research. It is the typical commuter scenario where travellers are already to possess some knowledge about the traffic network, its dynamics, and its topology. The uncertainty inherent in humanlike decision-making is the factor to grant variability in demand formation.

Each driver has the goal of reaching certain destination, its work place for instance, departing from an origin within the traffic network, such as its home. Two basic decisions have to be made in order to accomplish such a goal, namely what time to depart and which route to take. Bearing this situational configuration in mind, one can consider as basic arguments for this goal the origin, the destination, the route, and the departure time chosen. Nonetheless, it is getting to be a common practice using some facilities that provide updated information about the actual state of the network, both prior and during the journey. This represents an opportunity toward overcoming the very inaccessible nature of the traffic environment, even for commuters. For example, radio broadcast, variable message signs (VMS), dynamic route guidance (DRG), and the Internet are used in this way. Other technologies, such as mobile communication, have improved and made the access to information easier.

Since individuals can access reliable and updated information about the system state, they now are able to make efficient decisions. This is conceivable as exogenous sources can improve the cognitive representation of the world within drivers' reasoning. For example, one could avoid unwanted situations such as traffic jams, or could choose the ideal path in terms of different interpretations of cost to its destination. TIS will definitely influence the way drivers make decisions and behave as traffic network users. Therefore, some level of interaction will also be necessary to some extent, even as just to allow drivers to receive information. With regard to this interactive nature, two kinds of such systems can be identified, namely the *passive* and the *interactive* ones. The former would include those that only send information that is current. On the other hand, the latter would be able to tailor contents to meet users' needs and would present some degree of adaptability, as well. Moreover, driver and information system could interact in a co-operative way in order to reach a certain destination efficiently. For example, an inquire-response mode of co-operative interaction could entail a more precise information to fill the driver's needs.

4.4.1 Description of traffic entities

An individual usually organises his knowledge of the traffic system in terms of network topology and dynamics. Topology has a quite simple representation, whereas dynamics is mostly associated with the recurrent traffic flow. Yet, such an association is mostly related to certain periods, for example, of the day, of the week, and even of the moth and of the year. Beliefs such as "certain road is always congested" or "that one would have a free traffic because it is wider" reflects the cognitive notion a driver may have about the capacity of a link. These beliefs are usually built on the basis of either the physical description of the road or after experiencing the necessary time to travel through a certain part of the road. In this way and considering the cognitive picture of the environment the driver conceives, one can identify the elements that compose the traffic system.

The traffic network is usually organised in terms of roads (links) connected to each other. In this way links form the network topology. Although commuters are very familiar with the system, their knowledge will be limited to few alternative routes that are identified as sequences of adjacent links. Each link is weighted with a cost, which is updated as drivers realise a trip through it or receive any sort of information on it. The interpretation for cost, in this case, will depend widely on the purpose of the journey. For example, as a work commuter travels it perceives the time it takes to perform each link within the route. However, the knowledge got from this very sensing act is commonly translated to qualitative terms as the traffic was free or congested, for instance. The average travel time for the entire journey is very likely memorised, though. An individual could deduce that a certain road would be congested from knowing the actual state of adjacent ones, as well. Further, such an inference could result from simply considering the physical structure of a link, for example, “that road is likely to have a free flow because it has five traffic lanes”, and “the other one is too narrow and will be probably blocked during lunch time”. So, contrary to the way adopted in traditional models, drivers are likely to make qualitative rather than quantitative assessments of the system.

In general terms, a traffic system can be seen as formed of moving particles, which are the vehicles and, implicitly, the driver, and of the network that is the environment (ROSSETTI, 1998). Basically, one can consider a population of potential commuters that are able to perform a journey throughout the network. Individuals that have decided to make a trip on a certain day will constitute the demand for travel on that day. The network, on its turn, is built up of links representing roads.

Thus, *Drivers* is a constant that represents a set of drivers. It can be referred to as the population of potential commuters in the traffic system. $Driver()$ is a unary predicate that determines its argument is a driver.

$$Drivers = \{d_1, d_2, d_3, \ldots, d_n\} \quad n \in \mathbb{N}$$

$$\forall d \quad Driver(d) \Rightarrow d \in Drivers$$

Similarly, $Network$ is a constant that represents a set of links connected to each other and $Link()$ is a unary predicate that determines its argument is a link.

$$Network = \{l_1, l_2, l_3, \ldots, l_n\} \quad n \in \mathbb{N}$$

$$\forall l \quad Link(l) \Rightarrow l \in Network$$

It is equally important to identify the current state of the driver. For example, it may be stationary at an origin or a destination, or it may be moving through a link. The predicates $WaitingAt(d,l)$ and $MovingOn(d,l)$ could be used to denote the state of a driver d with respect to a link l. It is important to bear in mind that origins and destinations are dealt with as links, as well. This simplification is adopted as the connection of source and drain zones to the traffic network is usually represented by means of *dummy* links.

A driver has the notion of adjacency, both for longer and shorter parts of a road. For example, an individual may know every transversal to certain road, such as the Ipiranga Avenue, in Porto Alegre, whereas another knows only the main junctions to that road ignoring all the others between them. For the sake of simplicity, a link is atomic, which means no transversal is considered to exist. The other assumption made about links is that they are directional. Hence, there is a downstream and an upstream node associated to each of them. In other words, a link can be represented in terms of a directional segment between two consecutive intersections (see Figure 4.3). The following formula expresses that l_1 and l_2 are adjacent with regard to one another.

$$\forall l_1, l_2 \quad Adjacent(l_1, l_2) \Rightarrow Connected(Upstream(l_1), Downstream(l_2))$$

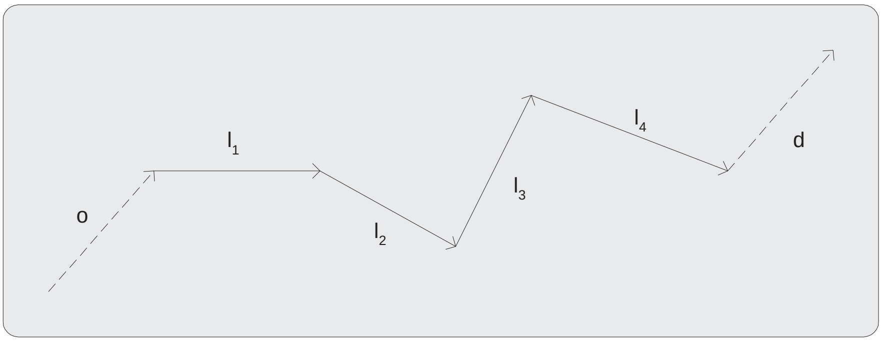

Figure 4.3: Links of a traffic network

The same idea is used to represent routes, which are lists of links. A necessary condition, however, is that the links should be adjacent to each other and consecutive. Similarly, *Routes* is a set of *route* symbols, which represent the alternatives a driver is able to choose from. The unary predicate $Route(route)$ relates its argument to a route object [1].

$$Route = \{origin, l_1, l_2, \ldots, l_i, l_{i+1}, \ldots, l_n, destination\} \quad i, n \in \mathbb{N}$$

$$\forall r \quad Route(r) \Rightarrow r \in Routes$$

$$\forall l_i, l_{i+1} \quad l_i, l_{i+1} \in Route \Rightarrow Connected(Upstream(l_i), Downstream(l_{i+1}))$$

Every driver who is familiar with the network to some extent has a cost assigned to each link within its internal model. However, contrary to be a quantitative notion, such a cost gives a qualitative idea of the link. Thus, *LinkStates* is a set of constants representing possible states of a link. In the commuter world, three different levels of cost could be considered, to mention some. The $LinkState(link, state)$ predicate relates a link to its actual state.

$$LinkStates = \{Congested, Normal, Free\}$$

$$LinkState(l_1, Congested) \quad LinkState(l_2, Normal) \quad LinkState(l_3, Free)$$

[1] An important assumption made in this work is that routes are previously determined and are part of a library for each agent. Routes can be built as a result of the Desopo's shortest path algorithm (VAN VLIET, 1977).

Temporal operators, such as *always* and *eventually*, could be associated to these formulas, as exemplified bellow.

$$\Box LinkState(l_1, Congested)$$

$$\Diamond LinkState(l_2, Free)$$

Contrary to the qualitative assessment made over each link, the average travel time may be assigned to the entire route. Thus, a possible predicate to represent this could be $TravelTime(route, time)$, which associates an average travel time to a certain route. A predicate such as $Minute(value)$ could be used to denote that time is given in minutes. Nonetheless, if the time unit is to be generalised, the formula $TravelTime(r, Minute(45))$ could be simply written as $TravelTime(r, 45)$.

Another important time dependent representation that should be present in a commuter model, is the notion of time associated to the instant an action is performed. What time a commuter needs to depart and what time it is supposed to arrive at work, are good examples. A possible inference as to such notions could be “if I departed at that time and got that route would I arrive at work in time?”. Instants are usually identified from within the day. Thus, the predicates $DepartAt(time)$ and $ArriveAt(time)$ could be used to represent the instant such actions are to be performed. In the same way, $DepartureTime(time)$ and $ArrivalTime(time)$ could be used to represent the instants those actions actually happened. Besides, operator symbols could also be used to further detail actions representation.

4.4.2 An example of a logic traffic system

In order to illustrate the logics presented in the previous sections, a simple example was devised. It gives a little flavour of the logics adequacy to model humanlike behaviour and decision-making in the traffic and transportation domains. The system instance is composed of a hypothetical network with three possible routes for an origin-destination pair, as depicted in Figure 4.4.

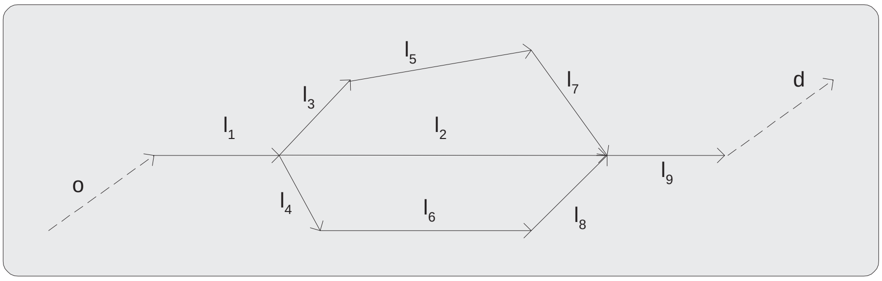

Figure 4.4: A hypothetical traffic network

The network can be represented by means of a set of its links, denoted as N_1. So, $N_1 = \{o, l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, l_9, d\}$ where the following conditions should be observed.

$Adjacent(o,l_1)$,
$Adjacent(l_1,l_2)$, $Adjacent(l_2,l_9)$,
$Adjacent(l_1,l_3)$, $Adjacent(l_3,l_5)$, $Adjacent(l_5,l_7)$, $Adjacent(l_7,l_9)$,
$Adjacent(l_1,l_4)$, $Adjacent(l_4,l_6)$, $Adjacent(l_6,l_8)$, $Adjacent(l_8,l_9)$,
$Adjacent(l_9,d)$.

R_1 is the set formed of the three possible routes between origin o and destination d, and is given as $R_1 = \{r_1, r_2, r_3\}$. These are the paths the driver actually knows for the journey. Nonetheless, such a set might be expanded as an individual experimented other roads, searched in a map, or was otherwise advised by a Route Guidance System, for instance. Each route from R_1 is described next.

$r_1 = \{o, l_1, l_2, l_9, d\}$,
$r_2 = \{o, l_1, l_3, l_5, l_7, l_9, d\}$,
$r_3 = \{o, l_1, l_4, l_6, l_8, l_9, d\}$.

Considering a population D_1 of commuters, a certain driver $d_i \in D_1$ could have the following mental attitudes in a given moment in time.

$\mathsf{BEL}(d_i, WaitingAt(d_i, o))$,
$\mathsf{BEL}(d_i, TravelTime(r_2, Minute(45)))$,
$\mathsf{BEL}(d_i, \Diamond LinkState(l_5, Free))$,
$\mathsf{BEL}(d_i, \Box LinkState(l_2, Congested))$,
$\mathsf{GOAL}(d_i, WaitingAt(d_i, d))$,
$\mathsf{GOAL}(d_i, ArriveAt(t+x))$.

The events are path formulas representing the course of actions that enable an agent to reach a desired future state. In other words, they can be seen as the strategy of an individual to accomplish its goals. Such a commitment to realise certain course of actions to the detriment of other possible ones is the abstraction for the agent's intention. In this specific example, the driver commits to execute the following actions.

$\mathsf{INTEND}(d_i, DepartAt(t))$,
$\mathsf{INTEND}(d_i, TakeRoute(r_3))$.

Figure 4.5 depicts this simple example in terms of beliefs-, goals-, and intentions-accessible worlds.

4.4.3 Planning a trip

Basically, a initial planning task is executed to set the components of a trip. Thus, a trip could be defined as the tuple $Trip\langle O, D, P, DAT, R, DT\rangle$. For the sake of simplicity, the terms within the tuple were abbreviated. O represents a link o which can be identified as the origin for the trip, $Origin(o)$, whereas D is the destination link d, given by

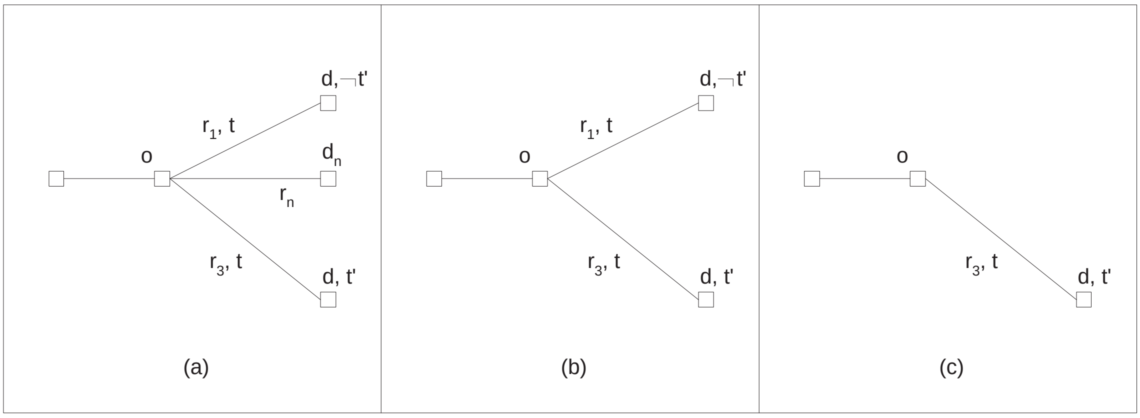

Figure 4.5: The possible worlds for the traffic example: (a) beliefs-accessible world; (b) goals-accessible world; and (c) intentions-accessible world

$Destination(d)$. The purpose for the journey, given by P, can be any from a set of possible motivations, such as *work*, *leisure*, *shopping*, and so forth. The reason for making a journey commonly implies an arrival deadline, which the driver attempts to meet. This is the *DAT* (desired arrival time) term of the trip. In order to effectively accomplish its trip end, the driver should choose a route *R*, from the set of routes it knows for the specific *OD* pair, and a departure time DT. Thus, the desired arrival time can be seen as a goal of a commuter. Route and departure time express the commitment to attempt to achieve that goal, which results from the driver's decision-making. Hence, this idea could be represented in the following way.

$\mathsf{GOAL}(ArrivalAt(time))$,
$\mathsf{INTEND}(DepartAt(time))$.
$\mathsf{INTEND}(TakeRoute(route))$,

The plans of a driver are pre-determined and possible routes between origins and destinations are stored in a plan library. This limit the number of options available to the drivers, as in real life people are usually presented up to three or four route choices, at most.

4.4.4 Strategies for decision-making

Representing drivers' behaviour and decision-making is a topic of main interest in works aimed at assessing variable demand. Some models, such as DRACULA (LIU; VAN VLIET; WATLING, 1995), deal with decision-making on the basis of past experiences. This way, a driver is endowed with memory and is able to store travel experiences in terms of cost, usually travel time for commuters. Thus, each link of its internal model is weighted by any form, which is a quantitative approach, though. In practice, the cognitive reasoning of human beings is mostly associated to qualitative aspects of the environment rather than exact numeric measures.

It is intuitive that a driver does not explicitly remember the travel time for each link, and sometimes neither for the entire route. Nevertheless, it is possible to identify an implicit cognition performed by individuals in order to acquire the qualitative attributes of any element within the system. This could be produced by comparisons. For example, as to the link states a driver could make the following associations.

- when travelling through a link l with an average speed s at least equal to the desired speed s_d, the link could be associated with a free flow state;

$$\forall s, s_d, d, l \quad Speed(s) \wedge Speed(s_d) \wedge Driver(d) \wedge Link(l)$$

$$\wedge Higher(s, DiseredSpeed(d, s_d)) \Rightarrow LinkState(l, Free)$$

- if the average speed s performed through a link l is lower than the desired speed s_d, but higher than certain value, say $20km/h$, defining a stationary flow, then l could be associated to a normal state;

$$\forall s, s_d, d, l \quad Speed(s) \wedge Speed(s_d) \wedge Driver(d) \wedge Link(l)$$

$$\wedge Lower(s, DiseredSpeed(d, s_d)) \wedge Higher(s, Speed(20)) \Rightarrow LinkState(l, Normal)$$

- finally, a congestion can be associated to a link l when the average speed s is equal or lower than the value that characterise a stationary flow.

$$\forall s, s_d, d, l \quad Speed(s) \wedge Speed(s_d) \wedge Driver(d) \wedge Link(l)$$

$$\wedge Lower(s, Speed(20)) \Rightarrow LinkState(l, Congested)$$

Contrary to the way routes are usually dealt with, the average travel time for the entire journey is not evaluated on a link-by-link basis. Rather, it is identified by the difference between the arrival time at destination and the departure time from origin. It could be seen as a more intuitive way to represent the reality, albeit it is just a different manner to write the sum of all link costs throughout the route.

$$\forall r, t, t_o, t_d \quad Route(r) \wedge Time(t) \wedge Time(t_o) \wedge Time(t_d)$$

$$\wedge TravelTime(r, t) \Rightarrow t = Difference(ArrivalTime(t_d), DepartureTime(t_o))$$

This way, many decisions could be made in terms of assessing qualitative aspects of the options available to each driver. As to route choices, for instance, the driver would evaluate quality of the flow through each link of the route, which the driver believes to hold at the instant the decision is made. If an individual believes a link within certain path is always congested, it very likely would take an alternative way. Otherwise, the driver could keep the usual route and opt for departing a little bit earlier.

$$\forall r_1, r_2, l_1, l_2, d \quad Route(r_1) \wedge Route(r_2) \wedge Link(l_1) \wedge Link(l_2) \wedge\ l_1 \in r_1 \wedge\ l_2 \in r_2 \wedge \neg(l_1 = l_2)$$

$$\wedge \mathsf{BEL}(d, \Box LinkState(l_1, Congested)) \Rightarrow TakeRoute(d, r_2)$$

There are many other ways to represent human cognition in commuter scenarios, as are the different manners humans make decisions. For example, the state of a link could be associated to certain period of the day, or even could a route be thought of as a relaxing one for the sake of landscape. This section served to illustrate how complex and arduous it would be to describe the ontological level of a system. And this is specially the case of traffic and transportation domains when the task is to model the system from the perspective of drivers. Also, some epistemological nature of systems can demand more appropriate representation, as the one offered by fuzzy logics, for instance.

4.5 Practical ways to implement BDI models

A logical formalism allows for the efficient representation of all the knowledge an agent must possess about the world and how to reason on it. However, it is easy to realise from the brief discussion in the last section that describing complex domains both ontologically and epistemologically may become an arduous and extensive task. Yet, it is also central to MAS providing an operational model that supports the implementation of the agent architecture. BDI formalisms have demonstrated a natural ability to design humanlike cognitive behaviour. It is also commonsense that in fact the so quoted ‘gap’ between modelling and practical implementation has discouraged using it despite its expressiveness power. Efforts have then been devoted to finding a way to bridge such a gap and turn BDI models into real applications.

Móra et al. (1999) identify at least two basic approaches to overcome limitations of BDI formalisms. It is possible to extend existing logics with appropriate operational models, or one can use other logical formalism that is powerful enough both to provide a cognitive representation of the domain and to offer operational procedures for practically building agents. In his thesis, Móra (1999) tackle the problem of devising computational BDI models by following a similar way as adopted in (CORRÊA; COELHO, 1993). Instead of defining a new BDI logic or choosing an existing one in order to extend it, the notions of beliefs, desires, and intentions are defined by means of a formalism that is both well-defined and computational. This is achieved by means of using *logic programming extended with explicit negation* (ELP) and *Well-Founded Semantics extended for explicit negation* (WFSX). In such a framework, an agent is defined as the tuple $\langle B, D, I, TAx \rangle$, where B is a set of beliefs, D is a set of desires, I is a set of intentions, and TAx is a set of time axioms. Plans are built out from an explicit declaration of actions and the period of time they should be carried out. This effort has also departed from Bratman’s (1987) philosophical work as have other attempts to formalise and implement BDI multi-agent systems (HADDADI, 1996; RAO; GEORGEFF, 1991, 1995).

Rao (1996) worked on an alternative formalisation of BDI agents to devise an operational and proof-theoretic language, AgentSpeak(L). The language can be seen as an abstraction of implemented BDI systems, such as PRS and dMARS, which determines the behaviour of the agents it implements. AgentSpeak(L) allows programs to be written and interpreted in a way similar to that of Horn-clause logics (SCHACHTE, 2002). D'Inverno and Luck (1998) further discussed its primitives and semantics on the basis of a Z specification. Curiously, despite these authors claimed and formally demonstrated the ability of AgentSpeak(L) to specify and allow for practical implementation of BDI agents, an interpreter for the language waited for relatively long time to be implemented.

After presenting a comparison between 3APL and other agent languages (HINDRIKS et al., 1997), Hindriks et al. (1998b,c) formally suggest that it is possible to simulate the operational semantics of AgentSpeak(L) within 3APL. In other words, every agent that can be programmed in Rao's (1996) language can also be programmed in 3APL. In a similar approach to overcome the lack of an interpreter for AgentSpeak(L), Machado and Bordini (2002) reported their experiences in running AgentSpeak(L) programs within SIM_AGENT framework (SLOMAN; POLI, 1995). SIM_Speak, as the authors coined their environment, is capable of converting AgentSpeak(L) specifications to SIM_AGENT agent programs. The latter is based on previous extensions to the POP-11 language (SLOMAN, 1999; BARRETT; RAMSAY; SLOMAN, 1985), allowing SIM_AGENT to be a general tool that leaves to programmers the task of determining the architecture of the agents (MACHADO; BORDINI, 2001).

Although approaching quite successfully the 'bridging the gap' between theory and practice, Bordini and Machado (2002) also suggest that the main advantage of a purpose-built interpreter as opposed to running AgentSpeak(L) agents within SIM_AGENT, would be in terms of efficiency and practicality. This was to become reality recently, as Bordini et al. (2002) presented an interpreter to their AgentSpeak(XL), an extension proposed to improve AgentSpeak(L) in various aspects and particularly for supporting the use of Design-To-Criteria (DTC) scheduler (BORDINI et al., 2001) to allow the generation of efficient intention selection functions. In both works (MACHADO; BORDINI, 2001; BORDINI et al., 2002), authors left a remarkable contribution to the BDI community as they presented further understandings on the operational semantics of AgentSpeak(L) and enhanced its interpreter as to AgentSpeak(XL).

It is also relatively recent that commercial tools for the development of multi-agent systems have been claimed to support the practical implementation of BDI-based models. JACKS Intelligent Agents (AGENTLINK, 2002; AOS, 2002) is a multi-agent system development environment commercialised by AOS. It is based on the JACK Agent Language (JAL) that extends the Java language to allow embedding BDI-based reasoning within Java objects. Bee-gent (Bonding and Encapsulation Enhancement Agent) is a framework for the development of agent-based distributed systems (KAWAMURA et al., 1999; TOSHIBA CORPORATION, 1999), which has been developed at TOSHIBA Corporation's Research and Development Center. A number of API supports the implementation of two types of agents, namely the agent wrapper and the mediation agent, which can be featured with BDI reasoning capabilities.

UMPRS (LEE et al., 1994) and JAM (HUBER, 1999a,b) are both BDI architectures that have been developed at Intelligent Reasoning Systems and University of Michigan (INTELLIGENT REASONING SYSTEMS, 2002), and explicitly include the original BDI theories and specification of PRS (HUBER, 1999a; GEORGEFF; LANSKY, 1987; INGRAND; GEORGEFF; RAO, 1992). So, their constructors and operational semantics resemble enormously the ones of AgentSpeak(L) language (HUBER, 2001, 1994). While agents specified in UMPRS are parsed to C++ code, the ones specified in JAM are parsed to Java code. In the special case of JAM, much effort have been devoted to provide users with technical aid and documentation, and both frameworks are made freely available for non-profitable ends, such as academic works.

4.6 AgentSpeak(L): Specifying and Programming BDI Agents

AgentSpeak(L) is a language devised to bridge the gap between formal modelling and practical implementation as far as BDI agents are concerned. It basically reduces the task of modelling intentional agents to identifying base beliefs, goals and plans. Given its expressiveness and ease of use, AgentSpeak(L) is applied in this work as an specification language. Such a decision is based on the assertions by Machado and Bordini (2002) as to the fact that Rao (1996) also devised a proof theory for AgentSpeak(L). Unfortunately, neither the SIM_Speak framework nor the AgentSpeak(XL) interpreter were effectively available for use when this thesis' proposal was presented. In order to experiment and demonstrate the approach of this research, a choice was made toward implementing AgentSpeak(L) drivers in JAM. Nonetheless, it is the step ahead to use AgentSpeak(XL) in further developments.

Before going further on the specification of the commuter scenarios presented in this work, it is believed to be worthwhile presenting the syntax of AgentSpeak(L), so as to facilitate the understanding of the specification of the BDI-like driver behaviour model. The following definitions giving the syntax of the language are taken from (RAO, 1996), where AgentSpeak(L) was first specified, and are presented here exactly as they were given in his original work.

Definition 1 *If b is a predicate symbol, and $t_1,\ldots,t_n$ are terms then $b(t_1,\ldots,t_n)$ or $b(\mathbf{t})$ is a belief atom. If $b(\mathbf{t})$ and $c(\mathbf{s})$ are belief atoms, $b(\mathbf{t}) \wedge c(\mathbf{s})$, and $\neg b(\mathbf{t})$ are* beliefs. *A belief atom or its negation will be referred to as a* belief literal. *A ground belief atom will be called a base belief.*

Definition 2 *If g is a predicate symbol, and $t_1,\ldots,t_n$ are terms then $!g(t_1,\ldots,t_n)$ (or $!g(\mathbf{t})$) and $?g(t1,\ldots,tn)$ (or $?g(\mathbf{t})$) are* goals.

Note that both beliefs and goals are predicate symbols (as are actions, as seen later). A predicate symbol is a goal if it is preceded by the operators '!' or '?' (see Definition 2). Thus, an agent *a* may have the *achievement goal* of being in location Y in a future state, expressed by $!location(a,Y)$. It may also check what its present position is, by the *test goal* $?location(a,X)$, given its set of base beliefs (updated through perception of the environment).

Definition 3 *If* $b(\mathbf{t})$ *is a belief atom,* $!g(\mathbf{t})$ *and* $?g(\mathbf{t})$ *are goals, then* $+b(\mathbf{t})$, $-b(\mathbf{t})$, $+!g(\mathbf{t})$, $+?g(\mathbf{t})$, $-!g(\mathbf{t})$, $-?g(\mathbf{t})$ *are* triggering events.

Agents go through repeated cycles where they observe the environment and, based on their observations and goals, they execute certain actions that may change the state of the environment. This may influence the agents' beliefs as well, which need to be revised.

Definition 4 *If a is an action symbol and* $t_1,\ldots,t_n$ *are first-order terms, then* $a(t1,\ldots,tn)$ *or* $a(\mathbf{t})$ *is an* action.

Definition 5 *If e is a triggering event,* $b_1,\ldots,b_m$ *are belief literals, and* $h_1,\ldots,h_n$ *are goals or actions then* $e : b1 \wedge \ldots \wedge b_m \leftarrow h_1;\ldots;h_n$ *is a* plan. *The expression to the left of the arrow is referred to as the* head *of the plan and the expression to the right of the arrow is referred to as the* body *of the plan. The expression to the right of the colon in the head of a plan is referred to as the* context. *For convenience, an empty body is rewritten with the expression* $true$.

Rao (1996) further mentions that a plan specifies the means by which an agent should satisfy an end. However, in none of the known work concerning AgentSpeak(L) (RAO, 1996; D'INVERNO; LUCK, 1998) the authors approach the issue of how beliefs and intentions are updated during the execution of a plan [2].

Definition 6 *An* agent *is given by a tuple* $\langle E,B,P,I,A,\mathscr{S}_{\mathscr{E}},\mathscr{S}_{\mathscr{O}},\mathscr{S}_{\mathscr{I}}\rangle$*, where E is a set of events, B is a set of base beliefs, P is a set of plans, I is a set of intentions, and A is a set of actions. The selection function* $\mathscr{S}_{\mathscr{E}}$ *selects an event from the set E; the selection function* $\mathscr{S}_{\mathscr{O}}$ *selects an option or an applicable plan* [3] *from a set of applicable plans; and* $\mathscr{S}_{\mathscr{I}}$ *selects an intention from the set I.*

Definition 7 *The set I is a set of intentions. Each* intention *is a stack of* partially instantiated plans, *i.e., plans where some of the variables have been instantiated. An intention*

[2] It is referred here to the updating of beliefs and deletion of intentions directly from the execution of a plan, not (in the case of beliefs) through changing the environment by means of actions and subsequent perception of the environment and the ensuing belief update, or (in the case of intentions) executing subplans. Although, in some of the examples given in the papers it appears that this is possible, formally a plan is only formed of goals and actions, not triggering events (i.e., addition and deletion of beliefs).

[3] Rao (1996) presents applicable plans in Definition 10 of his original work.

is denoted by $[p_1 \ddagger \ldots \ddagger p_z]$*, where* p_1 *is the bottom and* p_z *is the top of the stack. The elements of the stack are delimited by* $\ddagger$*.* For convenience, Rao (1996) refers to the intention $[+!true : true\texttt{<-}true]$ as the true intention, and denotes it by T.

Definition 8 *The set E consists of events. Each event is a tuple* $\langle e, i \rangle$*, where e is a triggering event and i is an intention. If the intention i is the* true intention, *the event is called an* external event*; otherwise it is an* internal event *(and i is the intention that has generated the event e).*

As described in (D'INVERNO; LUCK, 1998), there are two basic models of operation, which essentially involve either updating the intention set, reflected by responding to an event, or actually executing intentions. When updating the intention set, the agent selects an event *e*, from the set of events E, and generates all the plans whose invocation conditions, identified by the triggering event at the head of the plan, match this event. These plans are the *relevant plans*. Then, if the context part of a relevant plan is a logical consequence of the set of base beliefs, it will be selected as an *applicable plan* and will form a plan instance that is the agent's *intended means*. Ending the cycle, the agent updates its set of intentions I. If the event selected, which started the cycle, is an external event, a new intention is generated and inserted into the intention set. Otherwise, the event is internal and the plan instance is added to the head of the intention that posted the event.

In the second model, the agent selects an intention from the intention set I. The plan at the top of the selected intention is now the *executing plan*, and the next formula in the body of the plan is the *executing formula*. Depending on the selected intention and the executing formula of the executing plan, the agent starts one of the possible courses of action. If the executing formula is an achieving goal, a new goal event is generated and posted to the event set E, and the intention is suspended until the goal has been achieved. In case the executing formula is a query goal, the information retrieved is used to instantiate the corresponding terms in the executing plan. Finally, if the executing formula is an action, the action is posted to the action set *A*, awaiting execution. In the last two situations, the executing formula is removed from the executing plan. If there is no further formula in the executing plan, the agent starts the execution of the next plan in the selected intention. If there is no next plan, then the intention has succeeded and can be removed from the intention set.

It is important to note that an AgentSpeak(L) agent is specified simply as a set of base beliefs and a set of non-instantiated plans, which turn the task of modelling quite easier. Intentions are generated automatically from triggering events. This process is detailed in (RAO, 1996; D'INVERNO; LUCK, 1998). Also, Machado and Bordini (2002) give a remarkable insight into AgentSpeak(L) operational semantics.

In this thesis, the focus is given only to those elements that are necessary to characterise the BDI agents, as presented above. As suggested in (MACHADO; BORDINI, 2001), and already experimented in (ROSSETTI; BORDINI; BAZZAN; BAMPI; LIU;

VAN VLIET, 2002; ROSSETTI; LIU; CYBIS; BAMPI, 2002), AgentSpeak(L) will be used to specify BDI-like driver behaviours.

4.7 Summary

One main premise of ITS is to optimise traffic flow through enhancing driver behaviour patterns. This gives rise to the use of information technologies as the instrument to accomplish such an aim. Therefore, describing humanlike factors is the basis that allows for modelling, simulating, and assessing the impact and efficiency of TIS. As travellers enrich their knowledge about the very inaccessible traffic and transportation environment, decision-making is now to yield optimised choices. So, models should cope not only with reactive aspects, which are well-handled within the car-following and lane-changing representations, but also with the cognition level. However, very little work has succeeded in either addressing human behaviour or proposing a means to overcome their very complex nature (WATLING, 1994).

The BDI approach was presented in this chapter, which relies on Bratman's (1987) work. Contrary to many reductionist authors, intelligent agents are dealt with as intentional rather than only rational entities. In this way, intentions become an important component for reasoning, as are beliefs and desires. An agent then is motivated by its goals and commits to certain courses of actions in order to accomplish them. Despite its ability to represent cognition, BDI-based models have not been widely used. The lack of effective implementation tools for a while since it was first proposed can perhaps explain this. Nonetheless, some alternatives extending the BDI logics have been successfully used within domain-specific applications, such as the one presented in (TEDESCO; SELF, 2000). Yet in this way, only societies with a reduced number of elements have profited from the potentialities of such an approach.

It is relatively recent that people have again demonstrated some interest in using BDI models within a variety of applications. Two main reasons may justify this trend. Firstly, advances in computer architectures have enhanced both processing and memory capacities. More or less dependent on the former, developing environments now support the effective implementation of BDI-based agents. Some of which are commercially available. However, using it within domains formed of several reasoning entities has not, as yet, been actually experienced. This is specially the purpose of this work.

In order to model drivers as reasoning agents, the decision was made toward using Rao's (1996) AgentSpeak(L) language rather than devising a domain-specific extension of BDI logics from scratch. Its syntax and semantics rely entirely on Rao and Georgeff's (1991) formalism . Additionally, the language possesses a proof theory, which suits perfectly specification purposes (MACHADO; BORDINI, 2001). Curiously, no interpreter was made available until very recently, as reported in (BORDINI et al., 2002). To turn around this very absence, The JAM architecture (HUBER, 1999a,b) was used for practical implementation of the BDI-based drivers. Some of its constructors keep the same semantics and relations as in AgentSpeak(L). Following the approach suggested in (HINDRIKS et al., 1998a, 1997) it is possible to simulate Rao's (1996) language functioning in JAM. This way, drivers will be endowed with a BDI kernel to support their reasoning ability. This is expected to improve behaviour modelling and ease the assessment of decision-making on a variety of scenarios.

5 A BDI MODEL OF COMMUTER SCENARIOS

5.1 Overview

One goal of this work is to present a methodological approach to aid the modelling and implementation of driver behaviours in commuter scenarios. Machado and Bordini (MACHADO; BORDINI, 2001) claim that the AgentSpeak(L) language is suitable as a specification tool, despite the absence of a purpose-built interpreter for the language and therefore could be used to suit this end. This assertion is based on the fact that Rao (RAO, 1996) devised the language on the basis of both an operational and proof-theoretic semantics. Some experiences of using AgentSpeak(L) for specification purpose are reported in (ROSSETTI; BORDINI; BAZZAN; BAMPI; LIU; VAN VLIET, 2002; ROSSETTI; LIU; CYBIS; BAMPI, 2002).

This chapter, initially addresses how AgentSpeak(L) is used to specify BDI commuting drivers. Commuters are expected to be familiar with the traffic system, hence routes are chosen from a limited set of options. Also, for most commuters, there may be a sort of rigid arrival time that should be met to the extent of the purpose of the journey. Thus, the basic decisions a driver agent has to make are as to which route to take and what time to depart so that it can achieve its trip objectives.

Different premises may be assumed when people are reasoning and making decisions. In this sense, it is also likely that different individuals adopt different strategies to complete a certain goal. Such strategies will be grouped into driver personalities, which dictates the way an individual reasons about his/her base beliefs. Three initial personalities were devised on the basis of intuitive considerations, namely the random, the choosy, and the conservative ones. A fourth personality coined the habitual driver was devised on the basis of the driver behaviour currently implemented in DRACULA, as described in (LIU; VAN VLIET; WATLING, 1995). Further extensions to the habitual personalities are also proposed.

5.2 Traffic domain from a multi–agent system perspective

The task of assessing ITS technologies brought about the need for a more robust means to model the real world (CHATTERJEE; MCDONALD, 1999). MAS seems to be able to give an enormous contribution to this end owing its powerful expressiveness for ontological and epistemological representation of complex systems. Bearing in mind the structure for traffic systems in urban areas, it is possible to identify their basic components, namely the environment and the moving particles. The environment can be viewed as the road network itself and the control systems (such as traffic signals and traffic signs), which dictates the movement rules. Vehicles are the moving entities travelling throughout the road network (ROSSETTI, 1998; ROSSETTI; BAMPI, 1998b, 1999). Nonetheless, today's traffic and transportation are increasingly influenced by the presence of ITS technologies, which should be taken into account and integrated into the environment, as well. In a demand-supply perspective, the environment with all its resources can be seen as the supply for a demand formed of moving individuals that seek to accomplish their trip goals.

A traffic system is notably formed by heterogeneous entities, which are geographically and functionally distributed over the environment. Their very nature suggests that such components can be easily recognised as agents in a multi-agent system. Owing the complexity and dimensions of the domain, urban areas are usually divided into zones. In turn, zones within a city could be seen as open agent societies, through which individuals are able to move from one to the other.

Accounting for the ITS premise of being able to influence users' behaviours, drivers start playing a crucial role in the system and modelling it as an agent deserves special attention. Nonetheless, the other elements within the system are equally important. Some of the environment components have already been subjected to either reactive or cognitive modelling approaches, as seen in Section 3.8. However, either views cannot be simply applied to driver agents. Drivers could be understood as behaving in both reactive and cognitive ways. When answering to control systems or responding to some stimuli brought about by the presence of surrounding vehicles, drivers' behaviour is basically reactive. The car-following and lane-changing models are traditional representation of such a reactive behaviour. In these models, drivers are specified by means of rules that map actions to specific events, such as red light of a traffic signal and the break light of the vehicle ahead. However, when planning a trip, choosing route and departure time, or even even when deciding whether to divert in the presence of a traffic jam, drivers must exert their reasoning capabilities that strongly rely on mental states such as beliefs, desires, and intentions.

All of the agents in a traffic domain will interact with each other and with the environment in order to improve the system performance, albeit each of which has specific task and goals. Interaction between agents and between an agent and the environment is a factor of huge importance to the exchanging of information in a traffic system. Besides the built-in knowledge of drivers, they can acquire information by accessing an ATIS and by observing the environment in previous experiences. Another important issue is the time-dependent nature of interactions that makes it possible to see traffic systems as a real-time domain. This characteristic becomes more evident and significant with the adoptions of ITS technologies, as the communication is a key feature in such a concept. Both soft and hard real-time features can be identified. As consequence of failure in a Traveller Information System, drivers would experience increases in travel cost by virtue of an unexpected traffic jam, for example. This scenario could define a soft real-time system. However, some ITS technologies are strongly dependent on the reliability of the system. In automated cruise control systems, embedded systems are in control of the vehicle navigation. A failure in these systems could result in drastic consequences, such as crashing. In this case, ITS technologies can be seen as hard real-time systems.

5.2.1 The driver agent architecture

Drivers are autonomous in the sense they can make decisions on their own in order to accomplish their objectives. This way demand is built up as the result of the decision-making process carried out on a decentralised basis (ROSSETTI; LIU; CYBIS; BAMPI, 2002). So, it is the driver's own responsibility to identify its needs, to manage its resources, and to make its decisions. Drivers are also intentional in the sense that decisions are made as a result of a reasoning chain performed on driver's mental attitudes, such as beliefs, desires, and intentions. This process ends at pursuing a goal and committing to an attempt at achieving it.

In this work, drivers are dealt with as cognitive entities through the use of a BDI approach, where the internal model of each agent is represented by sets of beliefs, goals, and intentions (see Figure 5.1). The reasoning module of a driver hosts a BDI interpreter, which evaluates driver's mental attitudes in order to make decisions, as initially proposed in (ROSSETTI; BORDINI; BAZZAN; BAMPI, 2001). The presence of communication facilities allows drivers to interact with different IT technologies and in the specific case of this work with ATIS agents, which act as 'mediators' in order to sort out conflicting situations. In this way, some IT agents should have a global model of the world.

The main protagonist within this model is the driver. It is represented in terms of an

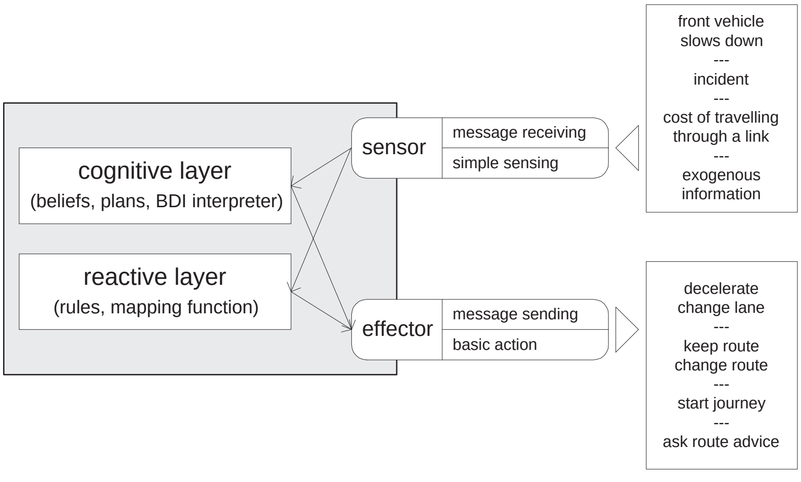

Figure 5.1: The driver agent architecture.

autonomous agent, capable of making decisions on its own. A two-layered architecture is devised to base this model. So, the driver is able to exhibit both reactive and cognitive behaviours. The reactive layer relies on a simple set of rules that map perceptions to actions. The car-following and lane-changing behaviours are implemented in this layer. However, it seems to be not suitable enough to represent more complex decisions, such as whether to travel, which itinerary to follow, and what time to depart at. This is addressed in the cognitive layer instead, which is built on the basis of the BDI logics.

Figure 5.1 roughly depicts the architecture of a driver. As in the basic structure of an agent (RUSSELL; NORVIG, 1995), drivers can perceive facts through sensors and act onto the environment through effectors. The communication ability is also present, which is modelled in terms of message passing. Messages are sent through basic actions and received as perceptions, as suggested in (ROSSETTI; BORDINI; BAZZAN; BAMPI; LIU; VAN VLIET, 2002). When a change in the environment happens, the agents' knowledge base is updated. This can either be associated to the premise of a perception-action rule, in the reactive layer, or trigger a more sophisticated reasoning process, at the cognitive level. At the current stage, the reactive and the cognitive layers are restricted to the supply and demand stages of the simulation, respectively. Nonetheless, implementing a dynamic selection mechanism between these two approaches is the very next step in this research.

According to the AgentSpeak(L) language, the driver agent is represented by the tuple $\langle E, B, P, I, A, \mathscr{S}_{\mathscr{E}}, \mathscr{S}_{\mathscr{O}}, \mathscr{S}_{\mathscr{I}} \rangle$, where E, B, P, I, and A are sets of events, base beliefs, plans, intentions, and basic actions, respectively. $\mathscr{S}_{\mathscr{E}}$, $\mathscr{S}_{\mathscr{O}}$, and $\mathscr{S}_{\mathscr{I}}$ are selection function for events, applicable plans, and intentions, in this order (see Definition 6 in Section 4.6). The task of defining an agent in AgentSpeak(L) is basically reduced to identifying the sets of base beliefs and plans. The perception of triggering events allows intentions to be dynamically generated.

5.3 The BDI Driver Modelled in AgentSpeak(L)

The main reason for modelling driver behaviour on its own right is to provide an adequate means for assessing how individual decision procedures can be affected toward optimising the overall system performance. Expressiveness and scalability are desired features of such a model, which must serve for designing and testing various components of ATIS, such as source, content, and media of the information provided.

Three basic scenarios are possibly envisaged as to whether and when a driver effectively uses exogenous information. In the first and simplest scenario, an individual relies solely on its own cognitive representation of the world. The set of base beliefs of a driver is updated as it executes the trips and evaluates their quality. No exogenous information is made available in this case. In a second scenario, commuters are able to access some sort of exogenous information prior to starting a journey. The content may be tailored to help drivers to plan the journey before departure. Some Internet applications are already available for this purpose (LYONS; MCDONALD, 1998). In a third scenario, travellers are provided with some sort of informative content during the course of a journey. Radio broadcast, variable message signs (VMS), and personalised assistance through dynamic route guidance systems (DRGS) are some examples of sources that can help drivers to constantly evaluate the quality of the journey during its execution. In-trip diversions could be considered for avoiding congested roads and minimising delays. The following AgentSpeak(L) models are aimed at demonstrating the flexibility and expressiveness of Rao's language (RAO, 1996) and its ability to represent the complexity inherent in ITS interactions.

5.3.1 Basic strategies for decision–making

In order to plan a daily journey, as initially suggested in this work, the commuter driver will basically make decisions on what time to depart and which route to take. A cognitive process is carried out in a way so as to find a combination of both allowing drivers to reach destination by a desirable arrival time. Representing the process of such decision-makings is the aim of the following models. To ease the representation of bunches of different behaviours, some characteristics are set within personalities that drive individual's choices. Three initial personalities were devised on the basis of intuitive considerations, namely the random, the choosy, and the conservative drivers. A fourth personality coined the habitual driver is the one currently implemented in DRACULA, as described in (LIU; VAN VLIET; WATLING, 1999). Further extensions to the habitual personality are also proposed. Mathematical formulations of each behaviour are presented prior to specifying commuter scenarios in AgentSpeak(L), which aims to facilitate understanding drivers' personalities and their cognitive mechanisms. A summary of the description of the main symbols used in this section is presented in Table 5.1.

A trip for a driver m, on day k, is given by the tuple $Trip_m^{(k)} = \langle i, j, p, a, r, d \rangle$, where i is the origin, j is the destination, p is the purpose (or the activit to be pursued by the driver at destination), a is the desired arrival time, r is the route to be followed, and d is the departure time at which the journey is suposed to start. Let set $R_{ijm} = \{r_1, r_2, \ldots, r_f, r_{f+1}, \ldots, r_g\}$, where $(f, g) \in \mathbb{N}$, to represent the route options known by a driver m. Thus, each route within R_{ijm}, for instance r_f, is given by a set of adjacent and consecutive links, such that $r_f = \{l_1, l_2, \ldots, l_u, l_{u+1}, \ldots, l_v\}$, where $(u, v) \in \mathbb{N}$. Therefore, the cost for route r_f is the sum of the travel time $\mathscr{T}\mathscr{T}$ of all links within the route, as given in Equation 5.1.

$$\mathscr{C}(r_f) = \sum_{u=1}^{v} \mathscr{T}\mathscr{T}(l_u) \tag{5.1}$$

In order to ease representation, it is convenient to consider some properties for the set R_{ijm}. There is a route $\bar{r}_{ijm}$, such that $\bar{r}_{ijm} \in R_{ijm}$, which represents the usual route from origin i to destination j for a driver m. Also, there is a route $\hat{r}_{ijm}$, such that $\hat{r}_{ijm} \in R_{ijm}$, which represents the best route from i to j and believed by driver m to be the less expensive path in terms of expected travel time. In this work, $\hat{r}_{ijm}$ is assigned randomly to each driver, whereas Expression 5.2 gives the meaning for the best route.

$$\hat{r}_{ijm} = r_f \quad | \quad \mathscr{C}(r_f) = min\{\mathscr{C}(r_1), \ldots, \mathscr{C}(r_g)\} \tag{5.2}$$

Instant and period of time are given in minutes, and are represented by real numbers. Therefore, $(a^{(k)}_{ijm}, d^{(k)}_{ijm}, t^{(k)}_{ijm}) \in \mathbb{R}$, which represent desired arrival time, chosen departure time, and travel time, respectively, for a trip from i to j on day k, relative to a driver m.

5.3.1.1 Random

The characteristic that better describes random drivers is the lack of a specific strategy for choosing a route on each day. An interpretation for such a behaviour can be associated to different activities a driver may decide to perform within the journey, before reaching final destination. For instance, a driver may decide to pass by a service station to supply vehicle with petrol on one day, and opts for a certain path; on the other day the driver may have to drop children at school, and should opt for a different path. In either case, it tends to adjust its departure time accordingly to the estimated travel time for the route chosen. Considering $R_{ijm} = \{r_1, r_2, \ldots, r_f, r_{f+1}, \ldots, r_g\}$, the route choice is made as given in Expression 5.3,

$$r^{(k)}_{ijm} = r_f \quad | \quad \mathscr{P}(r_f) = \frac{1}{g} \tag{5.3}$$

where $\mathscr{P}$ is the probability for a route $r_f \in R_{ijm}$ to be selected, assuming a uniform distribution. After choosing $r^{(k)}_{ijm}$, the departure time is adjusted as a function of the desired arrival time $a^{(k)}_{ijm}$ and the estimated travel cost for $r^{(k)}_{ijm}$, as given in Expression 5.4.

$$d^{(k)}_{ijm} = a^{(k)}_{ijm} - \mathscr{C}(r^{(k)}_{ijm}) \tag{5.4}$$

In this work, the estimated travel cost of a route given by $\mathscr{C}$, is assumed to be the travel time experienced by the driver the last time it travelled through that path.

5.3.1.2 Choosy

A choosy driver is fastidiously selective. It always try to choose the route that is believed to have the lowest travel cost. Considering the set R_{ijm} and the definition for best route given in Equation 5.2, the route choice is simply made as follows.

$$r^{(k)}_{ijm} = \hat{r}_{ijm} \tag{5.5}$$

After selecting route $r^{(k)}_{ijm}$, the departure time choice also follows the same adjustment approach as for random drivers (see Equation 5.4).

5.3.1.3 Conservative

A conservative driver, in turn, is always predisposed to maintain the usual route whatever the cost for that path is believed to be. This could represent the inertia for changing habits that most users familiar with the network very likely exhibit. Considering the set R_{ijm} and the definition for usual route, the route choice is simply made by selecting the usual option, as given in Expression 5.6.

$$r_{ijm}^{(k)} = \bar{r}_{ijm} \tag{5.6}$$

Again, the adjustment for departure time is the same as in random and choosy personalities (see Equation 5.4).

5.3.1.4 Habitual

The habitual personality is proposed on the basis of a decision-making approach currently implemented in DRACULA, as presented in (LIU; VAN VLIET; WATLING, 1995, 1999). According to Ben-Akiva (BEN-AKIVA; DE PALMA; KANAROGLOU, 1986), individuals use information gathered on day k in making their choices on next day $k+1$. Thus, considering that $t_{ijm}^{(k)}$ is the travel time realised for a trip from origin i to destination j, the absolute delay $\delta_{ijm}^{(k)}$ a driver m experiences on day k is given as in Equation 5.7.

$$\delta_{ijm}^{(k)} = d_{ijm}^{(k)} + t_{ijm}^{(k)} - a_{ijm}^{(k)} \tag{5.7}$$

Also, a habitual driver is assumed to be indifferent to a lateness of $\varepsilon_{ijm} \cdot t_{ijm}^{(k)}$. The term ε_{ijm} is a tolerance factor, and in this work it is assumed to be uniform to all drivers in the population. Thus, the perceived lateness $\Delta_{ijm}^{(k)}$ is given as in Equation 5.8.

$$\Delta_{ijm}^{(k)} = \delta_{ijm}^{(k)} - \varepsilon_{ijm} \times t_{ijm}^{(k)} \tag{5.8}$$

As suggested in (LIU; VAN VLIET; WATLING, 1999), drivers are likely to be indifferent to early arrivals. In this sense, travel time on day $k+1$ is adjusted as in Equation 5.9.

$$d_{ijm}^{(k+1)} = \begin{cases} d_{ijm}^{(k)}, & \text{if } \Delta_{ijm}^{(k)} \leq 0 \\ d_{ijm}^{(k)} - \Delta_{ijm}^{(k)}, & \text{if } \Delta_{ijm}^{(k)} > 0 \end{cases} \tag{5.9}$$

The route choice model for the habitual driver, currently implemented in DRACULA, follows the ‘bounded rational choice’ (SIMON, 1956), as suggested by Mahmassani and Jayakrishnan (MAHMASSANI; JAYAKRISHNAN, 1991). Individuals are assumed to use the same route as on the previous day unless the cost expected for the best route (see Equation 5.2) is significantly better. The route choice is set as in Equation 5.10.

$$r_{ijm}^{(k+1)} = \begin{cases} \hat{r}_{ijm}, & \text{if } \mathscr{C}(r_{ijm}^{(k)}) - \mathscr{C}(\hat{r}_{ijm}) > max\{\eta \times \mathscr{C}(r_{ijm}^{(k)}), \tau\} \\ r_{ijm}^{(k)}, & \text{otherwise} \end{cases} \tag{5.10}$$

The parameter η is a threshold level that, according to Mahmassani and Jayakrishnan (MAHMASSANI; JAYAKRISHNAN, 1991), may be interpreted as perceptual factors, preferential indifference, or persistence and aversion to switching with respect to the travel time experienced. On the other hand, τ is an absolute minimum travel time improvement below which driver m will not switch routes. This is also provided in order to retain a meaningful threshold effect and avoid unintended switching for shorter itineraries, for instance.

This model, as it will be seen later on, in Chapter 6, seems to be quite flexible as tolerance is evaluated with respect to the travel time experienced. This means that the longer the trip lasts, the more tolerant the driver will be with regard to lateness. Intuitively, commuters are very unlikely to present such a flexible arrival time. This is especially the case for those making home–work journeys. Moreover the model completely disregards early arrivals, which suggests the need for extending its initial structure toward supporting the definition of an earliness–lateness tolerance window. Such an extended abstraction considers one top lateness and one bottom earliness thresholds, within which no adjustment to departure is required. And in turn, any arrival experience perceived outside these bounds should be considered in future journeys. In this way two variant behaviours are derived from the habitual personality, namely the habitual driver with relative lateness–earliness tolerance window, and the habitual driver with absolute lateness–earliness tolerance window. They both differ from one another basically in terms of how lateness and earliness thresholds are identified. In the former case limits are drawn from the total travel time, whereas in the latter boundaries are given in absolute terms.

As to the relative lateness–earliness window, let λ_{ijm} be the earliness tolerance factor as ε_{ijm} still represents the lateness tolerance factor, both related to a driver m. As in the original habitual behaviour, perceived lateness and earliness will be drawn from trip cost as $\varepsilon_{ijm} \cdot t_{ijm}^{(k)}$ and $\lambda_{ijm} \cdot t_{ijm}^{(k)}$, respectively. The term $t_{ijm}^{(k)}$ refers to the total travel time from i to j on day k. The sign of the absolute delay $\delta_{ijm}^{(k)}$ (as defined in Equation 5.7) is also important as it allows one to identify whether the driver has arrived earlier or later. Bearing in mind the definition for perceived lateness $\Delta_{ijm}^{(k)}$ (see Equation 5.8), let $\Theta_{ijm}^{(k)}$ be the perceived earliness, as given in the Expression 5.11.

$$\Theta_{ijm}^{(k)} = \left|\delta_{ijm}^{(k)}\right| - \lambda_{ijm} \times t_{ijm}^{(k)} \tag{5.11}$$

One should notice that the absolute value of δ is used instead as its sign is negative meaning the agent was earlier. Thus, the departure time on the next day $k+1$ is then adjusted according to the following criterion.

$$d_{ijm}^{(k+1)} = \begin{cases} d_{ijm}^{(k)} - \Delta_{ijm}^{(k)}, & \text{if } \delta_{ijm}^{(k)} > 0 \text{ and } \Delta_{ijm}^{(k)} > 0 \\ d_{ijm}^{(k)} + \Theta_{ijm}^{(k)}, & \text{if } \delta_{ijm}^{(k)} < 0 \text{ and } \Theta_{ijm}^{(k)} > 0 \\ d_{ijm}^{(k)}, & \text{otherwise} \end{cases} \tag{5.12}$$

A similar approach is used for the absolute lateness–earliness window. Let ι_{ijm} be the absolute lateness and ν_{ijm} be the absolute earliness tolerances. Then, the perceived lateness, Δ', and perceived earliness, Θ' are given as in Expression 5.13 and Expression 5.14, respectively.

$$\Delta'^{(k)}_{ijm} = \delta^{(k)}_{ijm} - \iota_{ijm} \tag{5.13}$$

$$\Theta'^{(k)}_{ijm} = \left|\delta^{(k)}_{ijm}\right| - \nu_{ijm} \tag{5.14}$$

The adjustment for departure time on day $k+1$ happens at the same conditions as in the case of the relative lateness–earliness tolerance window (recall Expression 5.14).

$$d^{(k+1)}_{ijm} = \begin{cases} d^{(k)}_{ijm} - \Delta'^{(k)}_{ijm}, & \text{if } \delta^{(k)}_{ijm} > 0 \text{ and } \Delta'^{(k)}_{ijm} > 0 \\ d^{(k)}_{ijm} + \Theta'^{(k)}_{ijm}, & \text{if } \delta^{(k)}_{ijm} < 0 \text{ and } \Theta'^{(k)}_{ijm} > 0 \\ d^{(k)}_{ijm}, & \text{otherwise} \end{cases} \tag{5.15}$$

It is important to notice that both relative and absolute lateness and earliness factors are very likely to depend on the trip rather than being global parameters. In this sense, all factors are given in terms of the origin i and the destination j they are related to.

Table 5.1: Summary of symbols

Symbol	Description
$Trip_m^{(k)}$	trip performed by a driver m on day k
R_{ijm}	set of alternative routes from origin i to destination j known by a driver m
r	a route identifier
l	a link identifier
$\mathscr{T}\mathscr{T}(l)$	travel time through a link l
$\mathscr{C}(r)$	travel cost through a route r
$\bar{r}_{ijm}$	usual route r for a driver m when travelling from origin i to destination j
$\hat{r}_{ijm}$	best route r for a driver m when travelling from origin i to destination j
$a_{ijm}^{(k)}$	desired arrival time for a driver m when travelling from origin i to destination j on day k
$d_{ijm}^{(k)}$	chosen departure time for a driver m when travelling from origin i to destination j on day k
$t_{ijm}^{(k)}$	travel time experienced by a driver m after travelling from origin i to destination j on day k
$\mathscr{P}(r)$	probability for choosing route r
$\delta_{ijm}^{(k)}$	absolute delay experienced by a driver m after travelling from origin i to destination j on day k
ε_{ijm}	lateness tolerance factor, relative to travel time when a driver m is travelling from origin i to destination j
$\Delta_{ijm}^{(k)}$	perceived lateness of a driver m after travelling from origin i to destination j on day k
η	threshold level that forces route switching
τ	absolute minimum travel improvement necessary for route switching
λ_{ijm}	earliness tolerance factor, relative to travel time when a driver m is travelling from origin i to destination j
$\Theta_{ijm}^{(k)}$	perceived earliness of a driver m after travelling from origin i to destination j on day k
ι_{ijm}	absolute lateness tolerance factor, when a driver m is travelling from origin i to destination j
ν_{ijm}	absolute earliness tolerance factor, when a driver m is travelling from origin i to destination j
$\Delta'^{(k)}_{ijm}$	absolute perceived lateness of a driver m after travelling from origin i to destination j on day k
$\Theta'^{(k)}_{ijm}$	absolute perceived earliness of a driver m after travelling from origin i to destination j on day k
$\mathbb{N}$	set of the natural numbers
$\mathbb{R}$	set of the real numbers

5.3.2 The first scenario

In the first scenario, drivers are expected to make every decision with respect to the trip before they start the journey. Afterwards, diversions are not allowed. It is also assumed that no information is provided to individuals, neither before nor during the trip. Therefore, decisions are based solely on their set of base beliefs.

Before going through plan specifications, it is worth to explain some basic conventions. Plans are identified by a number, indicating the scenario, followed by consecutive letters. They are presented in an order so as to ease the explanation of the decision-making process and the group of drivers they are applicable to. For the same reason plans are also gathered in terms of the specific aspect of the decision making process they are designed to address. It is important to notice that AgentSpeak(L) uses similar convention as in the Prolog language, namely variables start with capitals whereas constants and predicates begin with lower-case letters. Thus, as plans are partially instantiated, capitals are used to indicate variable terms. The AgentSpeak(L) syntax is also extended to allow for integer constants, relational operators, and list notation to be used, as suggested in (MACHADO; BORDINI, 2001; ROSSETTI; BORDINI; BAZZAN; BAMPI; LIU; VAN VLIET, 2002). These simple extensions permits more clear specifications. Finally, the symbol $\hookleftarrow$ is used throughout the text to split a long line of text in `typewriter` font. This typographical convention is adopted due to the limits imposed by the left and right margins of the page.

Plan 1.a *Starting the day.* Perceiving that a new day starts is an external stimulus, which is represented by the event $+today(day)$, in Line 1. This perception causes the set of base beliefs to be updated, and triggers a cognitive chain on the new day that starts.

```
01. +today(Day)
02.  : tripInfo(Day, Purpose, ZoneDst, TimeArv)
03.  <- !planTrip(ZoneDst, TimeArv);
04.     !move(ZoneOrg, ZoneDst).
```

If the driver has any motivation for a trip on a certain *day*, expressed by the context in Line 2, it then commits to perform a course of actions that is expected to bring about a desired state of affairs. The belief $tripInfo(day, purpose, zone_{dst}, time_{arv})$ denotes the mental state of the driver with respect to its reason for making a trip. So, on the subject *day*, owing to a given *purpose*, an individual may need to perform a trip to destination $zone_{dst}$. The traveller is expected to arrive at $time_{arv}$ so that the *purpose* for the trip can be satisfied. If there is no reason for making a trip then, the agent's behaviour is not affected at all. This happens when the agent either has no $tripInfo$ entry in its set of base beliefs or no *day* term of any of its $tripInfo$ entries unifies with the value held by the *day* term in $today(day)$. In the case the driver has committed to make the trip, it should set out how to do so, by performing the achievement goal $!planTrip(zone_{dst}, time_{arv})$, and later on by trying to achieve the goal of moving from its current location to its destination site, identified by the achievement goal $!move(zone_{org}, zone_{dst})$ (see Line 4 of Plan 1.a).

It is assumed that there is useful meaning for planning a trip to $zone_{dst}$ only if that is not the agent's current $location(zone_{org})$. Also, the way a traveller carries out the cognitive process to set out the trip depends on its $personality(p)$. Thus, according to the basic strategies presented in Section 5.3.1, p can be any of $\{random, choosy, conservative, habitual\}$, meaning the driver will behave according to the model defined for each personality. To differentiate between the habitual behaviour originally implemented in DRACULA and its extensions as proposed in this work, the notations $habitual_{aw}$ and $habitual_{rw}$ are used to designate whether the driver is adopting an absolute or a relative lateness-earliness tolerance window.

Trip is planned basically by means of choosing a route and a departure time. However, the strategy for *habitual* commuters is slightly different from the formers. While random, choosy, and conservative drivers select the path at first and then the departure time, the habitual drivers do exactly the contrary. This can be represented as in Plans 1.b and 1.c, as follows.

Plan 1.b *Planning the trip.* After realising it is not at destination already, by way of comparing $zone_{org}$ to $zone_{dst}$ in Line 3, the driver seeks to achieve the goals of finding a route to $zone_{dst}$ and a departure time to start the journey, in this order. One should notice that in Line 4 personality is set to *choosy*. This means that only when the belief $personality(choosy)$ holds in the agent's set of base beliefs this plan can be entailed from the agent's knowledge. Thus, the same plan is written to *random* and *conservative* drivers as the condition in Line 4 is replaced by $personality(random)$ and $personality(conservative)$, respectively. This sort of design allows one to specify as many behavioral strategies as it is possible to set out from heterogenous entities of a system.

```
01. +!planTrip(ZoneDst, TimeArv)
02.  : location(ZoneOrg)
03.  & (not(ZoneOrg = ZoneDst))
04.  & personality(choosy)
05.  <- !chooseRoute(choosy, ZoneOrg, ZoneDst);
06.     !chooseDeparture(choosy, ZoneOrg, ZoneDst, TimeArv).
```

Plan 1.c *Planning the trip for habitual drivers.* The only difference for planning a trip between habitual drivers and former personalities is the order as for whether to choose the route or departure time first. As defined in Section 5.3.1.4, habitual drivers set departure prior to choosing a path.

```
01. +!planTrip(ZoneDst, TimeArv)
02.  : location(ZoneOrg)
03.  & (not(ZoneOrg = ZoneDst))
04.  & personality(habitual)
```

```
    <- !chooseDeparture(habitual, ZoneOrg, ZoneDst, TimeArv);
       !chooseRoute(habitual, ZoneOrg, ZoneDst).
```

The achievement goal $!chooseRoute(personality, zone_{org}, zone_{dst})$ induces the driver to find a way of selecting a path from $zone_{org}$ to $zone_{dst}$. The selected course of actions depends on its *personality*, as well. In addition, selecting a departure time to arrive at $zone_{dst}$ by $time_{arv}$, as meant by $!chooseDeparture(personality, zone_{org}zone_{dst}, time_{arv})$, is understood as depending on mental calculation rather than only on a qualitative assessment.

The predicate $route(zone_{org}, zone_{dst}, time_{rte}, [links])$ is used to represent the paths a driver recognises for going from $zone_{org}$ to $zone_{dst}$. An agent has as many clauses of such a kind as is the number of routes it is familiar with. The term $time_{rte}$ denotes the expected travel time associated to the path, which is represented by the list $[links]$ containing consecutive and adjacent segments of roads. For the sake of readability and to ease exchanging parameters among plans, a variable term, say *R*, is used to capture an instance of the link list of $route(zone_{org}, zone_{dst}, time_{rte}, [links])$. So, when *R* is instantiated, it is assigned the content of $[links]$.

The belief predicate $tripRoute(zone_{org}, zone_{dst}, [links])$ is used to represent the particular itinerary chosen by the driver, so as it is able to make the move from $zone_{org}$ to $zone_{dst}$. Thus, whenever a route is selected, the set of base beliefs is updated and the belief *tripRoute* is added to represent the driver's selection.

Plan 1.d *Choosing any route.* A random driver may have several applicable plans for selecting a path as every route clause it knows, from $zone_{org}$ to $zone_{dst}$, will generate an instance for this plan. Thus, the selection function $\mathscr{S}_{\mathcal{O}}$, as defined in Section 4.6, may select any of them. To confirm the selection, an update to the set of base beliefs occurs with the addition of the *tripRoute* clause, in Line 3.

```
01. +!chooseRoute(random, ZoneOrg, ZoneDst)
02.  : route(ZoneOrg, ZoneDst, TimeRte, R)
03.  <- +tripRoute(ZoneOrg, ZoneDst, R).
```

The belief predicate $expectedTravelTime(zone_{org}, zone_{dst}, time)$ is used to represent the expected travel time from $zone_{org}$ to $zone_{dst}$ when no path is considered. In other words, a driver may have an estimation of the necessary time to reach a certain destination without considering any route, at a first glance.

Plan 1.e *Finding out the best route.* In the context part, the agent evaluates whether the travel time for the route it is considering for selection is lower than the travel time it is currently expecting to experience. In a first execution of this sub-plan, before considering any alternative, the driver has a "pessimistic behaviour" and sets a very high value for its expected travel time (so as to set reasonable expected times after whatever first attempt is made). The agent keeps making attempts at finding the best route by recalling the plan in Line 7. One must notice that the context part of the plan, as in Line 4, will not be satisfied once the best route has been found, thus finishing the recursion.

```
01. +!chooseRoute(choosy, ZoneOrg, ZoneDst)
02.  : route(ZoneOrg, ZoneDst, TimeRte, R)
03.  & expectedTravelTime(ZoneOrg, ZoneDst, TimeExpected)
04.  & (TimeRte < TExpected)
05.  <- +tripRoute(ZoneOrg, ZoneDst, R);
06.     +expectedTravelTime(ZoneOrg, ZoneDst, TimeRte);
07.     !chooseRoute(choosy, ZoneOrg, ZoneDst).
```

Plan 1.f *Keeping the instantiation for the best route*. This plan finally keeps the instantiations of the appropriate parameters for $tripRoute$ when the recursion in Plan 1.e finishes. Thus, when all attempts to satisfy the context part of Plan 1.e fail, the best route then instantiated is the one to be chosen.

```
01. +!chooseRoute(choosy, ZoneOrg, ZoneDst)
02.  : true <- true.
```

Plan 1.g *Choosing the usual route*. As stipulated for the conservative behaviour (see Section 5.3.1.2), each driver in the population is assigned a preferred route from $zone_{org}$ to $zone_{dst}$. The belief predicate $usualRoute(zone_{org}, zone_{dst}, [links])$ is used to identify the agent's usual path within a given OD pair. The decision is as simple as setting the usual path as the trip route, as in Line 3.

```
01. +!chooseRoute(conservative, ZoneOrg, ZoneDst)
02.  : usualRoute(ZoneOrg, ZoneDst, R)
03.  <- +tripRoute(ZoneOrg, ZoneDst, R).
```

Choosing a route option for habitual drivers demands some further considerations, though. Both the threshold level η and the absolute improvement τ seem to be rather related to the trip, as presented in Section 5.3.1.4. This is also the case for the tolerance factor ε. Hence, all these parameters were considered to be better identified as terms within the $tripInfo$ belief predicate. So, all the information on the journey due on a certain day is memorised as $tripInfo(day, purpose, zone_{dst}, time_{arv}, \varepsilon, \eta, \tau)$. Yet another important consideration is that habitual drivers choose their routes on the perspective of both the best and the usual path, so both should be identified.

Plan 1.h *Identifying the best route by habitual drivers*. Habitual drivers identify the best route following the same recursive design of Plan 1.e, except for the fact that the personality is $habitual$. The addition of the $tripRoute$ belief in the agent's knowledge base works a sort of as a temporary container of the decision until a final choice is made.

```
01. +!chooseRoute(habitual, Zone$_{Org}$, Zone$_{Dst}$)
02.  : route(Zone$_{Org}$, Zone$_{Dst}$, Time$_{Rte}$, R)
03.  & expectedTravelTime(Zone$_{Org}$, Zone$_{Dst}$, Time$_{Expected}$)
04.  & (Time$_{Rte}$ < Time$_{Expected}$)
05.  <- +tripRoute(Zone$_{Org}$, Zone$_{Dst}$, R);
06.     +expectedTravelTime(Zone$_{Org}$, Zone$_{Dst}$, Time$_{Rte}$);
07.     !chooseRoute(habitual, Zone$_{Org}$, Zone$_{Dst}$).
```

Plan 1.i *Assessing best route with respect to the absolute improvement τ.* After evaluating $max\{\eta \times \mathscr{C}(r_{ijm}^{(k)}), \tau\}$ in Line 7, the agent decides whether the improvement is good enough for a route switch, in Line 8, so as to keep the best route and terminate recursion.

```
01. +!chooseRoute(habitual, Zone$_{Org}$, Zone$_{Dst}$)
02.  : tripInfo(Day, Purpose, Zone$_{Dst}$, Time$_{Arv}$, ↩
       Epsilon, Eta, Tau)
03.  & tripRoute(Zone$_{Org}$, Zone$_{Dst}$, R$_{Bst}$)
04.  & route(Zone$_{Org}$, Zone$_{Dst}$, Time$_{Bst}$, R$_{Bst}$)
05.  & usualRoute(Zone$_{Org}$, Zone$_{Dst}$, R$_{Usl}$)
06.  & route(Zone$_{Org}$, Zone$_{Dst}$, Time$_{Usl}$, R$_{Usl}$)
07.  & ((Eta * Time$_{Usl}$) < Tau)
08.  & ((Time$_{Usl}$ - Time$_{Bst}$) > Tau)
09.  <- true.
```

Plan 1.j *Assessing best route with respect to the threshold level η.* As in Plan 1.i, this plan evaluates the improvement expected, in Line 8, after checking $max\{\eta \times \mathscr{C}(r_{ijm}^{(k)}), \tau\}$ in Line 7, and decides to switch.

```
01. +!chooseRoute(habitual, Zone$_{Org}$, Zone$_{Dst}$)
02.  : tripInfo(Day, Purpose, Zone$_{Dst}$, Time$_{Arv}$, ↩
       Epsilon, Eta, Tau)
03.  & tripRoute(Zone$_{Org}$, Zone$_{Dst}$, R$_{Bst}$)
04.  & route(Zone$_{Org}$, Zone$_{Dst}$, Time$_{Bst}$, R$_{Bst}$)
05.  & usualRoute(Zone$_{Org}$, Zone$_{Dst}$, R$_{Usl}$)
06.  & route(Zone$_{Org}$, Zone$_{Dst}$, Time$_{Usl}$, R$_{Usl}$)
07.  & ((Eta * Time$_{Usl}$) >= Tau)
08.  & ((Time$_{Usl}$ - Time$_{Bst}$) > (Eta * Time$_{Usl}$))
09.  <- true.
```

Plan 1.k *Choosing the usual route.* In the case any improvement is to be discarded as Plans 1.i and 1.j fails to instantiate, the habitual drivers choose the usual route in the same way as conservative travellers (see Plan 1.g). This plan also terminates the recursion started in Plan 1.h, as would Plan 1.i and Plan 1.j if either succeeded.

```
01. +!chooseRoute(habitual, Zone_Org, Zone_Dst)
02.  : usualRoute(Zone_Org, Zone_Dst, R)
03.  <- +tripRote(Zone_Org, Zone_Dst, R).
```

Plan 1.l *Choosing departure time.* Random, choosy, and conservative behaviours implement the same strategy for selecting departure time. The only difference is that Line 1 should be rewritten with the correct *personality* term. The predicate *tripDeparture* $(zone_{org}, zone_{dst}, time_{dpt})$ is used to denote the agent intends to start its journey from origin $zone_{org}$ to destination $zone_{dst}$ at $time_{dpt}$.

```
01. +!chooseDeparture(random, Zone_Org, Zone_Dst, Time_Arv)
02.  : tripRoute(Zone_Org, Zone_Dst, R)
03.  & route(Zone_Org, Zone_Dst, Time_Rte, R)
04.  <- +tripDeparture(Zone_Org, Zone_Dst, (Time_Arv - Time_Rte)).
```

As for habitual drivers, one should bear in mind that the decision-making process on departure time is carried out prior to selecting the journey path. Thus, Plans 1.h, 1.i, 1.j, and 1.k are evaluated after the agent has gone through Plans 1.m and 1.n. As it was originally proposed for this behaviour (see Section 5.3.1.4), a lateness tolerance with respect to the travel time in the previous journey (the ε factor) should be taken into account, as well. This is assumed to be processed by the driver as pure mental calculation, hence some intermediate parameters are only identified on on-the-fly rather than stored in bunches of other belief clauses.

Plan 1.m *Evaluating scheduled delay.* As already mentioned above, the tolerance factor ε is associated to the trip and its purpose rather than being considered to be an independent parameter. Hence ε is retrieved from within the *tripInfo* belief clause. It is important to notice that departure, path, and travel time of the previous journey are given within belief predicates *tripRoute*, *route*, and *tripDeparture*, in Lines 3, 4, and 5. The absolute delay δ is represented by the predicate *arrivalCost*$(zone_{dst}, time_{dly})$, in Line 6, with respect to the actual arrival at $zone_{dst}$ the last time a trip was executed to that destination. The traveller evaluates whether the perceived delay Δ is significant, in Line 7, and updates the *tripDeparture* for next journey accordingly.

```
01. +!chooseDeparture(habitual, Zone_Org, Zone_Dst, Time_Arv)
02.  : tripInfo(Day, Purpose, Zone_Dst, Time_Arv, ↩
       Epsilon, Eta, Tau)
03.  & tripRoute(Zone_Org, Zone_Dst, R)
04.  & route(Zone_Org, Zone_Dst, Time_Rte, R)
```

```
05.  & tripDeparture(Zone_Org, Zone_Dst, Time_Dpt)
06.  & arrivalCost(Zone_Org, Time_Dly)
07.  & ((Time_Dly - (Epsilon * Time_Rte)) > 0.0)
08.  <- +tripDeparture(Zone_Org, Zone_Dst, (Time_Dpt - ↩
        (Epsilon * Time_Rte))).
```

Plan 1.n *Keeping previous departure time.* In the case the perceived delay is considered to be irrelevant, as Line 7 of Plan 1.m fails and the plan is not considered to be applicable, the driver then keeps the previous departure choice.

```
01. +!chooseDeparture(habitual, Zone_Org, Zone_Dst, Time_Arv)
02.  : true <- true.
```

The same approach used for the original habitual behaviour is applied to the extensions proposed in Section 5.3.1.4, namely the habitual behaviour with a relative lateness–earliness tolerance window and the one with an absolute lateness–earliness tolerance window. So, ε, λ, ι, and ν factors are to be included into the $tripInfo$ predicate associated to the respective behaviour; $tripInfo(day, purpose, zone_{dst}, time_{arv}, \varepsilon, \lambda, \eta, \tau)$ and $tripInfo(day, purpose, zone_{dst}, time_{arv}, \iota, \nu, \eta, \tau)$ denotes the trip information for the relative and absolute window-based behaviours, respectively.

Plan 1.o *Adjusting departure according to relative lateness.* The belief given by the predicate $arrivalCost(zone_{dst}, time_{dly})$ serves as an indicator of whether the driver is late or early, as $\delta < 0$ or $\delta > 0$. This is tested in Line 7, whereas the perceived lateness Δ is evaluated in Line 8. In the case of being late, which means the driver has arrived beyond the top boundary of the relative lateness-earliness window, then departure is adjusted.

```
01. +!chooseDeparture(habitual_rw, Zone_Org, Zone_Dst, Time_Arv)
02.  : tripInfo(Day, Purpose, Zone_Dst, Time_Arv, ↩
       Epsilon, Lambda, Eta, Tau)
03.  & tripRoute(Zone_Org, Zone_Dst, R)
04.  & route(Zone_Org, Zone_Dst, Time_Rte, R)
05.  & tripDeparture(Zone_Org, Zone_Dst, Time_Dpt)
06.  & arrivalCost(Zone_Org, Time_Dly)
07.  & (Time_Dly < 0.0)
08.  & (((abs(Time_Dly)) - (Epsilon * Time_Rte)) > 0.0)
09.  <- +tripDeparture(Zone_Org, Zone_Dst, (Time_Dpt - ↩
        ((abs(Time_Dly)) - (Epsilon * Time_Rte))).
```

Plan 1.p *Adjusting departure according to relative earliness.* In the case of being early, as evaluated in Line 7, the agent checks whether Θ is significant and it is worth to change departure time, in Line 8.

```
01. +!chooseDeparture(habitual$_{rw}$, Zone$_{Org}$, Zone$_{Dst}$, Time$_{Arv}$)
02.  : tripInfo(Day, Purpose, Zone$_{Dst}$, Time$_{Arv}$, ↩
       Epsilon, Lambda, Eta, Tau)
03.  & tripRoute(Zone$_{Org}$, Zone$_{Dst}$, R)
04.  & route(Zone$_{Org}$, Zone$_{Dst}$, Time$_{Rte}$, R)
05.  & tripDeparture(Zone$_{Org}$, Zone$_{Dst}$, Time$_{Dpt}$)
06.  & arrivalCost(Zone$_{Org}$, Time$_{Dly}$)
07.  & (Time$_{Dly}$ > 0.0)
08.  & ((Time$_{Dly}$ - (Lambda * Time$_{Rte}$)) > 0.0 )
09.  <- +tripDeparture(Zone$_{Org}$, Zone$_{Dst}$, (Time$_{Dpt}$ + ↩
          (Time$_{Dly}$ - (Lambda * Time$_{Rte}$)))).
```

Plan 1.q *Adjusting departure according to absolute lateness*. As in Plans 1.o and 1.p, the value of δ is used to evaluate, in Line 7, whether the driver is late or early. On the basis of an absolute top lateness threshold, the driver considers whether Δ' is tolerable, in Line 8.

```
01. +!chooseDeparture(habitual$_{aw}$, Zone$_{Org}$, Zone$_{Dst}$, Time$_{Arv}$)
02.  : tripInfo(Day, Purpose, Zone$_{Dst}$, Time$_{Arv}$, ↩
       Iota, Nu, Eta, Tau)
03.  & tripRoute(Zone$_{Org}$, Zone$_{Dst}$, R)
04.  & route(Zone$_{Org}$, Zone$_{Dst}$, Time$_{Rte}$, R)
05.  & tripDeparture(Zone$_{Org}$, Zone$_{Dst}$, Time$_{Dpt}$)
06.  & arrivalCost(Zone$_{Org}$, Time$_{Dly}$)
07.  & (Time$_{Dly}$ < 0.0)
08.  & (((abs(Time$_{Dly}$)) - Iota) > 0.0 )
09.  <- +tripDeparture(Zone$_{Org}$, Zone$_{Dst}$, (Time$_{Dpt}$ - ↩
          ((abs(Time$_{Dly}$)) - Iota))).
```

Plan 1.r *Adjusting departure according to absolute earliness*. For early arrivals, as the agent realises $\delta > 0$ in Line 7, the perceived earliness is evaluated with respect to the bottom threshold ν in Line 8, and the departure time is adjusted accordingly.

```
01. +!chooseDeparture(habitual$_{aw}$, Zone$_{Org}$, Zone$_{Dst}$, Time$_{Arv}$)
02.  : tripInfo(Day, Purpose, Zone$_{Dst}$, Time$_{Arv}$, Iota, ↩
       Nu, Eta, Tau)
03.  & tripRoute(Zone$_{Org}$, Zone$_{Dst}$, R)
04.  & route(Zone$_{Org}$, Zone$_{Dst}$, Time$_{Rte}$, R)
05.  & tripDeparture(Zone$_{Org}$, Zone$_{Dst}$, Time$_{Dpt}$)
06.  & arrivalCost(Zone$_{Org}$, Time$_{Dly}$)
07.  & (Time$_{Dly}$ > 0.0)
```

```
08.  & ((Time_Dly - Nu) > 0.0)
09.  <- +tripDeparture(Zone_Org, Zone_Dst, (Time_Dpt + ↩
         (Time_Dly - Nu))).
```

If the arrival time suits either the relative or the absolute lateness-earliness tolerance window, as neither Plans 1.o and 1.p nor Plans 1.q and 1.r are considered to be applicable, the driver agent keeps its previous departure choice. This decision is confirmed by means of a plan similar to Plan 1.n. The terms $habitual_{rw}$ and $habitual_{aw}$ should be used accordingly, though.

After planing the trip, by means of selecting a route and a departure time, the driver seeks to reach destination, as denoted by the achievement goal $!move(zone_{org}, zone_{dst})$ in Line 4 of Plan 1.a. The belief $timeNow(time)$ is used to represent the notion of instant. Therefore, as time goes by the knowledge base of an individual is constantly being updated as for such a belief.

Plan 1.s *Moving to destination.* The agent pursues this plan until time for departure is perceived and the context part, as conditioned in Line 5, is satisfied.

```
01. +!move(Zone_Org, Zone_Dst)
02.  : tripRoute(Zone_Org, Zone_Dst, R)
03.  & tripDeparture(Zone_Org, Zone_Dst, Time_Dpt)
04.  & timeNow(Time)
05.  & (Time = Time_Dpt)
06.  <- moveAlong(R).
```

Plan 1.t *Ending day trip.* Reaching destination is perceived as an external event by means of sensing the new current location, as given in Line 1. The traveller then checks the time (the instant of arrival) by performing the test goal $?timeNow(time)$ in Line 5. As a result, both $arrivalCost(zone_{dst}, time_{dly})$ and path conditions as given in $route(zone_{org}, zone_{dst}, time_{rte}, [links])$ are updated so as to reflect the driver's day experience. One should notice the $tripInfo$ is in its simplest form, which means this plan is only applicable for random, choosy, and conservative drivers. Line 2 should then be rewritten for habitual drivers, though.

```
01. +location(Zone_Dst)
02.  : tripInfo(Day, Purpose, Zone_Dst, Time_Arv)
03.  & tripRoute(Zone_Org, Zone_Dst, R)
04.  & tripDeparture(Zone_Org, Zone_Dst, Time_Dpt)
05.  <- ?timeNow(Time_Dst);
06.     +arrivalCost(Zone_Dst, (Time_Dst - Time_Arv));
07.     +route(Zone_Org, Zone_Dst, (Time_Dst - Time_Dpt), R).
```

5.3.3 Communication and exogenous information

Before going on with further plan specifications, it is important to make some considerations on communication abilities in the multi-agent traffic system. Inter-agent communication is accomplished through message passing in the approach adopted in this thesis. Sending a message is assumed to be a basic action, whereas receiving a message causes, as with perception, the addition of a belief entry in the agent's set of base beliefs. Receiving a message may in turn trigger a plan execution. Some trip planner applications on the Internet are examples of information sources from which travellers can enhance knowledge on the current prevailing conditions of the network.

The communication mechanisms is assumed to be as follows. Consider that $b(t_1,\ldots,t_n)$ is a belief predicate as presented in Definition 1 (see Section 4.6). Thus, $communicate(ag,\text{“}b(t_1,\ldots,t_n)\text{”})$, $request(ag,\text{“}b(t_1,\ldots,t_n)\text{”})$, and $broadcast(\text{“}b(t1,\ldots,tn)\text{”})$ are special cases of basic action predicates. The term *ag* is used to identify the agent to which the message is addressed, whereas “$b(t_1,\ldots,t_n)$” represents the propositional content of the message. The *communicate* predicate is used to send *ag* the belief $b(t_1,\ldots,t_n)$. The *request* predicate asks agent *ag* for $b(t_1,\ldots,t_n)$. In this case, it is assumed that *ag* (as an information system, for instance) presents a “benevolent” behaviour and always replies to the request made (as by a driver, for instance). There is no agent addressed in the broadcast predicate, though. In such a case, the content is sent to all of the agents in the multi-agent system.

To illustrate such a communication mechanism, let a_1 to be an ATIS agent that sends a message $b(\mathbf{t})$ to a traveller, which is a_2. Then, a_1 executes $communicate(a2,\text{“}b(\mathbf{t})\text{”})$ (or $broadcast(\text{“}b(\mathbf{t})\text{”})$ to all drivers in the system). When a_2 receives the message, an event $+b(\mathbf{t})$ occurs and $b(\mathbf{t})$ is added to the set of its base beliefs, as with perception. So, a_2 cannot distinguish whether $+b(\mathbf{t})$ is a simple perception or a message passed through communication. In such a situation, it is also assumed that the *belief revision function* checks whether a_1 is trustworthy. If a_1 is considered to be trustworthy, besides adding $b(\mathbf{t})$ to the belief base, the function adds another belief predicate, $informed(a_1,b(\mathbf{t}))$, indicating who has informed a_2 about $b(\mathbf{t})$. Thus, it is possible to have plans associated both to the content of the message and to the informing agent, if considering the sender is relevant.

With respect to interactive information sources, two major groups of drivers are identified. The first group is formed of drivers who are eligible to use the information system, either because they are subscribers or because they are equipped to receive the information in an interactive way. Those drivers who are not users of information systems are gathered into the second group. In order to consider such a relation in the model, the belief predicates $preTripInfoUser()$ and $enRouteInfoUser()$ are used to identify whether drivers are users of pre-trip and en-route information systems, respectively.

However, saying whether the driver is user or non-user of information systems is not meaningful enough to guarantee that the information provided will be considered in the decision-making process. In order to make this criterion complete, the belief predicate $acceptanceWillingness(value)$ is used to represent whether the driver is willing to accept and to use the information provided. The term $value$ is a random number used to capture the probability of a driver to accept an advice, which may be associated to its personality in some way. A combination of both being user of information sources and willingness of using the content provided can be the way toward accessing contemporary ATIS technologies.

On the other hand, non-interactive information sources are aimed at reaching most users. Contrary to interactive systems, as drivers are able to make a request in order to get information, communication happens in one direction only. Good examples of this kind of sources are the mass media, as newspapers, radio, or television. Traffic signs and recently VMS also have the ability to reach most drivers travelling throughout the network. In the commuter world herein proposed, information provided is mapped to either link or route states. In practical terms, drivers will associate the content of information to possible states for links and routes. For example, whenever a belief $linkState(link, state)$ is updated, an event is generated to indicate that a message has been received; in this case it is just a reminder that some $link$ has a certain prevailing $state$, not a response to a request previously made. As mentioned before, it is the *belief revision function* that checks the trustworthiness of the information source, and an event of the type $+informed(ag, b(\mathbf{t}))$ is also posted to the base beliefs. At that moment, the driver may either accept or ignore it, on the basis on its acceptance willingness.

5.3.4 The second scenario

In the second scenario drivers are allowed to access information before starting a trip, which is expected to improve the decision-making process. No en-route diversion is possible, though. It is also assumed that users of exogenous information cannot interact with sources, and content may be regarded to either link or route states. In addition to the predicate $linkState(link, state)$ and in order to distinguish between qualitative and quantitative notions of routes, a predicate $routeState([links], state)$ is used to denote that a given path $[links]$ is found to be in a certain prevalent $state$. Hence whenever a driver receives information prior to starting a journey, it may consider the option of avoiding the corresponding path on the basis of its acceptance willingness.

Plan 2.a *Using information on link state.* Considering the current state of a certain $link$ is found to be “congested” and that it has been broadcast, a clause $linkState(link, congested)$ is posted to a pre-trip information user’s set of base beliefs. In such a situation the driver may consider the chance of selecting the best path it knows from $zone_{org}$ to $zone_{dst}$ to avoid the warned link. Then, finding the best route follows the same iteration as previously suggested in the first scenario. The agent tries to figure out whether the $link$ warned is in [*links*], so as to avoid it (Lines 3 and 4). The predicate *in*[1] (see Line 4) is a function that evaluates to *true* if L is in set R and to $false$ otherwise. This plan could be valid for any *personality*, as represented by the variable term in Line 1.

```
01. +!chooseRoute(Personality, ZoneOrg, ZoneDst)
02.  : route(ZoneOrg, ZoneDst, TimeRte, R)
03.  & linkState(Link, congested)
04.  & (not(in(Link, R)))
05.  & expectedTravelTime(ZoneOrg, ZoneDst, TimeExpected)
06.  & (TimeRte < TimeExpected)
07.  & preTripInfoUser()
08.  & acceptanceWillingness(V) & (V <= threshold)
09.  <- +tripRoute(ZoneOrg, ZoneDst, R);
10.     +expectedTravelTime(ZoneOrg, ZoneDst, TimeRte);
11.     !chooseRoute(Personality, ZoneOrg, ZoneDst).
```

Plan 2.b *Keeping the best alternative with regard to the warned link.* Using the same approach as in Plan 1.f, the best route as stored in *tripRoute* is confirmed, if any is found to not passing through the warned *link*.

```
01. +!chooseRoute(Personality, ZoneOrg, ZoneDst)
02.  : preTripInfoUser()
03.  & acceptanceWillingness(V) & (V <= threshold)
04.  <- true.
```

Plan 2.c *Selecting the usual route instead.* In the case of lacking a better alternative, the option of using the usual path is adopted in a plan similar to Plan 1.g.

```
01. +!chooseRoute(Personality, ZoneOrg, ZoneDst)
02.  : preTripInfoUser()
03.  & acceptanceWillingness(V) & (V <= threshold)
04.  & usualRoute(ZoneOrg, ZoneDst, R)
05.  <- +tripRoute(ZoneOrg, ZoneDst, R).
```

Plan 2.d *Using information on route state.* In this case, the information broadcast causes $routeState([links], congested)$ to be added to the agent's base beliefs. The only difference from Plan 2.a is that routes are evaluated with respect to a warned route, therefore having a different condition in the context part as in Lines 3 and 4.

[1] The function predicate *in* verifies whether $L \cap R \neq \emptyset$.

```
+!chooseRoute(Personality, Zone$_{Org}$, Zone$_{Dst}$)
 : route(Zone$_{Org}$, Zone$_{Dst}$, Time$_{Rte}$, R)
 & routeState(R$_{warned}$, congested)
 & (not(R = R$_{warned}$))
 & expectedTravelTime(Zone$_{Org}$, Zone$_{Dst}$, Time$_{Expected}$)
 & (Time$_{Rte}$ < Time$_{Expected}$)
 & preTripInfoUser()
 & acceptanceWillingness(V) & (V <= threshold)
 <- +tripRoute(Zone$_{Org}$, Zone$_{Dst}$, R);
    +expectedTravelTime(Zone$_{Org}$, Zone$_{Dst}$, Time$_{Rte}$);
    !chooseRoute(Personality, Zone$_{Org}$, Zone$_{Dst}$).
```

Plans 2.b and 2.c are used in the same way to confirm the best path with respect the the route information provided (if any is found) and the selection of the usual choice if no alternative is available, respectively. In the case of being *preTripInfoUser* and not willing to use the information provided, the agent may adopt its normal decision-making strategy as defined for its corresponding personality model.

5.3.5 The third scenario

In the third scenario, drivers can receive information both prior to starting (on the basis of what has been discussed for the second scenario) and during the course of a journey. VMS and DRGS are examples of exogenous sources that can be used during the trip. In addition drivers can also interact with the provider as it is allowed to ask for advice, as well. Being able to divert original route choice to resume the journey through an alternative path would imply that the BDI driver was able to play its cognitive abilities within the microscopic simulation of the movement, not only in trip-planing time.

Contrary to what has been assumed in the two previous scenarios, the formulation for the basic action predicate $moveAlong(route)$ should be updated to $moveAlong(link)$. Such an adaptation is important so as to allow for representing the movement on a link-by-link basis.

Every time the *moveAlong* action is executed, the driver is able to register its experience through that link, mainly on the basis of the travel cost observed (such as travel time, delays, and so on). After performing the journey through a link, the set of base beliefs is updated through the perception of the event $+location(link)$, as a result of such an action. Then, the driver is positioned at the upstream node of the next *link* getting ready to move again to its downstream node, and so on, so forth. In this way, drivers are able to re-evaluate the quality of the trip during the journey.

Plan 3.a *Starting the trip.* At departure time, the driver enters the network to effectively execute the trip planned. Both $zone_{org}$ and $zone_{dst}$ are said to be dummy links in the sense they are used just to connect origin and destination to the network, respectively.

Therefore, drivers will actually move only through the $[links]$ in the path R[2]. As the driver sets its location to the first link of the path, in Line 8, the links in the tail of the list are regarded as the remaining path for trip completion.

```
01. +!move(Zone_Org, Zone_Dst)
02.  : tripRoute(Zone_Org, Zone_Dst, R)
03.  & tripDeparture(Zone_Org, Zone_Dst, Time_Dpt)
04.  & timeNow(Time)
05.  & (Time = Time_Dpt)
06.  & (R = [L|Links])
07.  <- +tripRoute(Zone_Org, Zone_Dst, Links);
08.     moveAlong(L).
```

The Prolog notation is used to manipulate lists. Thus, the formula $R = [L|Links]$ in the context part of Plan 3.a, in Line 6, is used to instantiate the variables L and $Links$, which are the first and the remaining links of R, respectively. Hence, the $tripRoute$ is updated in Line 7. This is done to represent the fact that drivers keep attention on the links to come, whereas passed links are left behind.

Plan 3.b *Moving through the remaining links of the route*. Whenever the driver updates its current location to the next link within the path to go, it effectively moves as the basic action *moveAlong* is invoked in Line 5. This action causes an updating to the agent's location to the next link in *Links*, which is again perceived as $+location(L)$ is added to its base beliefs, after the action is terminated. In this way, an individual keeps moving until reaching destination. The remaining path is also updated in the agent's base beliefs, in Line 4.

```
01. +location(L)
02.  : tripRoute(Z_Org, Z_Dst, R)
03.  & (R = [L|Links])
04.  <- +tripRoute(Z_Org, Z_Dst, Links);
05.     moveAlong(L).
```

Plan 3.c *Reaching destination*. After performing all links within the path, the agent gets to the end of the journey. This is identified when moving through the last link in R results in positioning the agent in the connector link $Zone_{Dst}$, thus updating the base beliefs by $+location(Zone_{Dst})$.

```
01. +location(Zone_Dst)
02.  : tripRoute(Zone_Org, Zone_Dst, R)
03.  <- true.
```

[2] It is important to remember that $R = [links]$ has been adopted to ease representation.

Such a cognitive representation of movent in this third scenario is essential so that drivers can consider diversions as exogenous information is provided.

Plan 3.d *Making a request during the course of a trip.* At reaching a new link within the trip path, the driver may consider to ask for an aid (to DRGS, for instance). In this case an alternative route from the agent's current location to its final destination is considered. Such information is provided by the system under request from the driver, as represented in Line 6 by a basic action invocation. Then the agent expects to receive from the source the clause $sysRoute(origin, destination, [links])$, which is an alternative suggestion from the information system.

```
01. +location(L)
02.  : tripRoute(Zone_Org, Zone_Dst, R)
03.  & (R = [L|Links])
04.  & enRouteInfoUser()
05.  & acceptanceWillingness(V) & (V <= threshold)
06.  <- request(atis, "sysRoute(L, Zone_Dst, R_Sys)");
07.     +tripRoute(Zone_Org, Zone_Dst, Links);
08.     moveAlong(L).
```

Plan 3.e *Accepting the information requested.* When the answer arrives, as $+sysRoute(origin, destination, [links])$ is perceived, the driver may consider either to accept it or to retain its original choice. However, accepting the suggested itinerary depends very much on whether it is still meaningful for use. Otherwise, it is automatically discarded, as the context part of the plan will not be satisfied. In other words, this plan is only applicable if the driver is still moving through the link L.

```
+sysRoute(L, Zone_Dst, R_Sys)
 : tripRoute(Zone_Org, Zone_Dst, R)
 & location(L)
 & enRouteInfoUser()
 & acceptanceWillingness(V) & (V <= threshold)
 <- +tripRoute(Zone_Org, Z_Dst, R_Sys).
```

Notice that the remaining path is updated with the itinerary suggested, which is sufficient to guarantee the driver will update its future location accordingly when the current execution of action $moveAlong(link)$ terminates.

Plan 3.f *Ignoring the information requested.* The plan to ignore the information provided is as simple as doing nothing.

```
+sysRoute(L, Zone_Dst, R_Sys)
```

```
: enRouteInfoUser()
& acceptanceWillingness(V) & (V > threshold)
<- true.
```

5.4 Summary

Modelling and simulating traffic and transportation systems can undoubtedly profit from MAS-based methodologies. The abstraction offered by multi-agent approaches allows for representing most entities and processes in the application domain in a straightforward manner. Most important, it preserves hierarchical configurations and interactions. Its ability to mimic cognitive reasoning and knowledge representation gives traditional structures of drivers an ideal framework to experiment and investigate humanlike behaviour. Moreover, as the number of autonomous and intelligent artifacts used to interact within the contemporary traffic systems increases, it is imperative to extend traditional modelling and simulating methodologies to contemplate the new performance measures brought about by ITS. This way, theory should rely on an adequate means to implement, to validate, and to deploy such advanced technologies.

Endowing the driver structure with a BDI reasoning kernel has facilitated the representation of knowledge and cognition. Three commuter scenarios were devised by means of using AgentSpeak(L) as an specification tool, as suggested by Machado and Bordini (2001) and following the same modelling approach initially presented in (ROSSETTI; BORDINI; BAZZAN; BAMPI; LIU; VAN VLIET, 2002; ROSSETTI; LIU; CYBIS; BAMPI, 2002). The scenarios are intended to cover different aspects of contemporary traffic systems, mainly with regard to human behaviour and its interaction with advanced technologies. The methodology seems to be flexible to support the representation of different driver profiles and decision-making strategies within several personalities. The predicate logics used in the BDI architecture turns knowledge representation closer to humanlike cognition. Nonetheless, the plans presented in this chapter do not represent a unique design alternative, but rather are used to demonstrate the potential of AgentSpeak(L) to represent and to specify the complexity that is inherent in real systems, such as the traffic and transportation domain. In addition, a purpose-built interpreter integrated within a simulation framework, could turn AgentSpeak(L) into a powerful API for developing and testing different behavioural approaches for ITS assessment.

6 MADAM+DRACULA: A FRAMEWORK TO ASSESS VARIABLE DEMAND

6.1 Overview

The MADAM (**M**ulti-**A**gent **D**em**A**nd **M**odel) model is devised on the basis of the cognitive methodological approach presented in the previous chapter. Starting from the perspective of seeing contemporary traffic systems and ITS technologies as a multi-agent world, drivers are represented in terms of cognitive agents. This abstraction is the relying approach used to build a population of BDI commuter drivers, which are capable of making decisions on the basis of mental attitudes such as beliefs, desires, and intentions. Then, travels are generated as the result of decisions made as to which route to take and what time to depart.

In order to demonstrate the methodological approach suggested in this thesis, MADAM was integrated into the microscopic simulation environment of DRACULA (**D**ynamic **R**oute **A**ssignment **C**ombining **U**ser **L**earning and microsimul**A**tion). This way the BDI commuters can perform their trips by means of carrying out their journeys through the selected path on a vehicle-by-vehicle basis. Such a microscopic simulated environment allows individuals to evaluate the quality of their decisions day after day. Some experiments were designed on the basis of the first and the second scenarios specified in AgentSpeak(L) and simulated in the MADAM+DRACULA framework. Aggregate travel time is the main performance measure used in the discussions of simulated results.

6.2 The DRACULA model

DRACULA is a framework in which special emphasis is given to the microscopic simulation of individual trip makers and individual vehicles. This environment comprises basically two main models: the demand and the supply. Both of them are based on a microscopic simulation approach. In the demand model, travellers are individually represented, and demand is predicted from a full population of potential drivers. In the supply model, movement is simulated throughout the network on the basis of individual vehicles that follow their chosen routes toward their desired destination.

6.2.1 The within-day decision-making and day-to-day dynamics

The integration of demand and supply gives rise to the main premisses in DRACULA, namely the within–day decision–making process and the day–to–day dynamics. These are two important concepts that deserve special attention in modelling traffic systems with regard to users' behaviour.

The within-day formulation focuses on the travel choices made by individuals. These choices are made with regard to each specific journey to take place at a given time on a given day. All trip preferences, such as travel goals and purpose, travel needs, and other traveller parameters, such as perceptions, behavioural tendencies, and cognitive abilities that influence the decision-making process are reflective of the state of those variables at the instant the choice is being undertaken. The dynamic formulation, on the other hand, is concerned with modelling how the state of the network changes from one day to the other and evolves over time. In addition, the spatial knowledge of a driver is constantly evolving in response to travel made throughout the network. Figure 6.1 roughly depicts the DRACULA framework on the basis of the concepts mentioned above. Such a structure has been used as an attempt at improving the representation and simulation of the complexity and the uncertainty inherent in traffic domains.

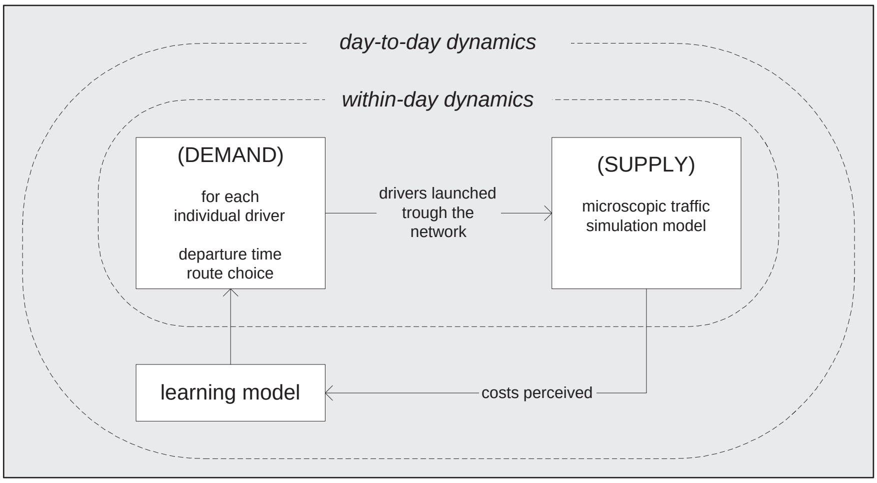

Figure 6.1: DRACULA: an example of demand–supply models.

6.2.2 The structure of DRACULA

The DRACULA basic structure, as presented in Figure 6.1 can be blown up into submodels, which are oriented to specific tasks within the simulation process. Figure 6.2 illustrates the basic DRACULA schema, and a detailed explanation of the simulation process can be encountered elsewhere (LIU; VAN VLIET; WATLING, 1995; LIU, 2001; LIU; VAN VLIET; WATLING, 1999; WATLING, 1995).

Roughly, travellers are individually represented in demand side where daily trip parameters are set up. Departure time and route choices are made on the basis of both past travel cost experiences and perceived knowledge of the network conditions. Con-

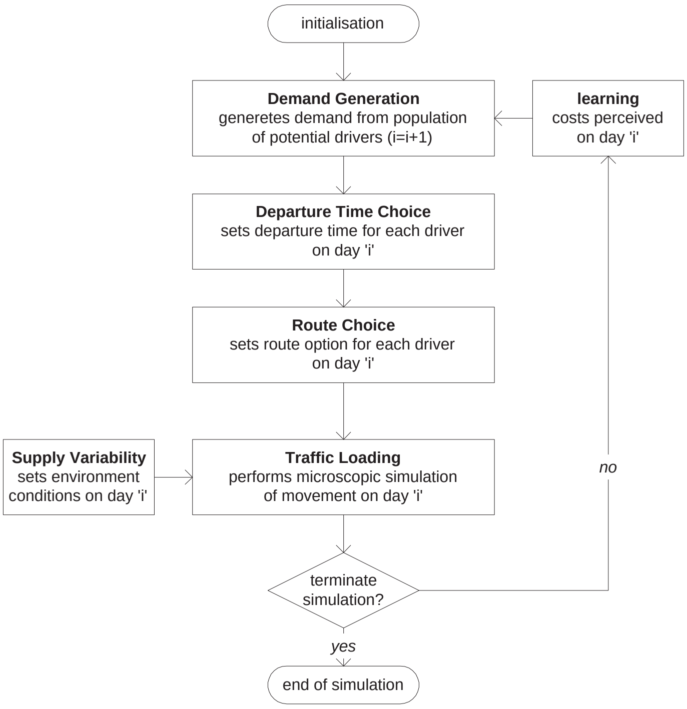

Figure 6.2: The basic structure of DRACULA framework.

trary to models based on a fixed matrix approach, the demand stage predicts the level of individual demand for a day k from a full population of potential drivers (LIU; VAN VLIET; WATLING, 1995). In the supply model, on the other hand, vehicles are individually moved throughout the network. They are launched onto the network and follow drivers' chosen routes according to both car-following and lane-changing rules. The resulting travel conditions for the subject day k and costs experienced by drivers are then re-entered into their individual knowledge basis. Such a dynamic knowledge will affect the demand model for the next period, that is day $k+1$. This process continues for a pre-specified number of days before simulation is terminated.

After each journey drivers make use of the experienced cost gathered from each link performed along the chosen route to update their information about the network conditions. This is the way individuals' spatial knowledge is maintained and it can be seen as the learning mechanism associated to each driver structure, which is discussed later on. A supply variability module ensures the stochastic nature of the environment, providing different perceptions of traffic conditions on each day over the simulation period. It is also important to mention that route choices are taken prior to the journey, which means that drivers will keep their chosen routes to their destinations and will not make en-route diversion in order to avoid either any incident or any accident, which may compromise the expected journey time. Hence, once drivers leave their origins they are not able to change their paths within the journey. Nonetheless, dynamic route choice and supplying drivers with route advice during the journey are interesting capabilities to be added to the DRACULA model.

6.2.3 The learning and decision-making processes

Historical costs are used to build a knowledge basis, which helps drivers at drawing an expectation for the network state so that they can use it to improve their decision-making process. There are a number of ways cost can be referred to, and in the commuter world it is basically regarded as travel time. In DRACULA, the travel time from each link used along the chosen path is recorded for future consideration. This can be done simply by keeping only the last experience, or by providing drivers with a memory capacity for computing the average travel time over a pre-specified number of days.

There are basically two ways to assign departure time choice in DRACULA, as it is implemented so far (LIU; VAN VLIET; WATLING, 1999). Nevertheless, the system is open to deal with departure time issues in a number of different ways. The first and simplest method is to randomly assign a desired departure time for each potential driver in the modelled population. When drivers choose to travel on a certain day, they will depart at that desired departure time independently of route choice and any previous experiences. The departure time profile could be flat or distributed according to some user-specified model. The second method incorporated into DRACULA, and quite more complex, implements the choice in response to travellers' experience. This is the model detailed in Section 5.3.1.4. Departure time is chosen prior to every journey on the basis of both travellers' preferred arrival time and previous experiences. As seen before, drivers will try to adjust next departure every time arrival is beyond a scheduled delay. However, the model completely disregards early arrivals.

As for the route choice, one model currently implemented in DRACULA is based on the works reported in (MAHMASSANI; JAYAKRISHNAN, 1991; BEN-AKIVA; DE PALMA; KANAROGLOU, 1986), and assumes a 'bounded rational choice' (SIMON, 1956; MAHMASSANI; JAYAKRISHNAN, 1991) (this model was implemented as the habitual behaviour, as described in Section 5.3.1.4).

6.3 The MADAM Model

MADAM is an agent-based model aimed at representing variability in traffic demand by means of a population of driver agents. This approach relies on an extension to the DRACULA framework, as initially proposed in (ROSSETTI; BAMPI; LIU; VAN VLIET; CYBIS, 2000b), to support the microscopic simulation of the traffic environment.

Rather than building demand through centralised procedures that assign values to global parameters of data structures, MADAM allows for autonomous behaviour and decision-making on the basis of individual preferences. Demand results from a population of traveller agents with their own profile and behavioural model. In this work, each individual is implemented according to the architecture proposed in Section 5.2.1. A BDI model drives the cognitive behaviour used in choosing departure time and trip route, whereas movement is performed by means of a reactive structure. This model is specially oriented to representing the stochastic nature of travel demand, and is assumed to coexist within a multi-agent environment. Hence, social ability and multi-agent interaction are two major features of the model.

The original schema of DRACULA (see Figure 6.2) can be adapted into the schema presented in Figure 6.3 to support the MADAM approach.

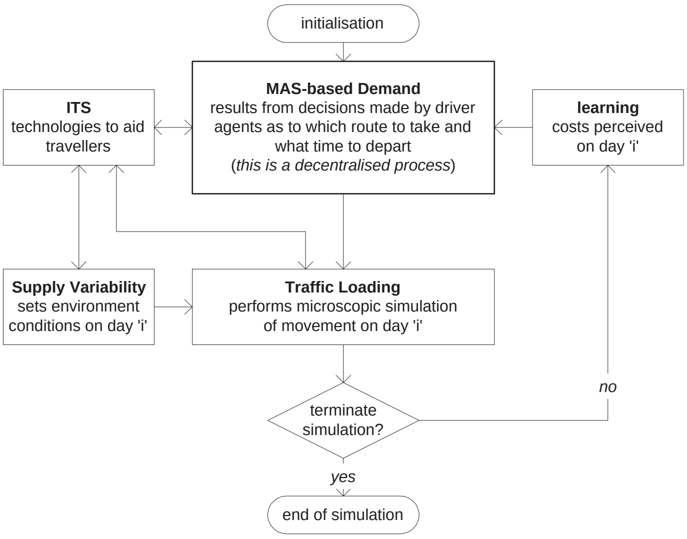
initialisation
MAS-based Demand
results from decisions made by driver agents as to which route to take and what time to depart
(*this is a decentralised process*)
ITS
technologies to aid travellers
learning
costs perceived on day 'i'
Supply Variability
sets environment conditions on day 'i'
Traffic Loading
performs microscopic simulation of movement on day 'i'
terminate simulation?
no
yes
end of simulation

Figure 6.3: The extended DRACULA schema.

At each simulation iteration, demand is given rise as BDI drivers make their choices on the basis of their individual trip preferences. Departure time and route are assigned on individual basis by means of a cognitive procedure carried out by each member of the population. At traffic loading, drivers are launched throughout the network to perform their trips along chosen paths starting at desired departure times. A reactive behaviour drives the movement on the basis of car-following and lane-changing predefined rules. As individuals execute their journeys, cost and other information from the environment are perceived through sensing and are used to enrich driver's internal model of the world (represented in BDI agents by means of a set of base beliefs). Contrary to drivers, that have restrict access to the whole world, the multi-agent system allows for the presence of *ITS agents* as well. The ITS agent is the abstraction used to represent all technologies available within Intelligent Transportation Systems, such as ATIS. Depending on its purpose, an ITS agent may possess a considerably broad model of the world. This can therefore be used to anticipate updated information on the system state to aid drivers' decision-making. Conceptually, such an interaction could take place both prior and during the journey.

6.4 MADAM+DRACULA: the simulation framework

In order to simulate and test the approach proposed in this thesis, MADAM was integrated into the DRACULA framework. It replaces the former demand side as depicted in

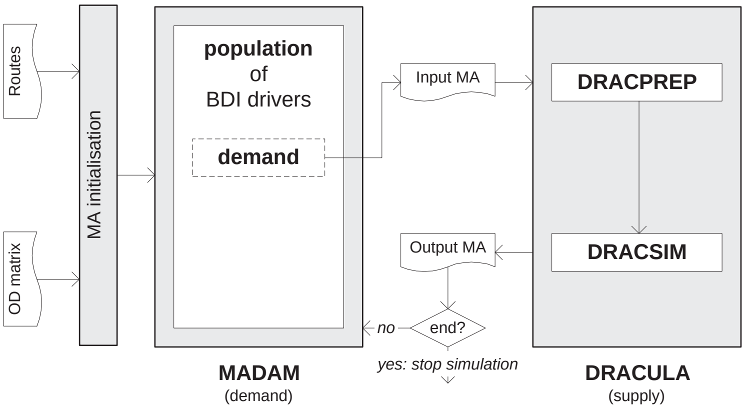

Figure 6.4: MADAM+DRACULA simulation framework.

the conceptual structure of Figure 6.4.

Both *demand* and *supply* are originally implemented as stand-alone facilities that communicate to one another via file exchange. The *MA Initialisation* module synthesises the population for the experiment from an *OD matrix* and different alternatives of paths are assigned to each driver from a list of possible *Routes* for each origin and destination pair. The initial set of base beliefs is generated in the format of the JAM BDI kernel for each individual in the population. The *Input MA* file gathers drivers' decisions on route and departure time, so that they can be launched onto the network to perform their journeys at the departure time selected. Such decisions may have been influenced by the information provided by an *atis agent*, that keeps a global model of the traffic environment condition. On the other hand, the *Output MA* file returns the travel costs experienced by each driver in terms of realised travel time. The perceptions gathered during the course of the journey simulated in DRACULA are used in the updating of the base belief sets. On the following day, drivers will rely on their updated beliefs to make decisions all over again. This process is repeated for a specified number of days before the simulation is terminated.

The simulation framework is implemented in C/C++ programming languages, following the same development strategy as adopted for DRACULA. However, a different strategy was adopted in the development of the cognitive layer of the driver agent, which is implemented in the Java language. That was necessary as a means to base the JAM BDI kernel that drives the cognitive abilities of motorists.

6.5 Experiments and Result Analysis

Some experiments were carried out in order to demonstrate the methodological approach proposed in this thesis. A small network within the Otley urban area was selected for this purpose. The network topology has 54 links (roads segments), which are connected through 14 junctions. Most road junctions follow a priority regime whereas two of which are controlled by means of traffic signals. The network description file is presented in Appendix B. A snapshot of the network being simulated in the DRACULA

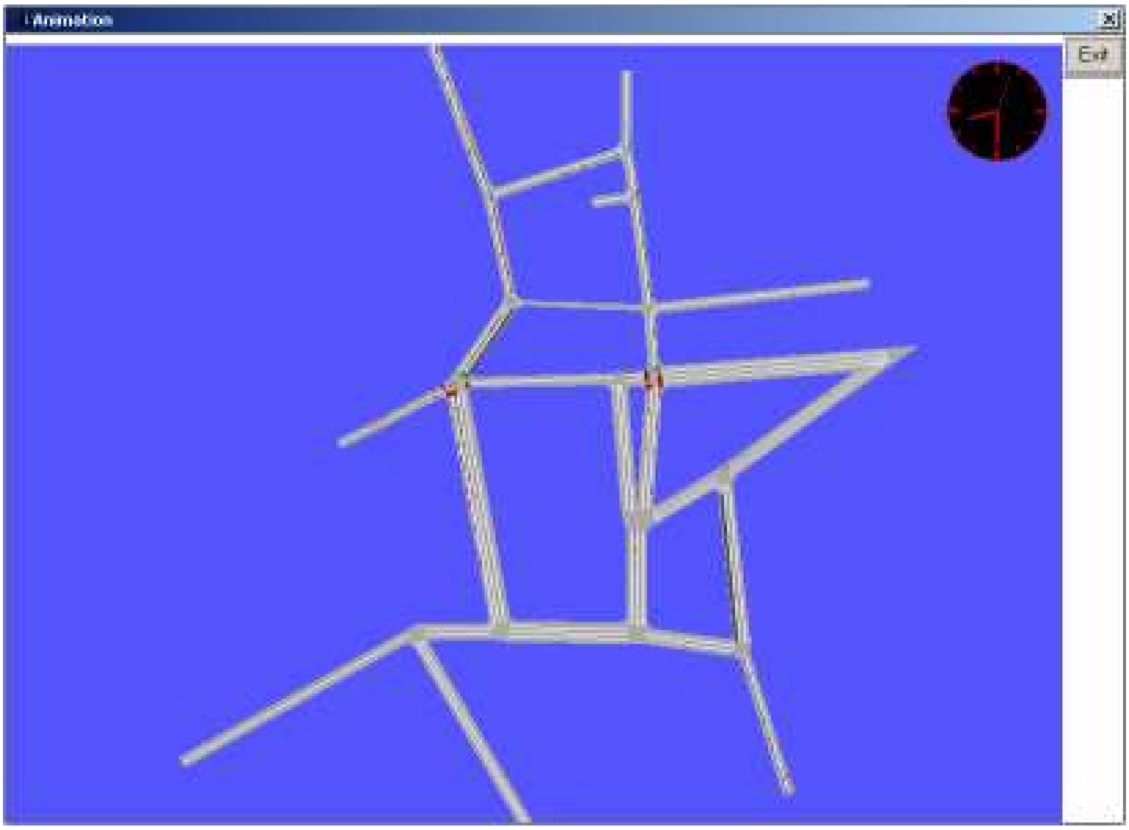

Figure 6.5: A snapshot of the Otley network.

environment is presented in Figure 6.5, whereas a schematic representation is depicted in Figure 6.6 containing node (junctions and zones) identification numbers.

Demand for travel results from the decision-making process performed by BDI driver agents in the population of commuters. The population is derived from the total number of trips described in an OD matrix, considering each trip as a driver. For this work, the total number of drivers from the selected OD matrix is 2323. Trips distributions among OD pairs within the network are detailed in Appendix A where the OD matrix description file is presented. Due to the lack of full integration of both demand and supply models within MADAM+DRACULA framework, only the first and second scenarios are considered in the simulation experiments. Some different configurations of the population by means of varying parameters and compositions of driver personalities were used in some of these experiments.

The main performance measure used in the discussion of experiments is the travel time of selected origin–destination pairs, on both individual and aggregate basis. Also, the quality of individual trips is evaluated for different behaviours in terms of both desired and actual arrival times. Each experiment consists of 101 runs of demand–supply iterations corresponding to *day* 0 through *day* 100. The morning peak period starting at hour 8 is considered for each day. In addition, a desired arrival time is assigned to each driver of the population on the basis of a uniform distribution. Thus, individuals are expected to arrive at their destinations by the stipulated arrival time, which is between minutes 45 and

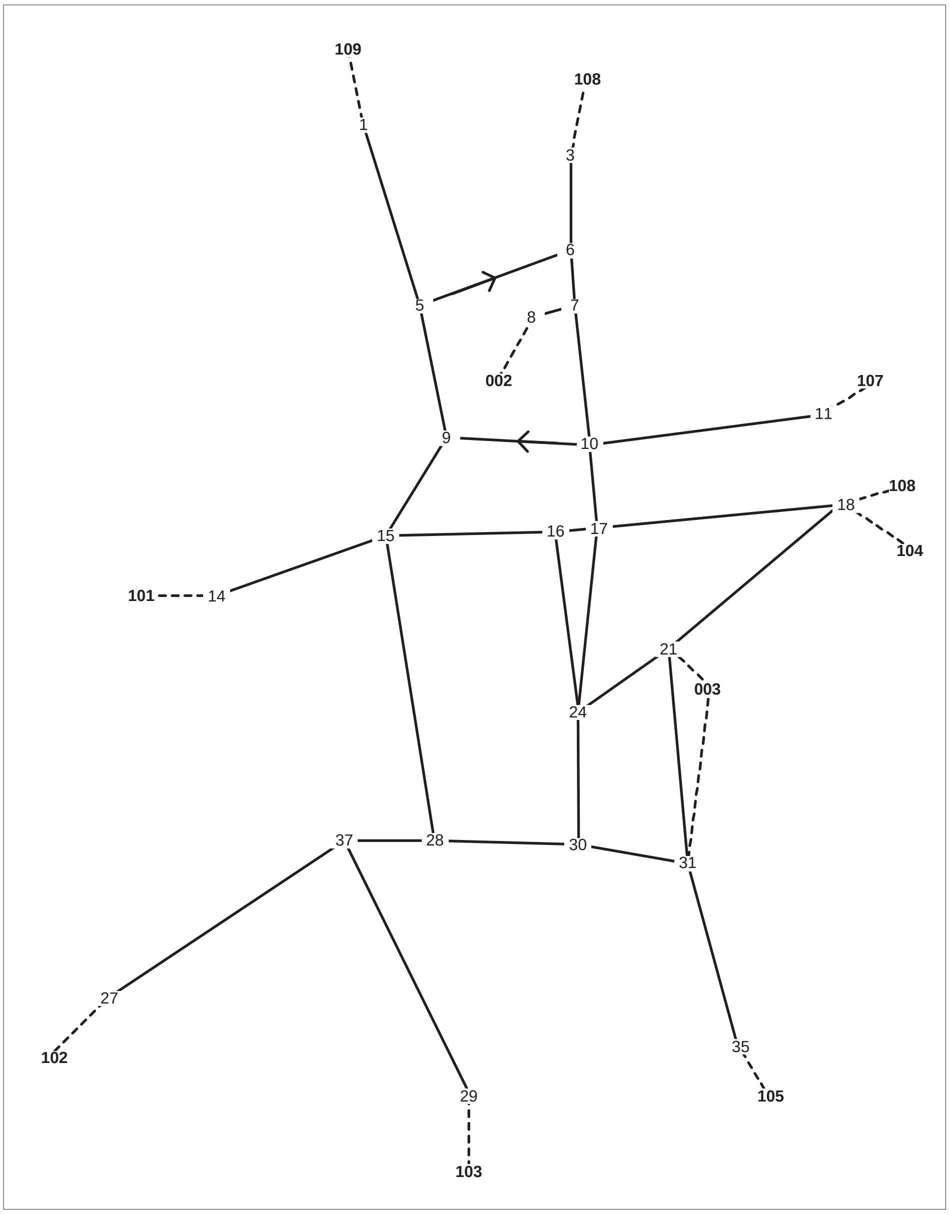

Figure 6.6: Otley network schematic representation.

60 after the starting hour. Such a short arrival period is suggested in order to induce the generation of more congestion throughout the network.

6.5.1 First Scenario

The experiments in the first scenario are focused on the observation of the behaviours suggested in Section 5.3.1. The first set of experiments considers that the population is fully composed of drivers exhibiting the same personality. One single agent is selected out from each of the random, choosy, conservative, and habitual populations to illustrate how personalities evolve over time as the driver makes its decisions and performs the journey. Despite trips are distributed all over a total of eleven OD pairs within the traffic network, only trips from origin 109 to destination 105 are initially considered. In this case, three possible routes are regarded for selection by the drivers, as depicted in Figure 6.7.

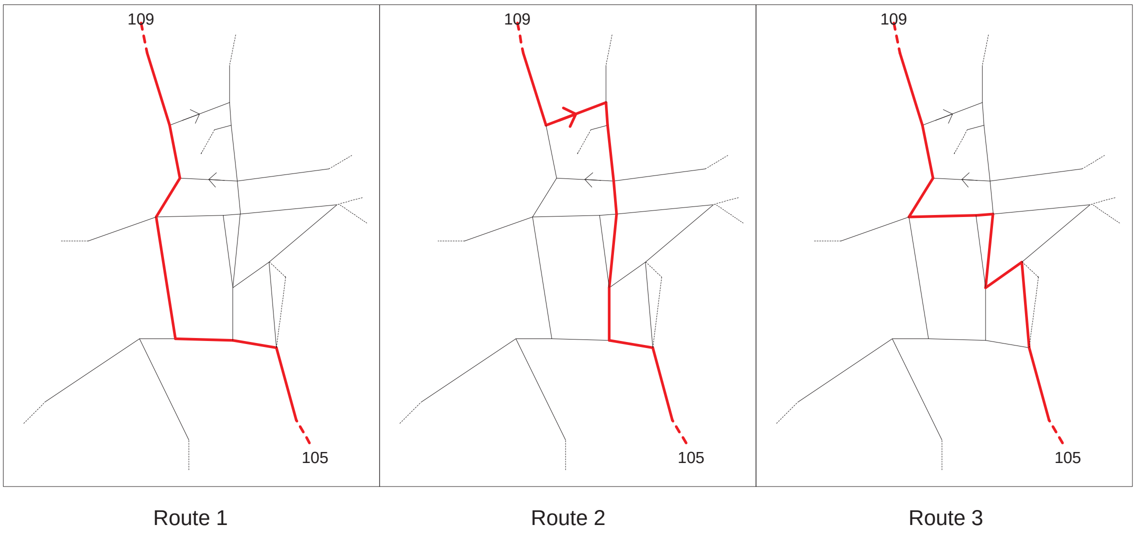

Figure 6.7: Route options from origin 109 to destination 105.

Random drivers do not care on route selection and may choose any of those path the driver is familiar with. Departure time, in turn, is adjusted on the basis of the expected travel time for the chosen itinerary. The random behaviour of a driver is depicted in the graphs of Figure 6.8. It is possible to notice from the graphs that the agent keeps no relation between route option and its expected cost, which is the total travel time realised the last time the route was selected. Also, owing a very strict desired arrival time as no tolerance for being either earlier or later is admitted, the travel time experienced on each day fluctuates considerably.

The same observation is carried out for the remaining basic behaviours, namely the choosy, the conservative, and the habitual ones. The behaviour of a choosy driver is illustrated in graphs of Figure 6.9. Contrary to the random personality previously discussed, the route selection strongly depends on the expected cost of each path. However, the cost is similarly given in terms of the travel time realised the last time the route was used. For the subject OD pair, the agent is able to opt among three alternative itineraries and keeps

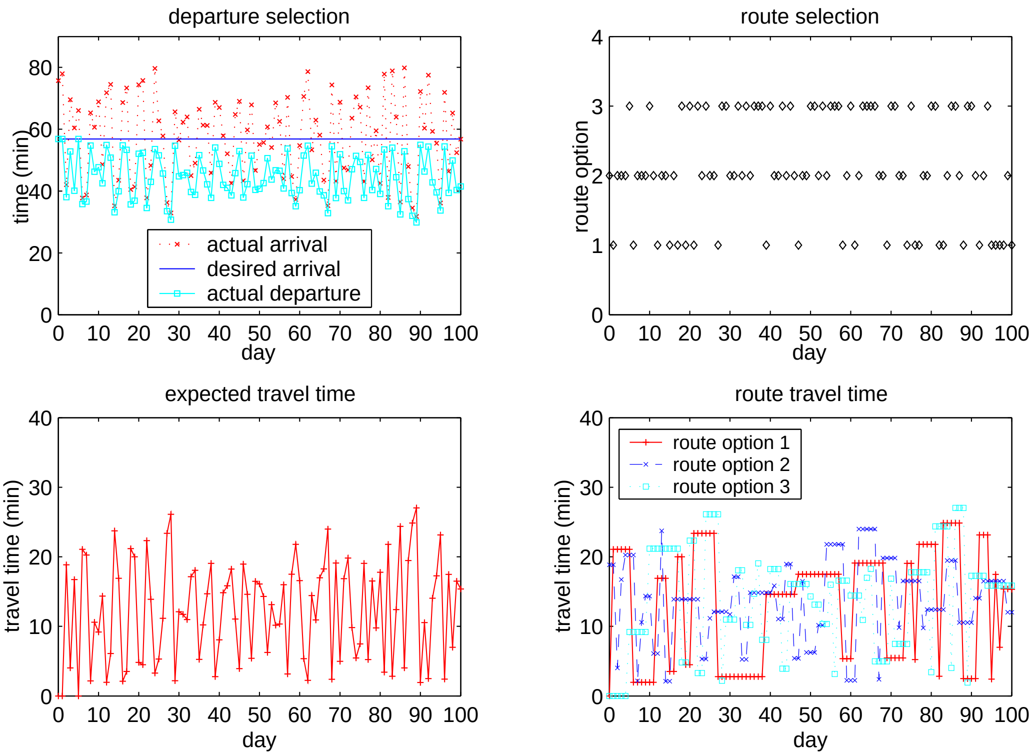

Figure 6.8: The random behaviour of a driver.

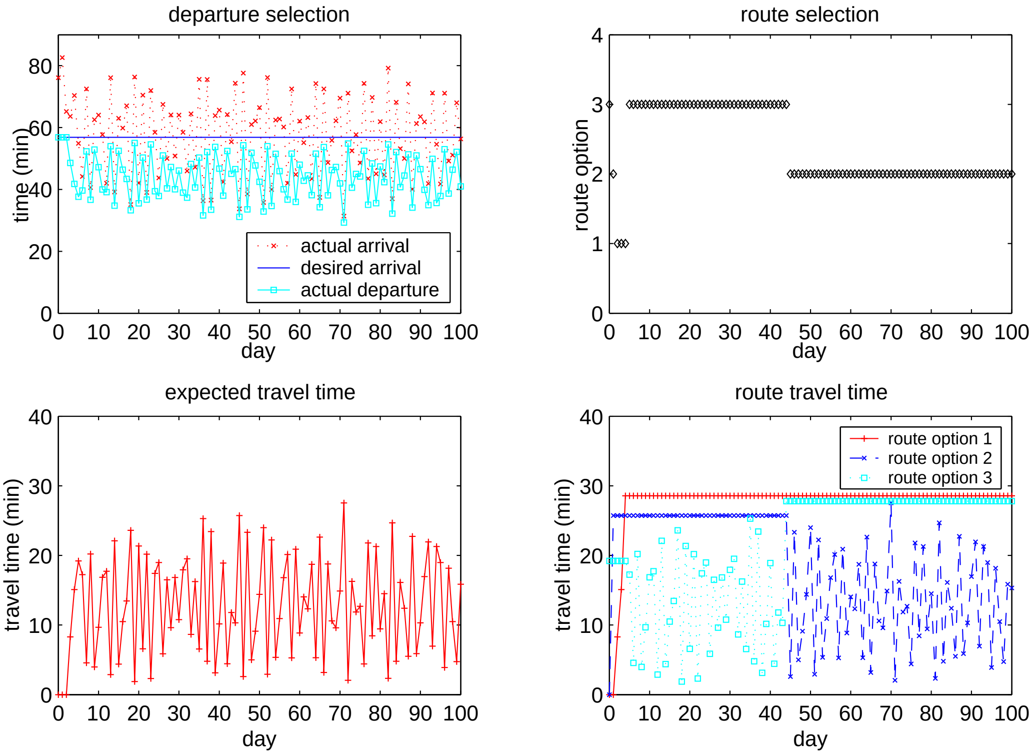

Figure 6.9: The choosy behaviour of a driver.

its choice until it finds out a better alternative. In this way, the expected travel time of a route remains the same and is updated only if it is selected on following days. Also, the number of route switches seems to be quite lower than it is for random drivers. This is basically due to the explicit use of selecting strategies.

Yet, as in the random behaviour, there is a considerable fluctuation in the agent's actual arrival time owing to the same reason. The strict desired arrival time forces the driver to continuously adjust its departure time as an attempt to meet its trip objectives. In the case of being any later, the agent anticipates its previous departure choice accordingly, and may experience a very short journey that results in an early arrival. In turn, the agent sets its departure for a later time on the next day. This behaviour seems to repeat indefinitely, which does not seem to correspond to the reality.

As for the conservative personality, travellers keep the same path option for the total period simulated, which means in practical terms that the expected cost for each journey is the travel time realised on the day before. Similarly to what is observed from random and choosy behaviours, conservative commuters also seem to fail in achieving destination at desired arrival time, as presented in Figure 6.10. Despite of fixing the route choice always at the same option, adjusting departure according to the full arrival delay demonstrates to be an efficient means to accomplish trip objectives.

The habitual driver, which is currently implement in DRACULA, relies on a more flexible approach both to departure time and to route choices. Graphs in Figure 6.11 are used to illustrate how the habitual behaviour evolves over time. The parameters ε, η, and τ are set to 0.2, 0.2, and 1 *min*, respectively. These values are selected as suggested in (BEN-AKIVA; DE PALMA; KANAROGLOU, 1986) and in (MAHMASSANI; JAYAKRISHNAN, 1991). Owing to the lateness tolerance of the model, drivers can experience a smoother arrival after few days from the start of the simulation. The departure choice remains the same unless new delay beyond what is tolerable by the driver is perceived. This means that stabilising arrival much earlier results in keeping the same departure choice. However, featuring agents with such a unlimited earliness tolerance may not be exactly the case for real commuters, specially during morning journeys. Switching routes is also constrained by an improvement factor, which means drivers do not make other option unless the gain for taking a better itinerary is considerably advantageous. Such a sort of behaviour with regard to path selection seems to be more prudent than the fastidious choosy personality.

Aggregate travel times are observed for trips from origin 109 to destination 105, as previously suggested, to represent how individual behaviours can effect the system overall performance (see Figure 6.12). The values for mean (μ) and standard deviation (σ) relative to the averaged travel time observed for the subject OD pair are listed in Table 6.1. A population estimation factor equals to 1.0 is used to emulate the population of BDI agents from the flow distribution described in the OD matrix. This way the total flow is mapped to the total number of drivers in the population on the basis of the $1:1$ rate. So,

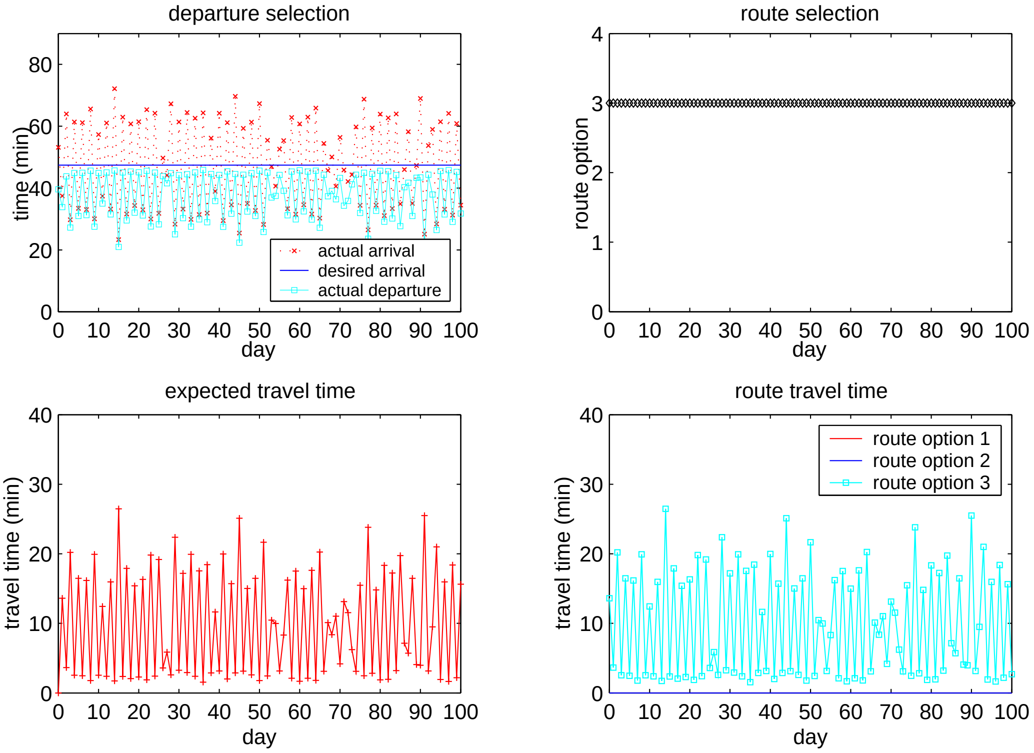

Figure 6.10: The conservative behaviour of a driver.

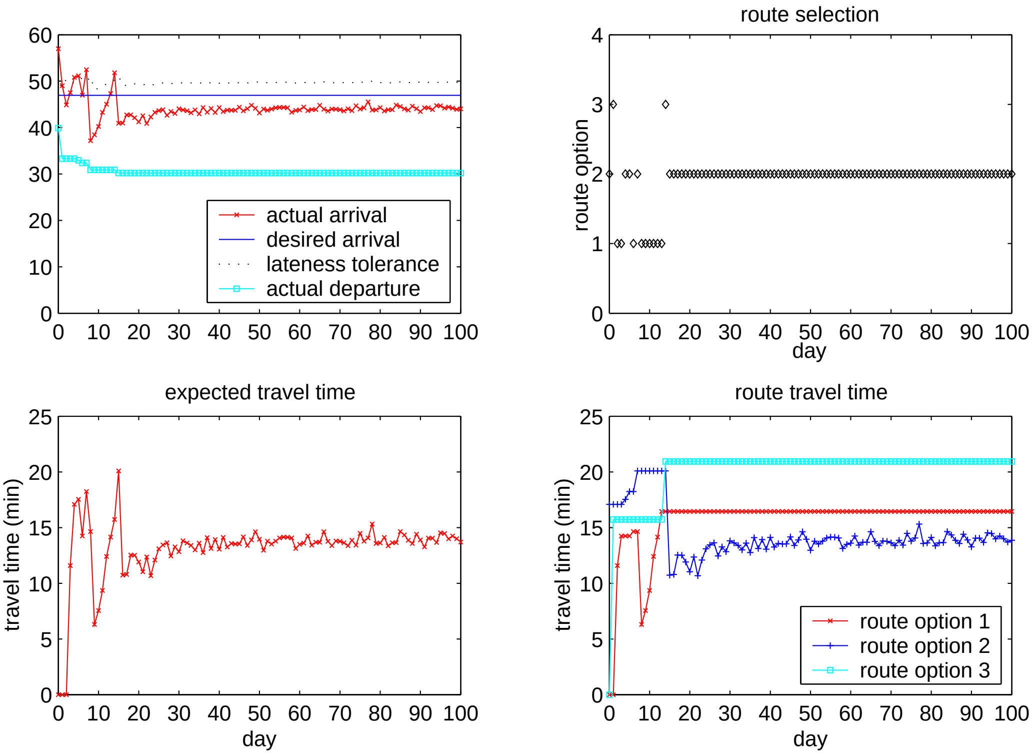

Figure 6.11: The habitual behaviour of a driver.

a total of 285 drivers perform their journeys from zone 109 to zone 105.

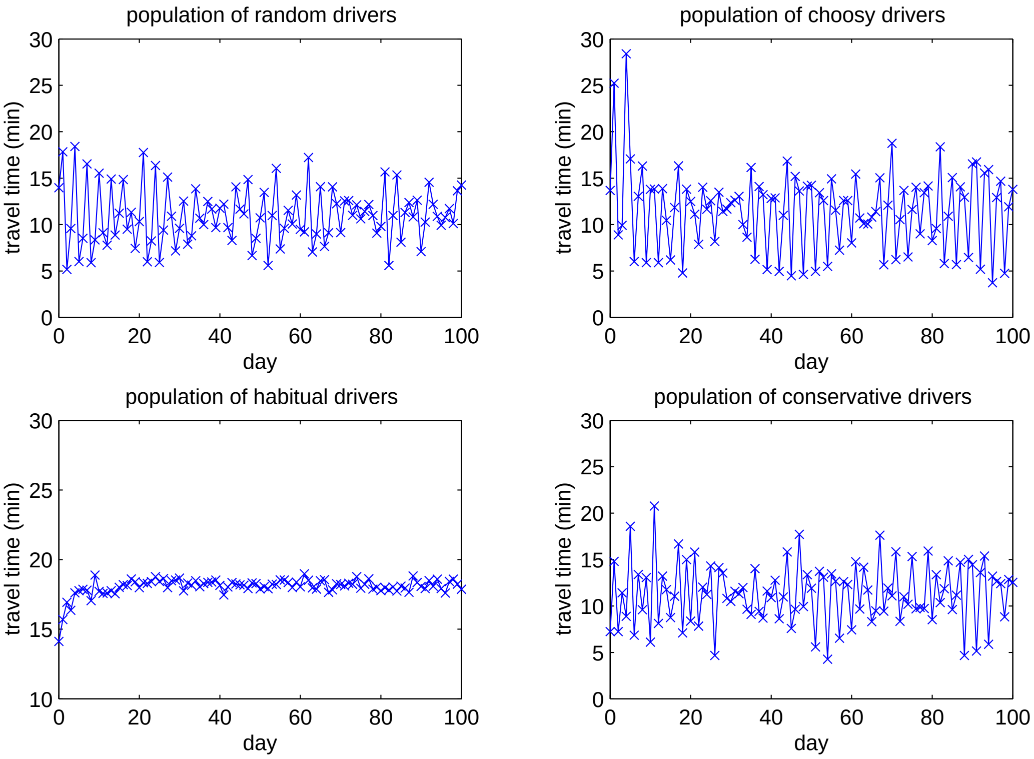

Figure 6.12: Average travel for homogenous populations.

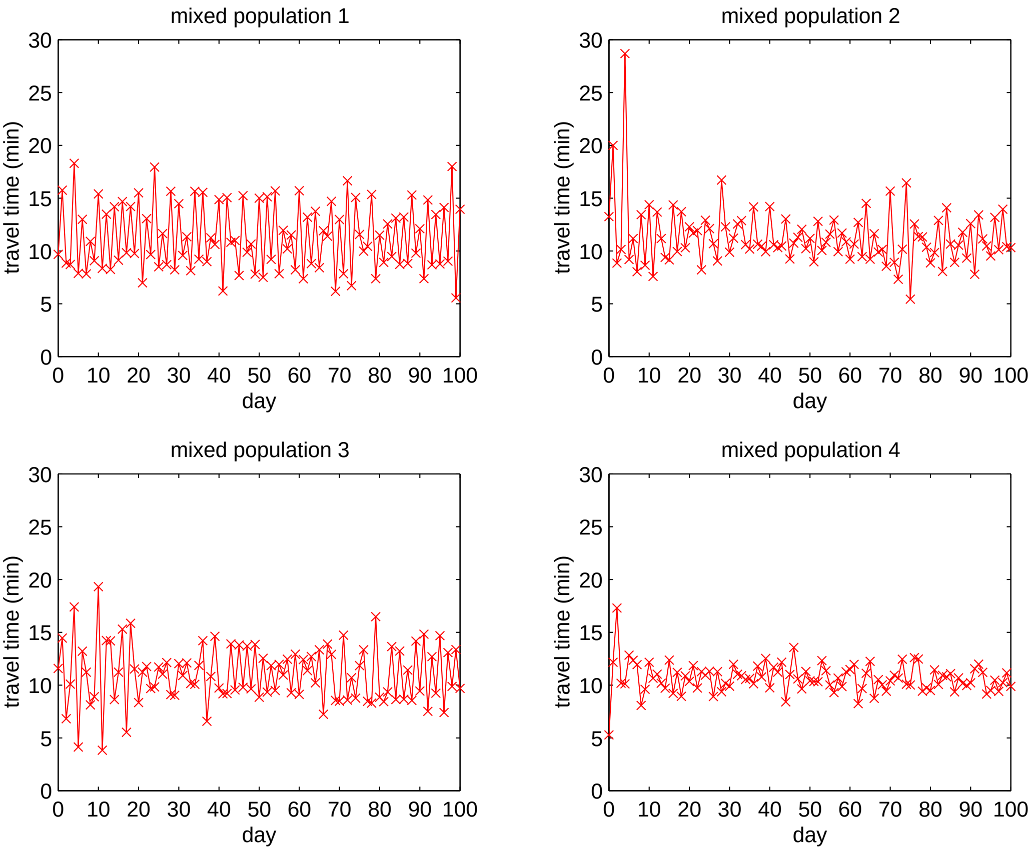

Figure 6.13: Average travel for mixed populations.

Populations fully formed of random, choosy, and conservative drivers present similar values for μ and σ, as presented in Table 6.1. Despite of presenting some criterion for route and departure time selection, the inflexible arrival of individuals makes aggregate travel time quite fluctuating with respect to the habitual behaviour (presenting the lowest value for σ). Another interesting observation is the high value for the mean average travel time of the habitual population ($\approx 18\,min$) with respect to the other behaviours. Owing to its very flexible arrival time, habitual drivers tend to keep their travel preferences even in the case of experiencing longer journeys. Then, it is quite possible for the system to stabilise the average travel time at higher levels provided that drivers can arrive at least before a certain scheduled delay. It is solely an effect produced by motorists that try to minimise their individual notion of cost. In the case of habitual agents, they only need to arrive before the tolerable lateness. Such a flexibility is not verified for random, choosy, and conservative personalities. As no tolerance is considered then, neither to lateness nor to earliness, departure time is set as the result of the agent's expectation to take the exact travel time as to arrive at destination on schedule.

Table 6.1: Populations formed solely by drivers of same personality.

Personalities	μ	σ
Random	11.0020	3.0288
Choosy	11.5581	4.3239
Conservative	11.3508	3.2675
Habitual	18.0609	0.6176

Undoubtedly the populations of travellers in any urban area is rather of a very heterogenous nature. Thus, considering that all drivers behave in the same way is very unlikely to correspond with reality. In this sense, another set of experiments is suggested and different populations are built up by means of mixing the number of drivers of the same behavioral stereotype, as suggested in Table 6.2. The same OD pair is selected for the observations and average travel times are depicted in the graphs of Figure 6.13. The number of random drivers is kept constant in a very low rate (10% of population) for all compositions. As mentioned in Section 5.3.1 such a behaviour may represent commuters that need to use different paths. Some times people plan to take a different itinerary, for example, to drop kids at school or supply vehicle with petrol. However, it is intuitively very unlikely that most commuting users will behave in this way. The remaining part of the populations is composed of fractions of choosy, conservative, and habitual agents. These behaviours are equitably distributed in Population 1, whereas a greater rate (70%) is considered for each personality in populations 2, 3, and 4. This assumption is suggested as a means to observe how prevailing behaviours can effect the overall system performance.

From the graphs of Figure 6.13 and from the values for μ and σ (see Table 6.2), it is possible to observe that all population compositions present fluctuations in the average travel time. Populations 1, 2, and 3 are still affected by the non-flexible arrival time as discussed above. Nonetheless, the prevailing habitual behaviour of Population 4 seems to

Table 6.2: Different compositions for mixed populations.

Populations	Compositions				μ	σ
	random	choosy	conservative	habitual		
Population 1	10%	30%	30%	30%	11.3096	3.1331
Population 2	10%	70%	10%	10%	11.3084	2.8119
Population 3	10%	10%	70%	10%	11.0243	2.7316
Population 4	10%	10%	10%	70%	10.6749	1.3884

smoothen the average travel time for the subject OD pair. Also, as possibly a side-effect of the other personalities in the same population, the mean travel time drops with respect to the first experiment considering the population formed solely by habitual commuters. However, it is very important to bear in mind the very flexible earliness tolerance of the habitual personality. Stability could be observed at any average travel time level provided drivers reached destination within the lateness tolerance threshold. In spite of that, this may suggest non-flexible behaviours can condition at which level such an stable state is to be settled.

As discussed above, the habitual driver seems to be quite flexible with regard to earliness. Also lateness tolerance increases for longer journeys as it is assumed to be relative to travel time. However, commuters tend to have rather strict arrival constraints and arriving much earlier may be disregarded for morning trips. Two extensions to the habitual personality are then suggested, namely the habitual driver with a relative lateness-earliness tolerance window and the one presenting an absolute lateness-earliness tolerance. The former only extends the concept of relative tolerance to consider an earliness threshold with respect to the travel time realised. The latter is less flexible in the sense lateness and earliness thresholds are considered to be constant in spite of how long the journey may take. To illustrate how these extended behaviours possibly evolve over time, different configurations for the tolerance window are suggested. In Table 6.3 lateness threshold is fixed in 20% of travel time, whereas different values are assigned to earliness tolerance factor. The behaviour of a single instance of a habitual driver presenting a relative lateness-earliness tolerance to desired arrival time is depicted in Figure 6.14. The fluctuation of the agent's arrival time is considerably high, even for the $[20\%, 100\%]$ relative window size. In this specific behaviour, when the driver tries to adjust its departure to avoid lateness, for example by means of departing much earlier, it may realise a considerably short journey yielding a very restrictive tolerance. As thresholds are dynamic, even if the driver experiences longer journeys some times, it seems to be very difficult to reach a steady state.

A similar experiment was carried out for the latter extension suggested, which is the habitual driver with absolute lateness-earliness tolerance window. The different configurations for the absolute window are presented in Table 6.4; considering most commuters are constrained by strict arrival times, lateness tolerance is fixed in $5min$ whereas earliness

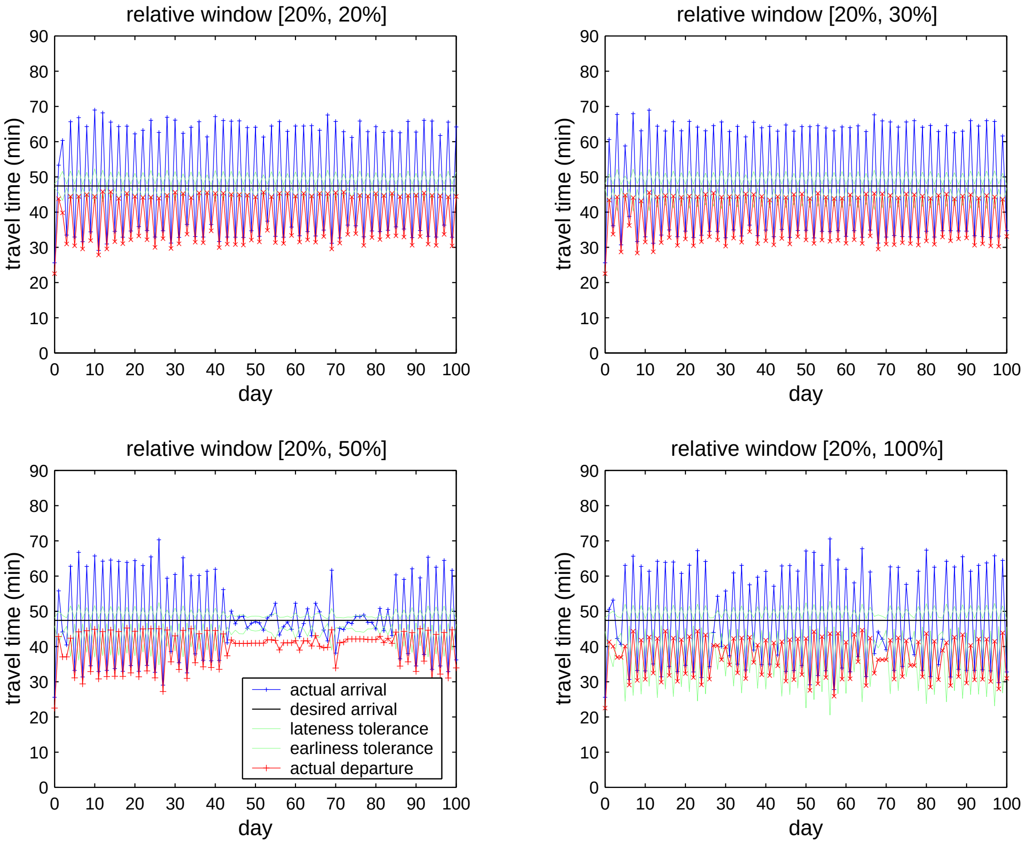

Figure 6.14: Habitual driver with relative lateness-earliness tolerance.

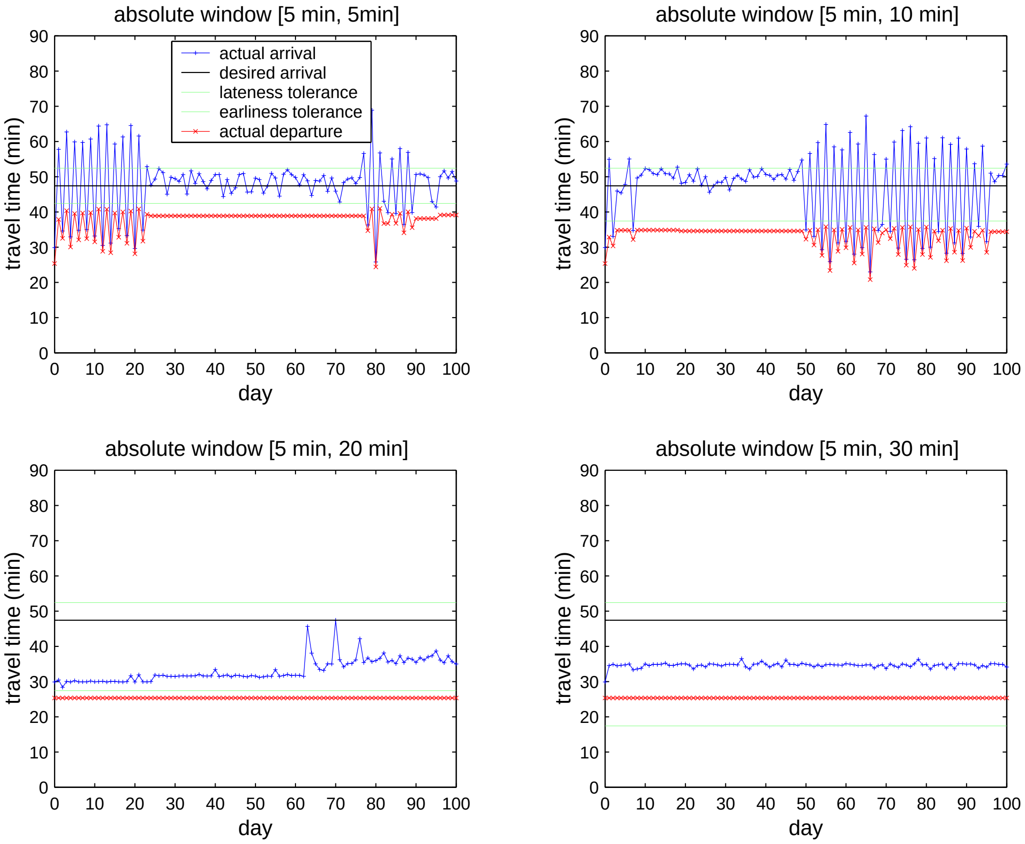

Figure 6.15: Habitual driver with absolute lateness-earliness tolerance.

is different for each population.

The graphs in Figure 6.15 are used to illustrate how this behaviour evolves over time. In this case, as the window boundaries are static all over the period simulated, it is easier for the driver to meet its lateness-earliness thresholds after a number of iterations and keeping this state for a longer period of time. The wider the distance between the upper and lower boundaries, the more tolerable the agent will be. In the same way the original habitual behaviour can settle in a steady state owing its flexible nature, can the driver with an wide absolute tolerance window. Thus, it seems to be impracticable to predict at which level a population with such characteristics may stabilise. It may be at any level, provided travellers can meet the window thresholds.

The average travel times for the populations presented in Table 6.3 and Table 6.4 are depicted in Figure 6.16. This aggregate assessment is made with respect to trips from zone 109 to zone 105, as well. The presence of an earliness relative tolerance makes the average travel time quite more fluctuating with regard to original specification for the habitual behaviour. On the other hand, absolute lateness and earliness tolerances tend to converge to a certain level at which average travel time starts to settle down. Intuitively, absolute tolerance windows seem to keep closer relation to the reality of commuters than relative ones.

Table 6.3: Relative lateness-earliness tolerance windows.

Populations	Lateness Tolerance	Earliness Tolerance
	ε (relative to $\mathscr{T}\mathscr{T}$)	λ (relative to $\mathscr{T}\mathscr{T}$)
Population 1	0.20	0.20
Population 2	0.20	0.30
Population 3	0.20	0.50
Population 4	0.20	1.00

Table 6.4: Absolute lateness-earliness tolerance windows.

Populations	Lateness Tolerance	Earliness Tolerance
	ι (min)	ν (min)
Population 1	5	5
Population 2	5	10
Population 3	5	20
Population 4	5	30

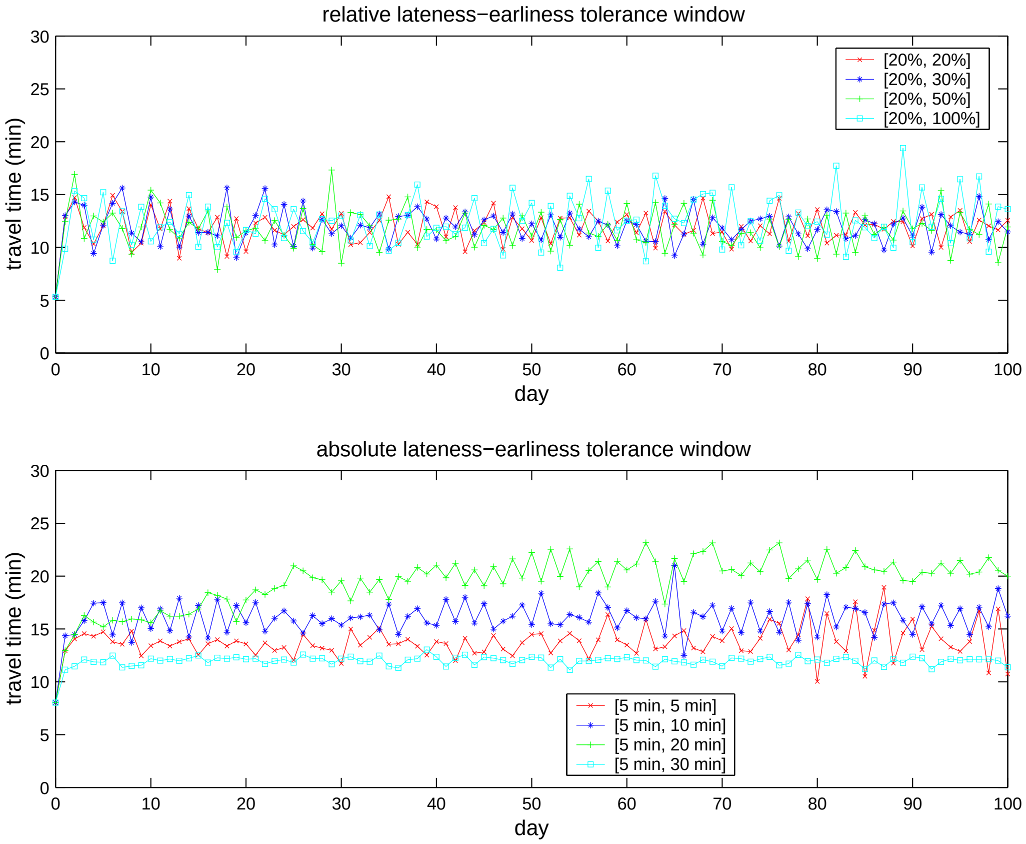

Figure 6.16: Average travel times for relative and absolute tolerance windows.

6.5.2 Second Scenario

Recalling from Section 5.3.4, drivers are allowed to receive information on the current state of traffic prior to starting daily journeys. This way commuters can consider using the content interpreted to improve the decision-making process. However, travellers are not supposed to interact with exogenous sources and contents are assumed to be perceived as a broadcast from mass media, for instance. In addition to $(109, 105)$ OD pair, trips from zone 105 to zone 104 and trips from zone 101 to zone 002 are observed as well. Both $(105, 104)$ and $(101, 002)$ OD pairs can be performed through two possible itineraries as presented in Figure 6.17 and Figure 6.18, respectively. Demand is built out from homogeneous population of habitual drivers, which means all travellers in the population exhibit the same personality.

Different populations are set out in Table 6.5 and simulated in order to illustrate the

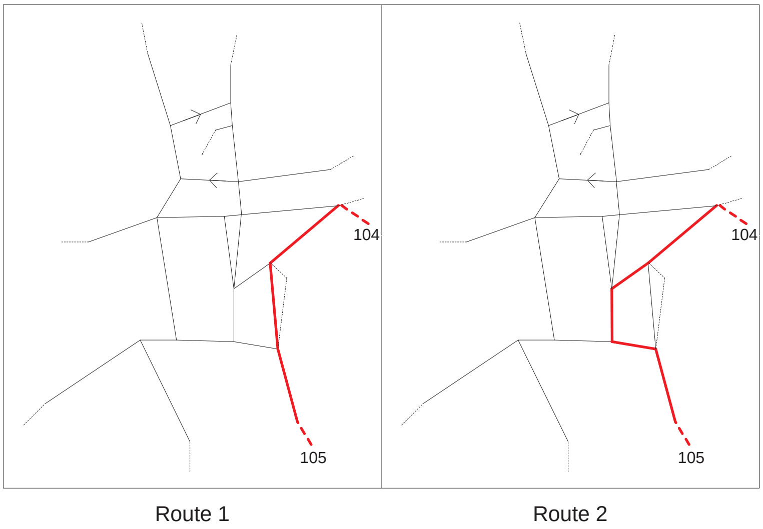

Figure 6.17: Route options from origin 105 to destination 104.

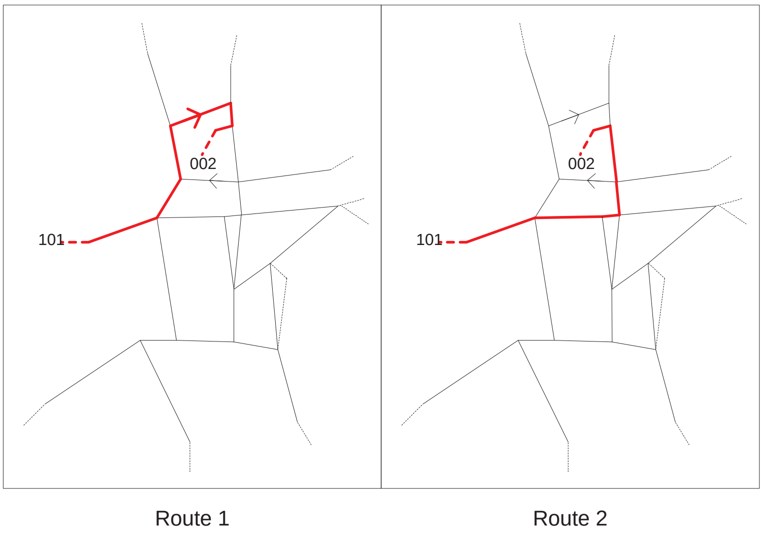

Figure 6.18: Route options from origin 101 to destination 002.

potentialities of the approach presented in this thesis to handle such a scenario. Populations are configured in terms of the total number of drivers and the fractions of pre-trip information users. Incidents were artificially produced in two links of the network affecting the current supply conditions for the recurrent demand. This is yielded by way of suppressing one traffic lane from both link $(9,15)$ and link $(31,21)$. The pairs represent upstream and downstream nodes in this order, which gives the direction of the suppressed lanes. In the supply model of DRACULA, incidents can be easily represented by means of defining a reserved lane and the purpose for such a reservation (see Appendix 7.5 for further explanations). The incidents have direct effects on both trips from zone 109 to zone 105 and trips from zone 105 to zone 104. This is ensured as at least one path of each OD pair contains the links with lanes suppression. Trips from zone 101 to zone 002 are also expected to be indirectly affected by the incident generated in link $(9,15)$. This is assumed on the basis of the high flow induced in that link, which may produce queues that extrapolate to link (5,9) blocking right-turning maneuvers from link (9,5) to link (5,6).

An *atis agent* is responsible for broadcasting the information on the current state of links (9,15) and (31,21). The perspective of incidents on road segments is interpreted as possible congestion, and such information is posted in the environment on day 50. Thus, the base beliefs of driver agents are updated through the perception of the clause

Table 6.5: Different compositions for populations with informed drivers.

Populations	Compositions			
	population factor	number of agents	trips for each OD pair	fraction of informed users
Population 1	0.5	1159	$(109,105)$ 142	0%
				25%
			$(105,104)$ 101	50%
				75%
			$(101,002)$ 6	100%
Population 2	0.8	1860	$(109,105)$ 228	0%
				25%
			$(105,104)$ 162	50%
				75%
			$(101,002)$ 9	100%
Population 3	1.0	2323	$(109,105)$ 285	0%
				25%
			$(105,104)$ 202	50%
				75%
			$(101,002)$ 12	100%
Population 4	1.2	2792	$(109,105)$ 342	0%
				25%
			$(105,104)$ 243	50%
				75%
			$(101,002)$ 14	100%

$linkSate(link, congested)$. This new entry in the beliefs set may then be accounted for during the route choice process. If the agent is a pre-trip information user and has the willingness of effectively using its content, it will definitely make an attempt at avoiding routes that include the congested link. The incidents last for the whole period simulated on each day, and the *atis agent* keep posting the messages until day 100.

The simulation results are given in terms of average travel times relative to the subject OD pairs and are presented in the graphs of Figures 6.19, Figure 6.20, and Figure 6.21. Each population as presented in Table 6.5 is identified by its corresponding population factor within the figures. It is possible to observe from the three set of graphs that the travel time for all the populations tends to settled down at different steady states after the incidents are introduced onto the network. However, it does not necessarily mean travel time will be brought to worse levels, as observed in Figure 6.20. This can be seen from the perspective of the Braess's paradox (BRAESS, 1969; MURCHLAND, 1970; SHEFFI, 1985). Such a seemingly counter-intuitive result can be explained by the fact that a motorist try to minimise her/his own notion of cost. Thus individual choices are carried out with no consideration of the effect of this action on other network users. And, according to Sheffi (1985), there would be no reason to expect the total travel time to be worse or better on certain circumstances.

Graphs also demonstrate that non-informed configurations of demand are very likely to produce the worst situations. On the other hand, different fractions of informed drivers may produce difference levels of stability. However, informing all drivers will not always produce the best result. Also, the experiments suggest that ideal penetration factors of information technologies may depend very much on demand configuration. This idea relies basically on the number of trips for each OD pair of the network and the resulting graphs of figures 6.19, 6.20, and 6.21.

Table 6.6: Summary of statistics for the (109,105) OD pair.

population factor	fraction of informed drivers	day 0–49		day 50–100	
		μ	σ	μ	σ
0.5	0%	3.9208	0.2118	7.7947	0.4047
	25%			7.4257	0.2797
	50%			7.3244	0.2871
	75%			6.7835	0.2597
	100%			7.1613	0.2579
0.8	0%	11.7635	0.6085	17.2555	0.7399
	25%			16.2070	0.5947
	50%			17.5906	0.6872
	75%			15.9085	0.2506
	100%			16.5513	0.5727
1.0	0%	15.9480	1.4969	22.0349	1.3894
	25%			23.0566	1.7372
	50%			18.9515	1.3005
	75%			20.8037	0.6656
	100%			21.4670	0.7124
1.2	0%	14.9382	1.2390	30.6920	1.1712
	25%			29.5433	1.7634
	50%			29.1563	1.9296
	75%			26.5099	2.0541
	100%			21.6514	0.8437

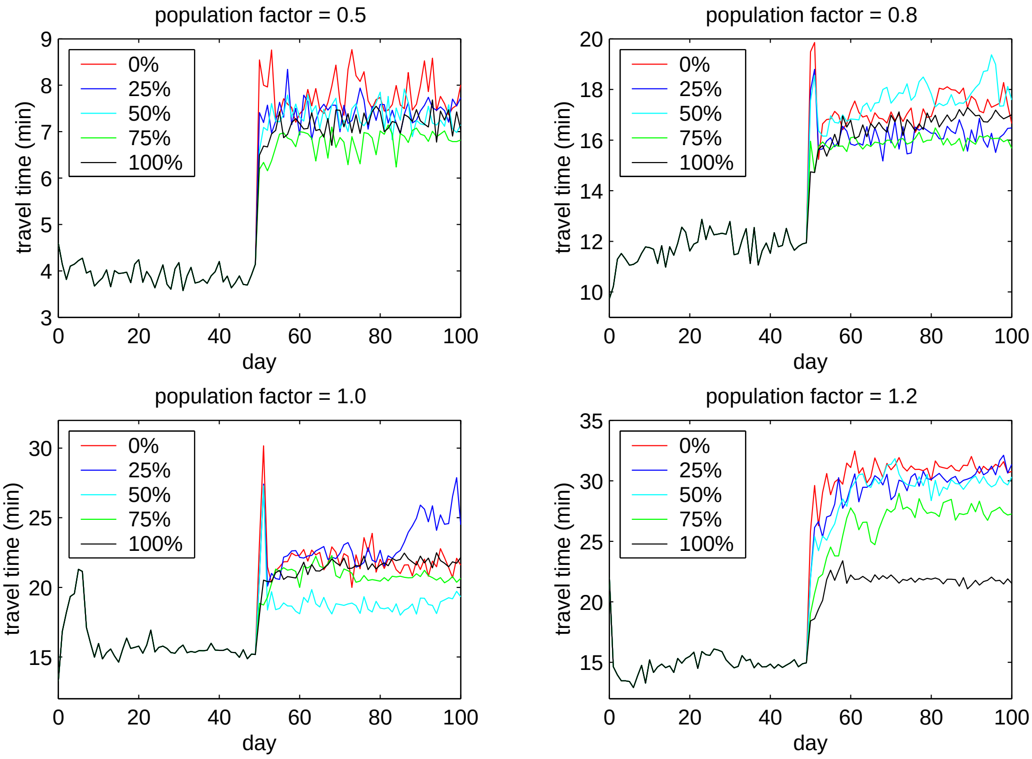

Figure 6.19: Average travel times relative to (109,105) OD pair.

Table 6.7: Summary of statistics for the (105,104) OD pair.

population factor	fraction of informed drivers	day 0–49		day 50–100	
		μ	σ	μ	σ
0.5	0%	5.7834	0.7476	5.1545	0.6021
	25%			4.6389	0.7725
	50%			3.1522	0.5461
	75%			1.9141	0.3065
	100%			1.5964	0.0691
0.8	0%	13.0971	1.0560	10.7561	0.9537
	25%			10.4312	0.8537
	50%			8.1870	0.7314
	75%			7.2055	0.4432
	100%			5.3103	0.2213
1.0	0%	17.7321	2.0563	12.3670	0.9086
	25%			11.2024	0.9258
	50%			11.0965	0.7030
	75%			8.9645	0.6181
	100%			7.7936	0.1683
1.2	0%	18.2241	2.1711	13.9007	0.9027
	25%			12.9853	0.7712
	50%			12.6078	1.1480
	75%			9.9759	0.4274
	100%			8.3903	0.1774

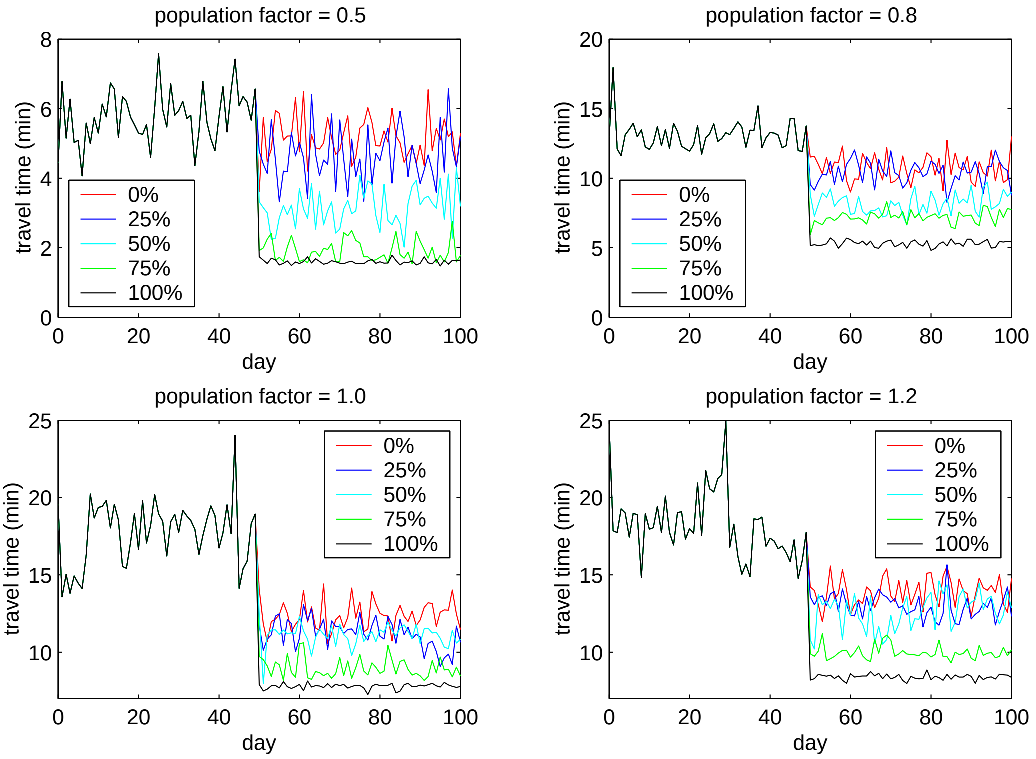

Figure 6.20: Average travel times relative to (105,104) OD pair.

Table 6.8: Summary of statistics for the (101,002) OD pair.

population factor	fraction of informed drivers	day 0–49		day 50–100	
		μ	σ	μ	σ
0.5	0%	4.4034	0.6327	4.2603	0.1168
	25%			4.2707	0.1012
	50%			4.2456	0.1149
	75%			4.2418	0.0843
	100%			4.2917	0.1435
0.8	0%	14.2028	2.5614	12.8585	0.2374
	25%			13.1715	0.3220
	50%			12.6788	0.2802
	75%			12.2342	0.1238
	100%			12.2416	0.1309
1.0	0%	22.2409	2.4364	27.2554	2.5552
	25%			22.6639	1.4758
	50%			22.8205	0.8266
	75%			20.1541	0.1735
	100%			19.8509	0.1410
1.2	0%	11.7650	3.7378	11.5785	0.8933
	25%			10.4111	0.3332
	50%			10.6383	0.4780
	75%			10.5401	0.4221
	100%			9.8394	0.1551

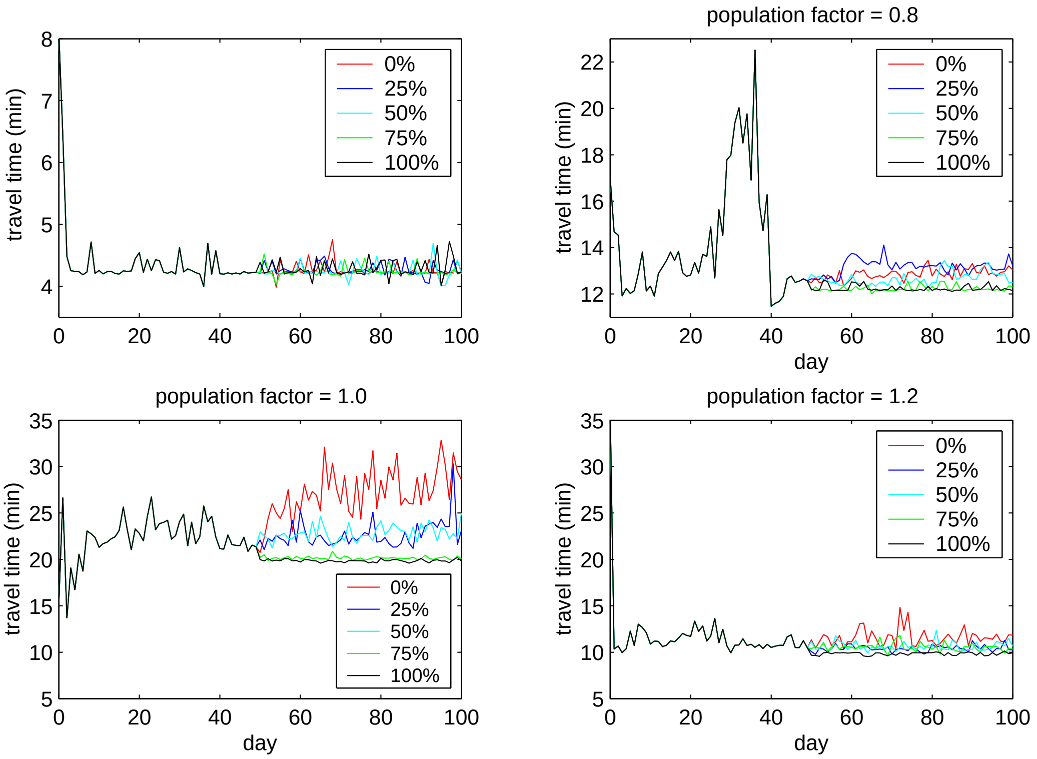

Figure 6.21: Average travel times relative to (101,002) OD pair.

6.5.3 Simulation performance

The simulation sets presented in this chapter were carried out in PC computers. Hardware and software characteristics are presented in Table 6.9.

Table 6.9: Simulation software and hardware environment.

HW and SW characteristics	
Processor	AMD Athlon at 1100 MHz
RAM capacity	256 Mb
Operating System	MS Windows 2000, Service Pack 2
Java Runtime Environment	Java 2 RE Standard Edition v.1.3.1_01
JAM Parser Version	$0.65+0.76i$

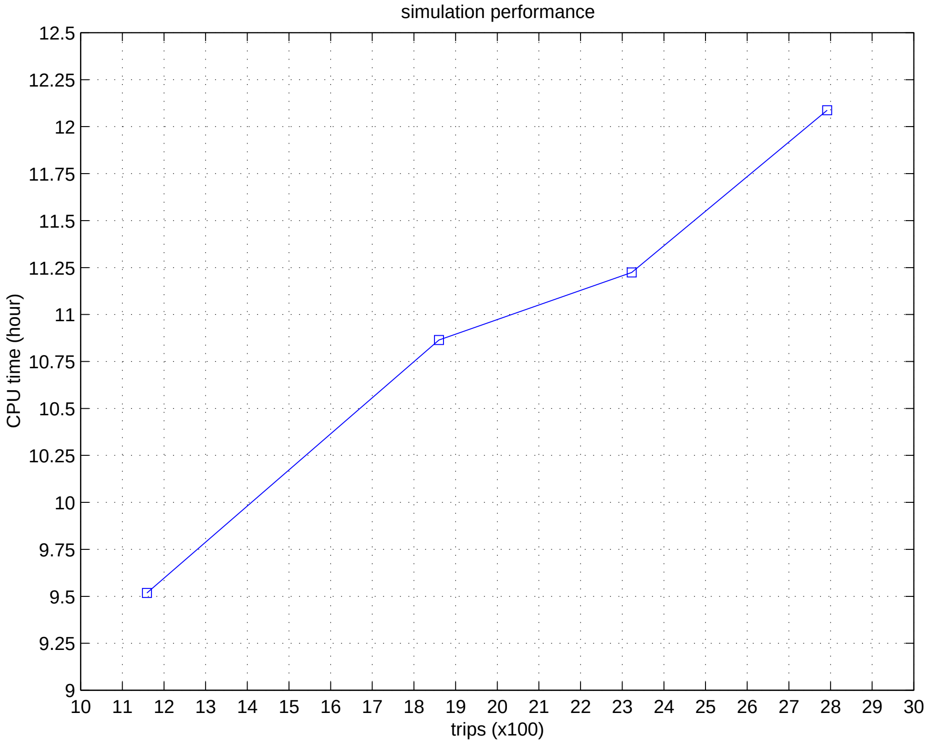

Figure 6.22: Simulation performance (number of agents versus CPU time/101 days).

The simulation performance for hardware and software is assessed in terms of CPU time used to run populations with different number of BDI driver agents for a total of 101 days (from day 0 to day 100). Results are presented in Figure 6.22. At a first glance, the performance of the agent-based approach may be found to be a bit disappointing, as the the necessary CPU time increases considerably for larger populations. Nonetheless, the adoption of other software development strategies using concurrent and parallel processing may improve the framework performance. A fully integrated environment is also a step toward this improvement, which can avoid the excessive number of file exchange between MADAM and DRACULA.

6.6 Summary

The flexible structure of AgentSpeak(L) used as an specification language gives rise to further improvements of legacy microscopic simulation frameworks, allowing engineers and practitioners to devise and implement different behavioural models. The MADAM+ DRACULA framework is an example of how cognitive agents can be practically applied to implement the humanlike reasoning commuter structures and their interactions with today's ITS technologies. This multi-agent-base methodology can be used to address the lack of suitable tools for simulating and assessing contemporary traffic scenarios.

The different experiments proposed and the simulation results are initially intended to demonstrate the methodology flexibility and its adequacy to cope with the traffic domain. However, scenarios are hypothetical and may not have so much correspondence with reality. In spite of that the intuitive syntax of AgentSpeak(L) was also used to specify and run the habitual personality, showing it is suitable to implement other models as well. Integrating a purpose-built interpreter for AgentSpeak(L) within the MADAM+DRACULA simulation framework can base the use of its syntax and operational semantics as a powerful API through which different behaviours can be implemented and tested.

As for the second scenario tested in this chapter, it was seen that users of exogenous information sources may rely on the information provided in order to improve the quality of their decisions and the outcomes of their journeys. So, this framework contributes to assessing the impact that exogenous information may have in the overall performance of the traffic system, the quality of the information provided, and the quality of decisions made by users. Also, it can give invaluable insights into variables dependence and great contribution in the calibration and validation of driver behaviour models.

7 CONCLUSIONS

7.1 Overview

The problems arising from the increasing demand in traffic systems mainly in urban areas have been challenging not only researchers but also the whole society all over the world. Alternative solutions are then necessary to tackle such problems. Physical modifications to increase capacity have revealed to get even more difficult as space lacks, and very expensive to implement, as well. Besides, such procedures often result in serious disruptions to the society and damages to the natural environment. Visual, atmospheric, and noise pollution are drawbacks to be avoided. Efforts to optimise control systems have resulted in considerable improvements in many situations. Apart from successful experiences, they alone are not sufficient to cope with the ever increasing demand in urban areas. The use of advanced communication and computer technologies has brought about the concept of Intelligent Transportation Systems. One of the ITS premises relies on optimising the use of the road capacity by directly influencing users' behaviour in order to modify travel patterns. Other key aspect of ITS is the integration of different technologies aimed at supplying users' needs on an individual basis.

Using simulation models to aid the assessment and design of traffic systems is an indispensable practice (BARCELÓ, 2001). So, tools should be more expressive and should present the adequate means to cope with reproducing the complexity of ITS scenarios. In this way, many efforts have been identified in order either to elaborate models from scratch or to adapt traditional ones to meet ITS requirements. However, traditional models have shown to be inadequate to handle the innate variability and uncertainty of contemporary traffic and transportation systems (WATLING, 1994). This is specially the case while modelling humanlike behaviours and decision-making processes, which has challenged both researchers and practitioners.

Agent-based techniques seem to be a very appropriate approach to represent such a domain. MAS presents a great potential to represent systems where entities are geographically and functionally distributed. Moreover, there is an explicit commitment of AI approaches to the ontological and epistemological representation of systems. The BDI formalism through its specification of beliefs, desires, and intentions, as well as of their relations, turns modelling humanlike behaviour and decision-making mechanisms into a straightforward and intuitive process. Such a framework features a robust and flexible way of specifying human cognition, which is desirable for ITS assessing tools.

Some efforts in applying multi-agent techniques to the field of traffic and transportation engineering are presented in the literature. Many of them are aimed at dealing with isolated issues, though. Traffic control systems and representing car-following and lane-changing behaviours in microscopic models are some examples of applications that can profit from agent features. However, integrating models aimed at specific aspects in order to lead to more detailed representations of the entire system has not been the purpose, as yet. Nonetheless, it is also necessary to offer mechanisms such that practitioners can evaluate how the real system evolves under the presence of new performance measures brought about by the very integrated ITS technologies.

7.2 Contributions of this thesis

There are three main areas in which the contributions of this research are identified.

- *it suggests a close coupling between ITS and MAS fields*. Following the same envisaging defended in (BOUCHEFRA; REYNAUD; MAURIN, 1995), ITS technologies, their components, and interactions can encounter in MAS the appropriate means to represent their complex nature in different levels of detail. Further, it is possible to identify the potential for a mutual benefit and cross-fertilisation, as can both fields profit from the advances of each other. Given its complex and stochastic nature, and the autonomy desired for its components, ITS can serves MAS as a ground where theories are devised and tested in a real and practical environment. In turn, as the level of detail in simulating traffic scenarios has increased in order to cope with new measures brought about by the ITS premises, it is unpractical to dissociate the design of data structures and algorithms from modelling the dynamics of the real world;
- *it reinforces the feasibility of applying BDI to real-world domains with a relatively large number of reasoning entities*. Turning BDI models into real applications have challenged researchers for a long time. Just quite recently the ‘gap’ between theory and practical implementation has been overcome, as development frameworks and architectures for multi-agent systems now support the effective use of BDI agents. However, as the use of cognitive approaches has suggested, the BDI theory has been relegated to representing domains with a few entities only (GIRAFFA, 1999; TEDESCO; SELF, 2000). In this work, a reasonable number of BDI drivers have been modelled, implemented, and run quite successfully;
- *microscopic traffic simulation can profit from agent-based modelling and simulation techniques toward providing robust and flexible means for assessing human-like behaviour of drivers*. One premise of ITS presented as an alternative to the

increasing of capacity by physical means and to advanced control systems is to optimise the use of existing traffic and transportation resources through influencing users behaviour patterns. This implies dealing with the complex nature of human reasoning and decision-making process. Microscopic simulation models have been specially devised to permit the representation of movement on individual basis and then tried to incorporate driver characteristics. The approach initially adopted in some models was to represent driver and vehicle indistinguishably, as a vehicle-driver unit (HIDAS, 2000). This has been suitable to simulate reactive aspects of the movement, such as car-following and lane-changing behaviours. However, when the issue is to model more complex reasoning and decision making toward a more effective comprehension of the interactions between users and ITS technologies, these models are no longer adequate. This research has suggested to model travellers in general as a single agent structure, which is capable of presenting humanlike reasoning through the use of the BDI approach.

The methodology applied in this thesis can serve to a wide range of applications, both in the field of traffic and transportation engineering, and in the field of multi-agent systems. Understanding driver behaviour, assessing the impact of different ITS technologies, devising strategies for information provision, assessing quality of information content and users' acceptance for different information media are some examples of potential applications of the simulation framework devised in this research. Assessing management strategies, such as inducing the use of a certain mode of transport, applying road pricing, and motivating car polling investments, can also be seen as potentialities for real-world scenarios. On the other hand, the multi-agent community can also profit from the knowledge gathered in designing applications of this sort. Multi-agent interaction and social ability models can be applied to travellers, pedestrians, and even to autonomous assistants that co-operate with users. Communication protocols is another subject of importance as travellers now may access exogenous sources of information and can interact with them. Also, the use of the limited capacity of the transportation infrastructure gives rise to likely conflicting situations, which should be sorted out. As to the perspective of a single reasoning entity, many other issues can be approached through MAS, such as learning, planning, and decision-making mechanisms in order to yield results that may be used in other knowledge domains, as well.

7.3 Further developments

The methodological approach devised in this research allowed to extend the DRACULA microscopic model to support demand generation as the result of decision-making processes carried out by BDI driver agents from within the MADAM model. Although the integration between the demand and the supply has been achieved to some extent, which demonstrated the suitability of using a cognitive-based architecture to improve the driver representation, further developments are suggested in order to enhance the multi-agent simulation framework presented.

- *use of AgentSpeak(XL) to implement the BDI drivers*. Unfortunately, neither the SIM_Speak framework nor the AgentSpeak(XL) interpreter were effectively available during the course of this research in time of being used. Then, the implementation of the AgentSpeak(L) specifications in JAM was the option chosen in order to demonstrate this work's approach. Nonetheless, the step ahead is to incorporate the AgentSpeak(XL) interpreter within the BDI driver structure, so that it will be possible to profit from the advantages of a purpose-built interpreter, such as efficiency and practicality, as claimed in (MACHADO; BORDINI, 2001). In addition, AgentSpeak(L) syntax and operational semantics can serve in this environment as a powerful API through which different behaviours can be implemented and tested;
- *full integration with DRACULA*. The rigid structure of DRACULA is formed of 'black-box' modules that interact with each other via file exchange. MADAM was initially implemented on the same basis, and replaced the original demand module used in DRACULA. However, this limits likely interactions which may emanate, for example, from using Dynamic Route Guidance during the journey. This would imply that the BDI driver could use its reasoning capabilities within the supply model as well. In order to accomplish so, it is necessary to intervene into the supply model and make it fully integrated with the demand side. This discards the drawbacks of exchanging files between modules;
- *implementation of the third scenario*. On the basis of the full integrated demand-supply framework, it is possible to avoid any discontinuity on the reasoning stream of an entity. This way drivers can behave cognitively at any moment within the entire simulation period, including while making in-trip decisions. This facilitates the implementation of the third scenario proposed, as the driver is able to exhibit cognitive abilities while interacting with other elements of the environment, such as exogenous information sources. Thus, the effects of VMS, dynamic route guidance and advice, radio broadcast, and personal assistant systems can be modelled and simulated.
- *validation of more realistic behaviours and calibration of parameters*. In this work, BDI models have been applied just to the specific scenarios of commuting drivers, and has proved to be a powerful tool to model and simulate humanlike reasoning and decision-making processes. Nonetheless, the number of possible interactions between humans and autonomous technologies composing the Intelligent Transportation Systems is considerably vast. In order to investigate the impact of such technologies and assess their efficiency, models should be validated against real world, which means they should reflect what happens in reality. Collecting real-world data in traffic systems has been a difficult task and much effort has been made to automate it. In some application domains, such as traffic

and transportation, it is sometimes necessary to submit subjects to virtual environments so that information on behavioral factors can be gathered in a controlled way (AL-SHIHABI; MOURANT, 2001; BONSALL et al., 1997). Another manner to gain insight into the way people behave is through adopting methodologies that are based on revealed and stated preference analysis (POLYDOROPOULOU et al., 1996a,b; KHATTAK; POLYDOROPOULOU; BEN-AKIVA, 1996; ADLER; RECKER; MCNALLY, 1992). A methodological approach to modelling driver behaviour for a multi-agent implementation is also suggested in (DIA, 2002). Moreover, model parameters should be calibrated so that simulation models can accurately represent field measured or observed conditions of the real world (MILAM; CHOA, 2001; HELLINGA, 1998; RAKHA et al., 1996).

- *concurrent and distributed execution of agents*. Undoubtedly, the more detail a reasoning model is capable of representing and processing the more computing resource and time it will consume. And this is also the case for BDI models, specially when applied to domains formed of several reasoning entities. Improvements to memory and processing of today's computer architectures certainly has motivated and contributed to the application of AI theories and models, in general. Nonetheless, further enhancements are believed to be possible with the concurrent and distributed execution of agent programs. This can both improve the scalability of multi-agent systems and reduce processing time to meet temporal constraints. Some aspects of the Intelligent Transportation Systems could profit from such an implementation strategy.

7.4 Future work

Coupling ITS and MAS in such a close way inspires a wide range of research subjects that can be approached in both fields. Some examples in the literature, as discussed in Section 3.8, illustrate the potential of this synergy. In the special case of the present work, some topics are discussed next as the following-up studies to be pursued.

- *learning mechanisms*. Although commuters are expected to be familiar with the traffic environment and make habitual decisions, experiencing unusual conditions on currency basis may lead individuals to behave differently. Modelling learning mechanisms (WEISS, 1999) as part of the driver cognition can be an interesting tool toward understanding long-term effects of ITS technologies, for instance;
- *dynamic planning*. A BDI agent is basically specified by means of its base beliefs and the plans it can pursue (RAO, 1996; MACHADO; BORDINI, 2001). Nonetheless, it is not rare for a driver to divert from the original chosen path owing, for example, a will to avoid the blocked roads. In such circumstances, the agent must be able to replan its journey. Otherwise, it very likely will experience long delays. However, this can constitute a trick subject as sometimes the traveller is bounded to

temporal constraints and inefficient replanning could become unpracticable (RUSSELL; NORVIG, 1995; BARBER; MARTIN, 1999);

- *driver architecture*. This work presents a relatively simple hybrid architecture for the driver agent. The driver can behave on a reactive basis when performing the journey throughout the network, as implemented in the car-following and lane-changing models. Also, it can exhibit cognitive capabilities when making decisions on the basis of a BDI reasoning kernel. Layered structures has been proposed (FERNANDES, 1998; KLÜGL et al., 2000) that allows for different levels of decisions. The complex nature of the Intelligent Transportation Systems and the wide range of different information that is involved suggest that different AI approaches may be more suitable to one or another interaction. Some examples are encountered in the literature as in (WU; MCDONALD; BRACKSTONE, 1998; NIITTYMÄKI, 2002; NIITTYMÄKI; KIKUCHI, 1998; CHEN; GRANT-MULLER, 2001; KANOH; NAKAMURA, 2000). Thus, a meta specification of a multi-layered traveller architecture could be a means to apply and test different AI theories and approaches. Also such a structure might allow the driver to behave accordingly in different situations by means of dynamically recognising and switching the execution to the control of the most appropriated layer. This selection could be made on the basis of either detailed and accurate behaviours or efficient behaviours with respect to some constraints. These layers could also be oriented to specific purposes, such as acting upon the environment, making decisions on actions, learning, planning, and communicating.

- *Multi-agent environment for ITS*. A fully integrated multi-agent framework for assessing ITS applications should rely on a well-designed multi-agent environment. A parameterised environment and meta specifications of agent architectures are the ingredients to support the perfect integration of travellers and different ITS technologies, as suggested in Section 5.2. Every time a different kind of traveller or ITS-technology agent is integrated into the environment, the adequate mechanisms should be provided for dynamic recognition of communication protocols, level of accessibility to the environment, and likely effects of the agent's actions. The CATIA framework (ROSSETTI, 1998; ROSSETTI; BAMPI, 1998b, 1999) offers an open object-oriented data model, which will be used as the basis for such an integrated multi-agent environment.

7.5 Final comments and remarks

The abstraction premises of MAS and its process-driven approach to systems modelling turns this multidisciplinary field into a precious tool to aid the representation and assessment of complex domains with very stochastic and dynamic nature. Allied to the microscopic perspective, as suggested in (SCHLEIFFER, 2000), agent-based techniques are a step toward the understanding of new performance measures brought about ITS-based solutions. In the specific case of this work, the use of a cognitive model on the basis of the BDI theory has demonstrated a great potential to describe reasoning mechanisms behind the decision-making processes. This way it is possible to overcome the disadvantages of traditional approaches relying on the vehicle-driver unit view. At the same time it provides the ways to yield invaluable insights into the behavioural patterns and responses to Advanced Traveller Information Systems.

APPENDIX A
The OD matrix description file in DRACULA syntax

The OD matrix description file is a text file that specifies how trips are distributed among each origin-destination pair within the network. In the DRACULA environment the file is usually named `<project_name>.mat`, where `<project_name>` is the name for the project being simulated, followed by the extension `<mat>` that identifies the file format. Figure A.1 depicts the content for the file `otley.mat`, used in this thesis.

```
PARAMETERS
nzones=11
nlevel=1
END

002   0.000   0.000    9.430   0.000   0.110    0.000 126.480  13.280  12.910    0.000    0.000
003   0.000   0.000    4.130   0.000   1.820    7.990  43.170   0.000   0.000    0.000   10.280
101  11.760   7.531    0.000   0.000   2.397    0.000  16.833   0.000   0.000  228.480  240.410
102   0.000   1.301    0.000   0.000  18.963    0.000   0.000   0.000   0.000    0.000    0.000
103   0.010   0.020    0.580  14.600   0.000    7.640  48.460   0.000   0.000    0.000    0.500
104   0.000  54.980    0.000   0.000  29.400    0.000  64.350   0.000   0.000    0.000    0.000
105  67.499  44.388   46.698   0.000   2.380  202.391   0.000   0.000   0.000    0.000  145.420
106   0.000   0.000    0.000   0.000   0.000    0.000   0.000   0.000   0.000    0.000    0.000
107  35.941   0.000    0.000   0.000   0.000    0.000   0.000   0.000   0.000    0.000   32.882
108   0.000   0.000   88.980   0.000   0.000    0.000   0.000   0.000   0.000    0.000    0.000
109  52.980   0.000  248.684  94.426   1.920    0.000 285.000   0.000   0.000    0.000    0.000
```

Figure A.1: Otley OD matrix description file.

APPENDIX B
The network description file in DRACULA syntax

The description of the network follows the same approach as it is defined for OD matrices. The network characteristics are also described within a text file following the format described in (LIU, 2001). The network description file in the DRACULA environment is named in the same way as `<project_name>.net`, where `<project_name>` is the name for the project being simulated, followed by the extension `<net>`, which identifies it as a network description file. Figure B.1 depicts the content for the file `otley.net`, used in this thesis. Again, the symbol ⟵ is used throughout the text to split a long line of text in `typewriter` font, due to the space limit imposed by the left and right margins of the page.

options list...

```
&OPTION
 PLOD=F,
&END
```

parameters list...

```
&PARAM
 PRINTF=T,
 BUSPCU=3.0,
 SPEEDS=T,
 XYUNIT=1,
 LTP=60,
 MAXZN=110,
 NITS=12,
 NITA=20,
 MASL=10,
 SAVEIT=T,
 QUANTA=F,
&END
```

network topology...

```
11111
    1    1    0
    0    5    1   40  125
    3    1    0
    0    6    1   30   70
    5    3    1    0    0    0    0    0    0
         1    1   40  125  900  1 1 1200  1 1
         6    0   30  115    0  0 0    0  0 0
         9    1   40   90 1200  1 1  700X 1 1
    6    3    1    0    0    0    0    0    0
         5    2   25  115  900G 1 1  700G 2 2
         3    1   30   70  900  1 1    0  0 0
         7    1   30   12    0  0 0  900  1 1
    7    3    1    0    0    0    0    0    0
         6    1   30   12  900  1 1  500X 1 1
        10    1   30  120  500  1 1  900  1 1
         8    1   20   10  500G 1 1  500G 1 1
    8    1    0    0    0    0    0    0    0
         7    1   20   10
    9    3    1    0    0  101    0    0    0
         5    1   40   90    0  0 0 1200  1 1
        10    1   25  110  600G 1 1  500G 1 1
        15    1   25  110 1200  1 1    0  0 0
   10    4    1    0    0   93    0    0    0
         7    1   30  120  800  1 1  900  1 1  700X 1 1
        11    1   30  170  800G 1 1  900G 1 1  700G 1 1
        17    1   30   65  800  1 1  900  1 1  700X 1 1
         9    0   30  110    0  0 0    0  0 0    0  0 0
   11    1    0
    0   10    1   30  170
   14    1    0
    0   15    1   30  110
   15    4    3    4    0  102    0    0    0
         9    2   25  110  900  1 1 1500  1 1  800X 2 2
        16    1   30  135  900  1 1 1300  1 1  800  1 1
        28    2   30  190  900  1 1 1500  1 2  700X 2 2
        14    1   30  110 1300  1 1 1500  1 1 1000  1 1
             38    3    6   14    9   14   16   14   28
             20    3    6   16   28   16   14   16    9
             13    0    6    9   16    9   28    9   14
             21    3   12    9   16    9   28    9    ↩
   14   28   14   28    9
                             28   16
   16    3    1    0    0   93    0    0    0
        15    1   30  135 1300  1 1    0  0 0
        17    1   30   30    0  0 0 1300  1 1
        24   -2   20  130  500G 1 1  500G 2 2
   17    4    3    4    0   93    0    0    0
        16    2   30   30  700  1 1 1500  1 2  800X 2 2
        10    1   30   65  900  1 1 1100  1 1  900  1 1
        18    2   36  190 1000  1 1 1500  1 1  800X 2 2
        24    1   40  130 1100  1 1 1200  1 1 1000  1 1
             14    8    6   24   16   24   10   24   18
             25    6    6   10   18   10   24   10   16
             24    0   12   16   10   16   18   16    ↩
   24   18   24   18   16
                             18   10
              8    8    6   16   10   16   18   16   24
   18    2    0
    0   21    2   40  160
    0   17    2   40  190
   21    3    1    0    0    0    0    0    0
        18    1   40  160  900  1 1 1000  1 1
        31    2   30  145  900G 1 1  900G 2 2
        24    1   30   85 1000  1 1  800X 1 1
   24    4    1    0    0   93    0    0    0
        16   -1   20  130    0  0 0    0  0 0  500G 1 1
        17    2   40  130  800  1 1 1100  1 2  500X 2 2
        21    2   30   85 1000G 1 1    0  0 0  900G 2 2
        30    2   30   85  500  1 1 1100  1 2  800X 2 2
   27    1    0
    0   37    1   30  290
   28    3    1    0    0  101    0    0    0
        37    1   30   10  800G 1 1  800G 1 1
        15    2   30  190 1200  1 2  600X 2 2
        30    2   40  108  800  1 1 1200  1 2
   30    3    1    0    0    0    0    0    0
        28    2   40  108  800  1 1 1200  1 2
        24    2   30   85  800G 1 1  800G 2 2
        31    1   30   90 1200  1 1  600X 1 1
   31    3    1    0    0    0    0    0    0
        30    2   30   90  900  1 1 1200  1 2
        21    1   30  145  800G 1 1  600G 1 1
        35    1   35  130 1200  1 1  900X 1 1
   35    1    0
    0   31    1   40  130
   37    3    1    0    0    0    0    0    0
        27    1   30  290 1100  1 1  800X 1 1
        28    2   30   10  900  1 1 1100  1 2
        29    2   20  150  800G 1 1  800G 2 2
   29    1    0
    0   37    1   20  150
99999
```

zone information...

```
22222
    2    8    7
    3   21   31   31   21
  101   14   15
  102   27   37
  103   29   37
  104   18   21
  105   35   31
  106    3    6
  107   11   10
  108   18   17
  109    1    5
99999
```

nodes and zones' co-ordinates...

```
55555
    1 3515 3840
    3 3672 3820
    5 3560 3720
    6 3672 3757
    7 3678 3720
    8 3645 3715
    9 3578 3635
   10 3690 3630
   11 3865 3650
   14 3440 3520
   15 3533 3570
   16 3665 3574
   17 3695 3575
   18 3880 3590
   21 3750 3495
   24 3680 3454
   27 3315 3265
   28 3570 3370
   29 3596 3206
   30 3680 3368
   31 3765 3355
   35 3802 3240
   37 3500 3369
C   2 3640 3715
C   3 3785 3473
99999
```

gap acceptance parameters...

```
&GAP
  31  21  10  5 60 180
99999
```

Figure B.1: Otley OD matrix description file.

APPENDIX C
Modelling incidents in DRACULA

Incidents in DRACULA can be modelled by means of reserved lanes. In practical terms, when a lane of a link is marked as reserved, it can only be used by vehicles that are contemplated in the reservation criterion. A number of types are defined and a combination of them can also be used — see (LIU, 2001) for a detailed list. Thus, an incident can be modelled as a blocked lane, for example, by setting it as reserved. This is simply yielded by means of defining the reservation criterion in a text file usually named `<project_name>.pub`, which is used to configure public transport details, as bus services, routes, and stops. Information on reserved lanes are coded under section 8 of the `.pub` file, as illustrated in Figure C.1.

```
01. 88888
02.     9    15    1
03.           0    0    0    0    0 5400
04.    31    21    1
05.           0    0    0    0    0 5400
06. 99999
```

Figure C.1: Code for reserved lane.

The example presented in Figure C.1 is extracted from the file `otley.pub` used in this work. The key `88888` in Line 1 opens the section `8` so as data of reserved lanes can be read. The downstream and upstream nodes, as well as the number of lanes to be marked as reserved are specified in Lines 2 and 4. Fields in Lines 3 and 5 are used to identify the position of the lane, the section length to be reserved, and the period of time such a situation should last. Key `99999` closes the section.

APPENDIX D

Uma abordagem baseada em modelos BDI para avaliação de processos de decisão de motoristas no tráfego urbano

Introdução

O rápido crescimento das regiões metropolitanas tem um impacto significativo nos sistemas de tráfego e transportes. Tem-se verificado um aumento acentuado da demanda que utiliza redes viárias cuja capacidade torna-se cada vez mais limitada. Além de atrasos, a ocorrência freqüente de congestionamentos tem contribuído para perda da qualidade de vida em centros urbanos, assim como provocado prejuízos econômicos, sociais, ambientais e de saúde, muitas vezes irreparáveis. As primeiras tentativas de solução do problema basearam-se na modificação direta da infraestrutura viária, com o objetivo de aumentar a capacidade de atendimento dos fluxos crescentes. A escassez de espaço tem inviabilizado este tipo de solução, tornando-a muito dispendiosa. Nas últimas décadas, uma solução alternativa tem focado a otimização dos sistemas de controle como meio de melhorar a qualidade dos sistemas existentes. Os excelentes resultados desta estratégia motivaram a evolução dos mecanismos de controle. Entretanto, não se pode concluir que se tenha chegado a uma solução definitiva. Recentemente, tem-se observado a utilização crescente de técnicas de computação e comunicação, que passam a fazer parte do quotidiano dos usuários dos sistemas de tráfego e transportes. Estas soluções inovadoras são denominadas de Sistemas Inteligentes de Transportes (ITS), e procuram assegurar a produtividade do sistema através da utilização eficiente dos seus recursos. Estas tecnologias partem da utilização de soluções distribuídas orientadas às necessidades individuais dos usuários. Uma das suas principais premissas é otimizar o desempenho do sistema a partir da influência direta do padrão de comportamento de quem o utiliza, transformando a natureza de dependência temporal e o comportamento humano em fatores de grande importância para modelagem, simulação e avaliação desta abordagem inovadora. O desenvolvimento de ferramentas capazes de auxiliar o processo de avaliação e desenvolvimento da área multidisciplinar de ITS tem sido tema de vários trabalhos, e faz parte da motivação da pesquisa desenvolvida nesta tese.

O comportamento dos usuários de sistemas de tráfego e transportes é complexo por natureza e algumas propriedades são facilmente identificadas na avaliação e caracterização deste comportamento. Um indivíduo (motorista, por exemplo) apresenta autonomia quando planeja sua viagem, e ao interagir com sistemas de informação (que passam a incorporar características do comportamento humano) e com outros usuários, manifesta sua habilidade social. É reativo, por exemplo, quando evita a colisão com outro veículo, e adaptável ao modificar seu comportamento de viagem para evitar os congestionamentos característicos de certas horas do dia. Quando restabelece as prioridades do seu plano de viagens para atingir seus objetivos de forma mais eficiente, o indivíduo também apresenta um comportamento pró-ativo. Estas são algumas característica que pesquisadores da área de Sistemas Multiagentes (MAS) buscam implementar em agentes de software. MAS é uma subárea da Inteligência Artificial Distribuída (DAI) com uma característica fortemente multidisciplinar. Um interesse que tem motivado muita pesquisa neste campo, mais do que solucionar um determinado problema, é a representação do raciocínio envolvido na elaboração da solução. Torna-se, portanto, necessário dispor de técnicas adequadas de abstração e representação do conhecimento, em vários níveis.

A estrutura básica de um agente compreende sensores para captação de informação do ambiente através de percepção, e atuadores através dos quais o agente é capaz de realizar suas ações. Por meio desta estrutura conceitual, também deve ser capaz de interagir com outros agentes, assim como com o ambiente onde está inserido, podendo inclusive alterar seu estado. Dois tipos básicos de agentes podem ser identificados, segundo o nível de raciocínio implementado para deliberação do seu comportamento. As estruturas reativas são as mais simples, baseadas na associação direta de percepções básicas a ações pré-definidas que o agente está apto a executar. Este tipo de estrutura é geralmente utilizada para representar domínios constituídos por um grande número de elementos, onde o desempenho global do sistema resulta do comportamento emergente dos indivíduos que o formam. Por outro lado, as estruturas cognitivas são complexas e implementam mecanismos de raciocínio mais elaborados sobre a representação do conhecimento. Geralmente são utilizadas para representar domínios constituídos apenas por um número reduzido de elementos. Entretanto, no intuito de representar mais realisticamente o comportamento humano, em muitas aplicações de agentes de software interessa que estes sejam capazes de apresentar tanto comportamento reativo como cognitivo, dependendo da situação em que estejam envolvidos.

De maneira geral, a premissa de abstração dos modelos baseados em agentes favorece a representação de sistemas cujos elementos componentes estejam geográfica e funcionalmente distribuídos, como são os componentes nos domínios de tráfego e transportes. Esta perspectiva tem motivado inúmeros trabalhos que sugerem a aplicação de MAS nestes domínios. Na Seção 3.7 do texto da tese, são apresentados alguns exemplos da literatura que reportam potencialidades interessantes das técnicas baseadas em agentes. Estes trabalhos podem ser agrupados em três categorias principais. Um grupo tem focado a aplicação de agentes em sistemas de controle e gerência de tráfego, que tem sido um dos temas mais abordados. A simulação microscópica e a representação do comportamento do motorista têm motivado outro grande grupo de trabalhos. Entretanto, o enfoque principal ainda tem sido a representação do movimento individual em modelos tradicionais como veículo seguidor e mudança de faixas, onde motorista e veículo são tratados indistintamente. Um terceiro grupo de trabalhos engloba aplicações diversas, como representação de pedestres, implementação de sistemas embarcados de auxílio ao processo de condução, e interação com usuários reais em ambientes virtuais simulados.

Objetivo e metodologia do trabalho

Embora se tenha observado a proliferação da aplicação das técnicas baseadas em agentes nos domínios de tráfego e transporte, e se reconheça sua utilidade como ferramenta de avaliação desses domínios em diferentes níveis, pouco se tem evoluído no que diz respeito à representação do motorista como um elemento cognitivo. Alguns trabalhos (KLÜGL et al., 2000; DIA, 2002) têm sugerido a utilização de representações mais adequadas para o processo de decisão do motorista sem, entretanto, terem apresentado sua implementação efetiva. Esta característica torna-se cada vez mais desejada em modelos voltados à avaliação das novas medidas de desempenho impostas pela tecnologia de ITS, onde os padrões de comportamento humano passam a desempenhar papel fundamental.

Este trabalho tem como objetivo principal contribuir para o desenvolvimento de ferramentas computacionais orientadas à modelagem e avaliação do comportamento do motorista, bem como dos efeitos de sua interação com os sistemas inteligentes de transportes em cenários urbanos. A abordagem proposta, em oposição às metodologias tradicionais de representação microscópica do tráfego, trata o motorista como uma entidade intencional, capaz de executar um processo cognitivo na tomada de decisões. Uma arquitetura BDI (*beliefs, desires, intentions*) é utilizada como base das habilidades cognitivas do agente motorista. A lógica BDI foi inicialmente proposta por Rao e Georgeff (1991), inspirada no trabalho filosófico de Bratman (1987). Uma das suas características principais é considerar as intenções como fatores tão importantes no processo cognitivo como as crenças e os desejos. Baseados nesta premissa, os autores formalizam sua teoria estabelecendo as relações entre os estados mentais de crenças, desejos e intenções. A metodologia adotada neste trabalho é basicamente composta pelas seguintes partes: descrição do domínio da aplicação por meio de agentes e suas características; elaboração de um modelo cognitivo para suportar a representação do processo de decisão dos motoristas; escolher uma teoria BDI capaz de suportar a implementação prática do modelo cognitivo do agente motorista; especificar e implementar a arquitetura para o agente motorista cognitivo; implementar um ambiente de simulação microscópica para testar a abordagem proposta; elaborar e executar experimentos de simulação a partir da abordagem sugerida.

Organização do texto

O texto da tese está estruturado em sete capítulos e quatro anexos. No primeiro capítulo apresenta-se uma breve introdução sobre a motivação do trabalho realizado, enfocando o problema do crescente aumento da demanda no tráfego urbano em regiões metropolitanas que resulta na alta incidência de congestionamentos. Os objetivos do trabalho bem como a metodologia utilizada durante sua elaboração também são apresentados. Os principais conceitos dos Sistemas Inteligentes de Transportes, com ênfase nos Sistemas Avançados de Informação aos Viajantes, são brevemente apresentados no segundo capítulo. A área dos Sistemas Multiagentes é introduzida no terceiro capítulo, onde também são apresentados exemplos de aplicação das técnicas baseadas em agentes nos domínios específicos da engenharia de tráfego e transportes. As relações entre os estados mentais de crença, desejo e intenções, que servem de base à especificação da arquitetura BDI utilizada neste trabalho como núcleo cognitivo dos agentes motoristas, são apresentadas no quarto capítulo. A especificação completa modelo cognitivo desenvolvido é apresentada no quinto capítulo. Este modelo foi implementado como uma extensão a um ambiente de simulação microscópica existente. O ambiente de simulação é descrito no sexto capítulo, onde também são apresentados e comentados resultados de uma série de simulações realizadas, com intuito de testar a abordagem que é sugerida nesta tese de doutorado. As conclusões do trabalho são apresentadas e discutidas no sétimo capítulo, seguidas de algumas sugestões para aprimoramento do modelo desenvolvido e idéias de tópicos para futuros tralhos de pesquisa. No primeiro anexo é apresentada a matriz OD utilizada na geração da população de agentes das simulações, seguida da apresentação da descrição da rede viária de Otley, no segundo anexo, e da descrição do mecanismo de geração de incidentes, no terceiro anexo. No quarto e último anexo do texto, é apresentada uma síntese do texto e dos principais resultados da tese, em língua portuguesa, visando uma apresentação formal do trabalho na língua nativa do autor. Não se trata, portanto, da tradução deste trabalho na sua totalidade, mas de um resumo da sua organização, principais características e contribuições.

O modelo multiagente

O domínio de aplicação pode ser representado em termos de múltiplos agentes que interagem entre si e com o ambiente no sentido de melhorar o desempenho do sistema. Assim, todos os componentes de uma arquitetura ITS, como apresentado na Seção 2.4, e ilustrado na Figura 2.1, podem ser descritos a partir de uma arquitetura multiagente, em diferentes níveis de abstração. O enfoque deste trabalho, entretanto, será orientado à estrutura do motorista e ao seu processo de tomada de decisão. Assim, a demanda recorrente resultará das escolhas de cada indivíduo da população de agentes motoristas, de forma descentralizada (ROSSETTI; BORDINI; BAZZAN; BAMPI; LIU; VAN VLIET, 2002). Portanto, é atribuído ao próprio motorista autonomia para identificar suas necessidades, gerir seus recursos, e tomar suas decisões.

Neste trabalho, motoristas são representados a partir de uma abordagem cognitiva, baseada na lógica BDI. A arquitetura do agente motorista é ilustrada na Figura 5.1 e está estruturada em duas camadas principais. A primeira camada é reativa e implementa o movimento do agente através da rede até seu destino. Interações com outros motoristas, como as descritas em modelos de veículo seguidor e mudança de faixa, são implementados nesta camada. A segunda camada é cognitiva e suporta a execução de um interpretador BDI. Esta camada é responsável pela execução dos processos de tomada de decisão do motorista durante o planejamento de uma viagem, como a escolha da rota a ser utilizada e o instante em que a viagem deve começar. O comportamento de um agente motorista é exteriorizado a partir de sensores, que lhe permitem perceber a dinâmica do ambiente, e atuadores, que lhe permitem executar suas ações e exercer seu papel no ambiente. A capacidade de comunicação com outros agentes também é implementada nesta arquitetura, a partir de um mecanismo simples de troca de mensagens. O envio de mensagens por um agente é implementado como uma simples ação, enquanto toda mensagem recebida é identificada a partir de uma percepção. Desta forma, o processo de comunicação pode ser incorporado à semântica operacional do interpretador BDI com facilidade.

Como objetivos e intenções são gerados dinamicamente durante o tempo de execução do agente, a especificação da estrutura cognitiva de um motorista restringe-se à identificação das suas crenças iniciais e de um conjunto de planos não instanciados. Neste trabalho foram implementados diferentes comportamentos, identificados por personalidades, que são basicamente caracterizados pelo grupo de planos que um agente pode vir a executar durante seu processo deliberativo. Esses diferentes comportamentos são descritos em detalhe na Seção 5.3, e especificados com recurso à sintaxe da linguagem AgentSpeak(L).

A seguir, apresenta-se sucintamente cada uma das personalidades e suas principais características. No comportamento aleatório (*random*), motoristas não possuem qualquer tipo de preferência na escolha da rota a ser utilizada, podendo optar por qualquer uma das que conhece nos sucessivos dias do período de simulação. Após selecionado o caminho, a hora de partida para início da viagem é avaliada com relação ao tempo de chegada desejado e o tempo de viagem esperado para a rota escolhida, que é igual ao tempo experimentado durante a última jornada realizada pelo itinerário selecionado. O tempo de partida será então igual ao tempo de chegada desejado menos o tempo de viagem previsto. O motorista seletivo (*choosy*), por outro lado, sempre tentará encontrar a melhor rota em termos de tempo de viagem. Esta estratégia é implementada a partir da comparação sucessiva de todos os tempos estimados para cada rota conhecida. Identificada a melhor rota, a escolha do tempo de partida segue a mesma abordagem implementada no comportamento aleatório. A terceira personalidade é identificada pelo comportamento conservador (*conservative*). Um motorista conservador nunca altera sua opção de rota, selecionando sempre o seu itinerário usual, ainda que este seja a pior opção em termos de tempo de viagem. Sua seleção de tempo de partida também segue a estratégia dos dois comportamentos anteriores. As três personalidades apresentadas acima foram modeladas a partir de uma avaliação intuitiva de possíveis padrões de comportamentos de motoristas, mas não correspondem à realidade de forma plena.

Uma quarta personalidade, representada pelo comportamento habitual (*habitual*), foi implementada a partir das estratégias de escolha de tempo de partida e de rota sugeridas em (LIU; VAN VLIET; WATLING, 1999). Ao contrário dos comportamentos anteriores, a primeira decisão de um motorista habitual é sobre o tempo de partida para início da viagem. Esta seleção é sempre realizada em função de uma percepção de atraso, da última viagem realizada. A percepção de atraso é definida como a diferença entre o atraso absoluto e uma tolerância, avaliada em relação ao tempo de viagem experimentado no dia. Assim, o tempo de partida para a próxima viagem, que será realizada no dia seguinte, será ajustado apenas pela diferença do atraso percebido, se este for maior que zero. Desta forma, um motorista habitual tenderá a manter seu último tempo de partida sempre que seu atraso percebido no dia anterior não for maior do que uma dada tolerância. Uma característica importante deste comportamento é sua indiferença às chegadas antecipadas, ou seja, anteriores ao tempo de chegada desejado; neste caso nenhum ajuste é computado ao tempo de partida. A rota será sempre a usual, a menos que a melhor rota, em termos de tempo de viagem, seja consideravelmente melhor. Esta avaliação é feita tanto em termos relativos, considerando um ganho relativo ao tempo de viagem estimado para a rota usual, como em termos absolutos. O motorista utiliza o maior ganho, entre relativo e absoluto, para condicionar sua eventual mudança de rota.

A personalidade representada pelo comportamento habitual aproxima-se mais do que se verifica em sistemas reais. Entretanto, sua indiferença à chegada antecipada e sua tolerância condicionada ao tempo de viagem experimentado motivaram a extensão deste modelo. A primeira modificação implementada para este comportamento foi considerar também uma tolerância à antecipação, em adição à tolerância ao atraso, ainda que em termos relativos ao tempo de viagem experimentado. Uma segunda modificação implementada foi considerar as tolerâncias à antecipação e ao atraso em termos absolutos. Desta forma, em ambas modificações, a indiferença, tanto à antecipação como ao atraso, é função do nível de tolerância assumido pelo motorista. Os comportamentos apresentados acima foram implementados como planos não instanciados, descritos em detalhe nas seções 5.3.2 e 5.3.3 do texto da tese.

O ambiente de simulação MADAM+DRACULA

Com intuito de testar a abordagem proposta, foi desenvolvido um ambiente de simulação microscópica, proposto inicialmente em (ROSSETTI; BAMPI; LIU; VAN VLIET; CYBIS, 2000b) como uma extensão ao modelo de simulação implementado no ambiente DRACULA (Dynamic Route Assignment Combining User Learning and microsimulAtion) apresentado em (LIU; VAN VLIET; WATLING, 1999). O simulador DRACULA tem sido desenvolvido na Universidade de Leeds desde 1995, e baseia-se em dois conceitos principais: demanda (*demand*) e oferta (*supply*). A demanda representa a dinâmica de formação dos fluxos recorrentes em cada dia, enquanto oferta está associada ao desempenho de cada arco da rede. A relação entre estes dois conceitos fundamentais está ilustrada na Figura 6.1, constituindo um processo iterativo que proporciona ao modelo de simulação a representação das dinâmicas do sistema durante um dia (*within-day dynamics*) e ao longo de vários dias consecutivos (*day-to-day dynamics*). Este processo é representado em mais detalhes na Figura 6.2. Após a geração da população de motoristas no início da simulação, a demanda diária é formada a partir da atribuição de tempos de partida e opção de rota para cada indivíduo. Esta atribuição é realizada a partir de funções que implementam alguma distribuição definida pelo usuário, de forma centralizada, alterando os valores de parâmetros das estruturas. Cada motorista é então posto a executar sua viagem na rede, iniciando-a no tempo de partida e itinerário que lhe foram atribuídos. Finalizada a viagem, as medidas de desempenho experimentadas durante o seu curso são “memorizadas” para que possam ser utilizadas pela função de atribuição na próxima iteração. Durante cada dia, um modelo de variabilidade de oferta é aplicado sobre a rede para emular a dinâmica do sistema ao longo de vários dias.

Na extensão proposta para o modelo de simulação implementado no DRACULA, demanda é gerada como resultado do processo de tomada de decisão autônomo, executado individualmente pelos agentes motoristas. Depois de escolher rota e tempo de partida, o agente iniciará sua trajetória ao destino desejado. Outra característica da estrutura proposta, como representado na Figura 6.3, é a presença de agentes que implementam o comportamento inteligente das novas soluções baseadas em tecnologias ITS. Estes agentes, como por exemplo os sistemas de informação, poderão estar integrados à iteração de simulação durante a formação da demanda assim como durante a execução microscópica do movimento.

MADAM (Multi-Agent DemAnd Model) é a implementação do modelo de demanda desenvolvido neste trabalho, baseado em uma população multiagente, integrado ao ambiente de simulação DRACULA, como representado na Figura 6.4. A interface entre os dois modelos, de demanda e de oferta, é implementada com base na troca de arquivos com sintaxe comum aos dois módulos. A população de agentes é estimada a partir da distribuição de fluxos representada em uma matriz origem-destino (OD). As opções de rotas entre cada par OD também são associadas aos motoristas durante a formação da população. Durante os dias que constituem o período de simulação desejado (onde cada dia representa uma iteração da relação demanda-oferta), os agentes da população escolhem suas opções de rota e tempo de partida, a partir da execução de um processo deliberativo baseado na lógica BDI. Estas opções são colecionadas em um arquivo (*Input MA*) que alimenta o modelo de oferta. Todas as percepções dos agentes durante o curso das suas viagens individuais (descritas em um arquivo *Output MA*) integrarão a base de conhecimento de cada agente e poderão ser utilizadas para melhorar a qualidade de suas decisões nas iterações futuras. O ambiente de simulação foi implementado nas linguagens C/C++, seguindo a mesma estratégia de implementação adotada no desenvolvimento da estrutura inicial do DRACULA. Entretanto, a camada cognitiva do agente motorista foi implementada em Java, permitindo assim a utilização do interpretador BDI JAM (HUBER, 1999a,b), no processo deliberativo.

Experimentos e análise de resultados

Dois cenários básicos foram criados para ilustrar a aplicação da abordagem proposta, e simulados com diferentes configurações. Os experimentos, apresentados na Seção 6.5, foram realizados para a rede de Otley, ilustrada na Figura 6.6, com população sintetizada a partir da matriz OD descrita no Apêndice A, durante um período de 101 dias (transcorrido entre os dias 0 e 100).

Primeiro cenário

No primeiro cenário não foi considerada a presença de qualquer tipo de sistema de informação capaz de auxiliar os motoristas no planejamento da viagem antes do seu início ou durante seu curso. Foram analisadas as viagens realizadas entre as zonas 109 e 105. As três opções de rotas consideradas são apresentadas na Figura 6.7. Inicialmente apenas populações formadas por motoristas de mesma personalidade foram simuladas. Os gráficos das Figuras 6.8, 6.9, 6.10 e 6.11, são utilizados para ilustrar o comportamento individual das personalidades aleatória, seletiva, conservadora, e habitual, respectivamente, a partir dos dados relativos a um único motorista da população. É possível observar nos gráficos mencionados a evolução das escolhas de tempo de partida e de rota, para cada comportamento. O motorista aleatório apresenta grande oscilação na escolha entre as três opções de rota. O motorista seletivo, que escolhe a melhor rota em termos de expectativa de tempo de viagem, apresenta menor alternação nas opções selecionadas. Apesar das três opções de rota, o motorista conservador mantém a mesma seleção ao longo dos dias simulados. Nos três comportamentos, entretanto, verifica-se uma grande oscilação na escolha do tempo de partida. Dada a rigidez da estratégia de seleção de tempo de partida, que não considera qualquer tolerância à antecipação ou ao atraso procurando sempre chegar exatamente no horário desejado, o motorista tende a ajustar diariamente seu tempo de partida. Por outro lado, a flexibilidade do comportamento habitual (Figura 6.11), caracterizada pela tolerância ao atraso e indiferença à antecipação, permite maior constância na seleção do tempo de partida. A mesma tendência é verificada na seleção de rota, condicionada pela necessidade de uma melhoria significativa.

Os gráficos da Figura 6.12 ilustram os tempos de viagem médios para cada população homogênea. A grande oscilação do tempo de viagem individual dos motoristas aleatórios, seletivos e conservadores é refletida no tempo médio de viagem para o par OD. O mesmo acontece com o comportamento habitual, onde a pequena oscilação dos tempos de viagem experimentados pelos indivíduos da população resultam em pequena variação do tempo médio de viagem para o mesmo par OD. Os valores de tempos médios e respectivos desvios padrão são apresentados na Tabela 6.1. Uma observação interessante é o alto valor da média verificada para a população de motoristas habituais, em relação às outra populações. A flexibilidade característica deste comportamento permite que tempos de viagem elevados sejam tolerados, desde que a condição de chegada anterior ao limite tolerado de atraso seja satisfeita.

Certamente que populações homogêneas representam situações hipotéticas que não são verificadas em sistemas reais. Uma segunda série de experimentos foi realizada, onde a composição da população foi variada em termos da fração de motoristas de mesma personalidade, como apresentado na Tabela 6.2. Os gráficos da Figura 6.13 ilustram a variação do tempo de viagem média para cada configuração de população, considerando as viagens realizadas entre as zonas 109 e 105 da rede simulada neste trabalho. Em todas as situações, verifica-se uma oscilação nos valores dos tempos médios, mesmo para a população com predominância de motoristas habituais. Ainda que neste caso se verifique o menor desvio padrão, a população é afetada pela rigidez do comportamento de motoristas como os aleatórios, seletivos e conservadores.

Nos gráficos das Figuras 6.14 e 6.15, ilustra-se o comportamento de escolha de tempo de partida para as duas extensões propostas ao comportamento habitual, ou seja, motoristas com tolerâncias à antecipação e ao atraso relativas ao tempo de viagem, e motoristas com tolerâncias à antecipação e ao atraso absolutas, respectivamente. Nesta série de experimentos, apenas populações homogêneas foram consideradas. Na Tabela 6.3 encontram-se listados os valores para as tolerâncias relativas, enquanto valores para as tolerâncias absolutas encontram-se na Tabela 6.4. Verifica-se uma grande variação na escolha do tempo de partida para motoristas com tolerâncias relativas, ainda que se aumente o nível de tolerância à antecipação para 100% do tempo de viagem experimentado. Como a tolerância é condicionada pelo tempo de viagem, para tempos longos a tolerância será maior, e será menor para tempos mais curtos. A oscilação dos tempos de viagens experimentados fazem com que os limites superior e inferior de tolerância do motorista também apresentem grande variabilidade, dificultando que o tempo de chegada verificado permaneça entre esses dois limites (Figura 6.14). Para tolerâncias absolutas, como os limites inferior e superior de tolerância não dependem do tempo de viagem e permanecem constantes ao longo do período de simulação, torna-se cada vez mais fácil fazer com que o tempo de chegada verificado permaneça entre esses limites (Figura 6.15). As variações dos tempos médios de viagem são apresentadas nos gráficos da Figura 6.16.

Segundo cenário

No segundo cenário considera-se a presença de sistemas de informação capazes de antecipar ao motorista o estado predominante da rede. Nenhuma interação do motorista com a fonte de informação é possível, entretanto. Além das viagens realizadas no par OD 109-105, também foram analisadas as viagens dos pares 105-104 e 101-002, com duas opções de rotas para cada par, apresentadas nas Figuras 6.17 e 6.18, respectivamente.

Com o objetivo de verificar o impacto das informações antecipadas sobre o estado do sistema, dois incidentes foram provocados em dois arcos da rede após o qüinquagésimo dia, permanecendo até o fim do período de simulação. Os incidentes foram inseridos na rede a partir da exclusão de uma das faixas de tráfego nos arcos 9-15 e 31-21. As populações utilizadas neste experimento são homogêneas, formadas por motoristas habituais. Dois parâmetros foram variados: o número total de agentes em cada população, a partir de um fator de multiplicação (*population factor*) aplicado ao número total de viagens da matriz OD original; e a fração de motoristas informados sobre os incidentes. As diferentes configurações de população são apresentadas na Tabela 6.5.

Os gráficos das Figuras 6.19, 6.20 e 6.21 ilustram os tempos médios de viagem para os pares 109-105, 105-104, e 101-002, respectivamente. Em todas as situações, verifica-se uma tendência dos tempos médios de viagens de estabilizarem em patamares diferentes do patamar original, anterior à aplicação dos incidentes no qüinquagésimo dia. Interessante observar que, no caso do par 105-104 (Figura 6.20), o sistema passa a estabilizar em patamares até mesmo inferiores à situação anterior aos incidentes. Alguns fatores podem estar associados a este comportamento, como por exemplo, a influência de outras rotas em arcos comuns aos das opções de rotas para este par OD, ou ainda a própria flexibilidade do comportamento dos motoristas habituais, que são indiferentes à antecipação. Observa-se contudo que, em quase todas as situações (para os três pares OD), uma população desinformada sobre incidentes na rede configuram os piores casos, estabilizando quase sempre nos piores patamares. Por outro lado, como no par 109-105 (Figura 6.19), informar todos os motoristas da população pode não ser a melhor estratégia. A variação do número de motoristas na rede, para os três pares OD, também parece exercer influência na configuração do sistema após a aplicação dos incidentes, não se verificando o mesmo padrão de reorganização do sistema para o mesmo par OD.

Conclusões

As características dos sistemas multiagentes, principalmente no que refere à sua premissa dirigida ao processo, torna este campo multidisciplinar um terreno fértil para a emergência de ferramentas orientadas à modelagem e à avaliação de sistemas complexos de natureza fortemente dinâmica. Aliados à abordagem de representação microscópica do tráfego, MAS apresentam-se como uma metodologia adequada que pode contribuir para o entendimento das novas medidas de desempenho impostas pelas soluções baseadas em tecnologias ITS.

Principais contribuições do trabalho

No caso específico deste trabalho, o uso de um modelo cognitivo baseado em uma teoria BDI demonstrou um grande potencial na descrição dos mecanismos de raciocínio envolvidos no processo de decisão. Três principais contribuições são identificadas neste trabalho. Em primeiro, o tema abordado nesta pesquisa promove uma associação mais estreita entre duas áreas multidisciplinares, MAS e ITS, sugerindo uma interface que produz benefícios mútuos. As tecnologias ITS encontram em MAS meios adequados para representarem sua natureza complexa, em diferentes níveis de abstração. Por outro lado, ITS também apresenta-se como um domínio de aplicação desafiador para os especialistas de MAS, onde podem testar suas técnicas e métodos em problemas reais, estimulando o desenvolvimento dos sistemas multiagentes. Em segundo, a aplicação de uma teoria BDI na elaboração do modelo cognitivo do motorista, sua implementação efetiva, e sua aplicação e simulação em diferentes cenários de tráfego, demonstram que é factível a utilização de agentes BDI para representação e simulação de sistemas compostos por um grande número de elementos cognitivos. Ao contrário de abordagens tradicionais, que optam pela utilização de agentes reativos na representação e simulação de sistemas com muitos elementos, as simulações de populações compostas com um considerável número de motoristas BDI foram realizadas com êxito. Este resultado também serve para motivar a utilização de abordagens baseadas em agentes BDI em aplicações reais similares. Finalmente, este trabalho demonstra a validade de modelos e simulação baseados em agentes como ferramenta adequada à representação do comportamento humano em simulação microscópica de tráfego. A representação e avaliação das características humanas têm papel fundamental no entendimento do impacto das tecnologias ITS, e grande esforço ainda tem sido orientado no desenvolvimento de modelos e ferramentas capazes de considerar e tratar as novas medidas de desempenho relacionadas com o perfil do usuário. O ambiente de simulação implementado representa uma grande contribuição neste sentido, permitindo a representação explícita do comportamento humano e sua interação com as tecnologias ITS.

Propostas para desenvolvimentos e trabalhos futuros

A abordagem metodológica sugerida permitiu a extensão do modelo microscópico do DRACULA que passa a suportar a geração de demanda por grande número de agentes. Entretanto, alguns desenvolvimentos ainda precisam ser realizados. Um primeiro passo seria a integração de interpretador capaz de executar os planos especificados em AgentSpeak(L). No momento, a camada cognitiva do agente motorista é implementada em Java, integrando um interpretador JAM usado para emular a semântica operacional da linguagem desenvolvida por Rao (1996). Como solução, a arquitetura do agente passará a incorporar o interpretador AgentSpeak(XL), apresentado em (BORDINI et al., 2002). Há também a necessidade de melhorar a integração operacional entre a implementação do modelo de demanda, MADAM, e o software DRACULA. A interface entre os dois módulos é implementada a partir da troca de arquivos, o que dificulta a extensão do modelo multiagente. Esta integração permitirá a implementação do terceiro cenário, que considera a interação do agente motorista com fontes exógenas de informação durante a execução da viagem, permitindo-lhe corrigir seu curso adequadamente. No momento, não é permitido aos agentes alterar seus itinerários, que são definidos antes do início da viagem. No que se refere aos modelos de comportamento, é necessária a validação e calibração para que possam ser utilizados em cenários reais. Outros modelos também podem ser facilmente integrados. Neste sentido, a linguagem AgentSpeak(L) poderá servir como uma interface de programação (API) para a implementação e simulação de diferentes estratégias de decisão. Por fim, sugere-se a utilização de técnicas de programação e desenvolvimento que permitam melhorar o tempo de execução das simulações, como, por exemplo, a execução distribuída ou paralela de agentes.

Além das propostas de aprimoramento do trabalho realizado durante o programa de doutorado, os resultados obtidos com sua conclusão motivaram algumas sugestões que podem ser utilizadas como temas de novos projetos de pesquisa. A implementação de diferentes mecanismos de aprendizagem e buscar capacitar os agentes a elaborar seus planos dinamicamente podem contribuir no desenvolvimento de modelos de comportamento dos motoristas. O desenvolvimento de uma meta-arquitetura para o agente motorista, capaz de suportar diferentes representações do conhecimento e processos de raciocínio (não apenas BDI) pode facilitar a integração de diferentes níveis de tomada de decisão. Por fim, a criação de um ambiente multiagente integrado, para modelagem, simulação e avaliação de diferentes tecnologias ITS, representa um desafio motivante para um futuro projeto neste domínio de pesquisa.

REFERENCES

ADAMATTI, D. F. **Estudo comparativo de ambientes de simulação baseados em agentes**. 2001. Trabalho Individual (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

ADLER, J. L.; BLUE, V. J. Toward the design of itelligent traveler information systems. **Transportation Research Part C: Emerging Technologies**, Amsterdam, v.3, n.6, p.157–172, June 1998.

ADLER, J. L.; RECKER, W. W.; MCNALLY, M. G. Using interactive simulation to model driver behavior under ATIS. In: CONFERENCE ON MICROCOMPUTER IN TRANSPORTATION, 4., 1992, Baltimore. **Proceedings. . .** Reston: American Society of Civil Engineers, 1992. p.344–355.

ADORNI, G.; POGGI, A. Route guidance as a just-in-time multiagent task. **Applied Artificial Intelligence**, London, v.10, n.2, p.95–120, April 1996.

AGENTLINK. **Agent Software**. Southampton, UK, 2002. Available at: ⟨http://www.agentlink.org/resources/agent-software.html⟩. Visited on Sept. 4, 2002.

AL-SHIHABI, T.; MOURANT, R. R. A framework for modelling human-like driving behaviors for autonomous vehicles in driving simulators. In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS, 5., 2001, Montreal. **Proceedings. . .** New York: ACM Press, 2001. p.286–291.

AOS. **JACK Intelligent Agents User Guide**. Carlton South, Australia: Agent Oriented Software Pty., 2002. Manual.

BAR-YAM, Y. **Dynamics of Complex Systems**. New York: Westview Press, 1997. (Studies in Nonlinearity Series).

BARBER, S.; MARTIN, C. Applying Dynamic Planning Frameworks to Agent Goals. In: MUSLINER, D.; PELL, B. (Ed.). **Agents with Adjustable Autonomy**. Menlo Park: The AAAI Press, 1999. (AAAI Spring Symposium Series, n.SS-99-06).

BARCELÓ, J. Microscopic Traffic Simulation: a tool for the analysis and assessment of ITS systems. In: HIGHWAY CAPACITY COMMITTEE, HALF YEAR MEETING, 2001, Lake Tahoe. **Proceedings. . .** [S.l.: s.n.], 2001.

BARFIELD, W.; CONQUEST, L.; SPYRIDAKIS, J.; HASELKORN, M. Information requirements for real-time motorist information systems. In: IEEE CONFERENCE ON

VEHICLE NAVIGATION INFORMATION SYSTEMS, VNIS, 1989, Toronto. **Proceedings...** Piscataway: IEEE, 1989. p.101–104.

BARFIELD, W.; MANNERING, F. Behavioral and human factors issues in advanced traveler information systems. **Transportation Research Part C: Emerging Technologies**, Amsterdam, v.1, n.2, p.105–106, 1993.

BARRETT, R.; RAMSAY, A.; SLOMAN, A. **POP-11**: a practical language for artificial intelligence. Chichester: Ellis Horwood, 1985.

BAZZAN, A. L. C. **An evolutionary game–theoretical approach for coordination of traffic signal agents**. 1997. PhD Thesis — Karlsruhe University, Karlsruhe, Germany.

BAZZAN, A. L. C.; ANDRIOTTI, G. K.; ROSSETTI, R. J. F.; WAHLE, J. Modelling driver agents within an object-oriented microscopic traffic simulator. In: WORKSHOP ON AGENTS IN TRAFFIC AND TRANSPORTATION, 2001, Sydney. **ITS World Congress**. [S.l.: s.n.], 2001.

BEN-AKIVA, M.; DE PALMA, A.; KANAROGLOU, P. Dynamic model of peak period traffic congestion with elastic arrival rates. **Transportation Series**, [S.l.], v.20, n.2, p.164–181, August 1986.

BÉRON, F.; CARPUAT, B.; GLIZE, P.; RÉGIS, C. La résolution de conflit sans négociation: modèle et évaluation. In: JOURNÉES FRANCOPHONES SUR L'INTELLIGENCE ARTIFICIELLE DISTRIBUÉE ET LES SYSTÈMES MULTI-AGENTS, JFIADSMA, 3., 1995, St Baldoph. **Actes...** London: Hermes Science, 1995.

BIRTWISTLE, G. M.; DAHL, O.-J.; MYHRHAUG, B.; NYGAARD, K. **SIMULA begin**. New York: Petrocelli/Charter, 1975.

BOCHNER, B. S. Realizing the potential of intelligent transportation systems (ITS). **ITE Journal**, Washington, v.68, n.10, p.16–17, October 1998.

BONSALL, P.; FIRMIN, P.; ANDERSON, M.; PALMER, I.; BALMFORTH, P. Validating the results of a route choice simulator. **Transportation Research Part C: Emerging Technologies**, Amsterdam, v.5, n.6, p.371–387, December 1997.

BORDINI, R. H.; BAZZAN, A. L. C.; JANNONE, R. O.; BASSO, D. M.; VICARI, R. M.; LESSER, V. R. Using the DTC scheduler to improve intention selection in BDI agents. In: MEETING FOR THE EVALUATION OF THE COOPERATIVE RESEARCH PROJECTS SUPPORTED BY CNPq, ProTeM-CC PROGRAMME, 2001, Rio de Janeiro. **Proceedings...** Brasília: CNPq, 2001.

BORDINI, R. H.; BAZZAN, A. L. C.; O. JANNONE, R. de; BASSO, D. M.; VICARI, R. M.; LESSER, V. R. AgentSpeak(XL): efficient intention selection in BDI agents via decision-theoretic task scheduling. In: INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS, AAMAS, 1., 2002, Bologna. **Proceedings...** New York: ACM Press, 2002. p.1294–1302.

BOTELHO, L. M. A control structure for agent interaction. In: IEEE INTELLIGENT VEHICLES SYMPOSIUM, IV, 2000, Dearborn. **Proceedings...** Piscataway: IEEE, 2000. p.398–403.

BOTELHO, L. M.; RAMOS, P. Emotionally controlled inter-agent communication. In: IEEE INTELLIGENT VEHICLES SYMPOSIUM, IV, 2000, Dearborn. **Proceedings...** Piscataway: IEEE, 2000. p.410–417.

BOUCHEFRA, K.; REYNAUD, R.; MAURIN, T. IVHS viewed from a multiagent approach point of view. In: IEEE INTELLIGENT VEHICLES SYMPOSIUM, IV, 1995, Detroit. **Proceedings...** Piscataway: IEEE, 1995. p.113–117.

BRAESS, D. Über ein Paradoxon aus der Verkehrsplanung. **Unternehmensforschung**, [S.l.], v.12, p.258–268, 1969.

BRATMAN, M. E. **Intentions, plans, and practical reason**. Massachusetts: Harvard University Press, 1987.

BROOKS, R. A. Intelligence without representation. **Artificial Intelligence**, Amsterdam, v.47, n.1–3, p.139–159, January 1991.

BROOKS, R. A. Intelligence Without Reason. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, IJCAI, 12., 1991, Sydney. **Proceedings...** San Fransisco: Morgan Kaufmann, 1991. p.569–595.

BRUGGEMANN, U.; LEHMANN, H. Systems Analysis of the BMBF project “Ecologically Tolerable Mobility in Urban Regions”. In: WORKSHOP ON AGENTS IN TRAFFIC AND TRANSPORTATION, 2000, Barcelona. **Autonmous Agents**. [S.l.: s.n.], 2000.

BURMEISTER, B. DASEDIS: a development environment for agent–oriented programming. In: MÜLLER, H. J. (Ed.). **Verteilte Künstliche Intelligenz**. [S.l.: s.n.], 1993.

BURMEISTER, B.; DOORMANN, J.; MATYLIS, G. Agent-oriented traffic simulation. **Transactions of SCS**, San Diego, v.14, n.2, p.79–86, 1997.

BURMEISTER, B.; HADDADI, A.; MATYLIS, G. Application of multi–agent systems in traffic and transportation. **IEE Proceedings on Software Engineering**, London, v.144, n.1, p.51–60, 1997.

BURMEISTER, B.; HADDADI, A.; SUNDERMEYER, K. Generic, Configurable, Cooperation Protocols for Multi-Agent Systems. In: CASTELFRANCHI, C.; MÜLLER, J. P. (Ed.). **From reaction to cognition**. Berlin: Springer–Verlag, 1995. p.157–171. (Lecture Notes in Artificial Intelligence, v.957).

BURMEISTER, B.; SUNDERMEYER, K. Cooperative problem-solving guided by intentions and perceptions. In: EUROPEAN WORKSHOP ON MODELLING AUTONOMOUS AGENTS IN A MULTI-AGENT WORLD, MAAMAW, 3., 1991, Kaiserslautern, Germany. **Proceedings...** Amsterdam: North-Holland, 1992. p.77–92.

CACI PRODUCTS COMPANY. **SIMSCRIPT II.5**: programming language. La Jolla, CA: CACI Products Company, 1983.

CARRIERO, N.; GELERNTER, D. LINDA in context. **Communications of ACM**, New York, v.32, n.4, p.444–458, April 1989.

CASTELFRANCHI, C. Modeling social action for AI agents. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, IJCAI, 15., 1997, Nagoya, Japan. **Proceedings...** San Diego: Morgan Kaufmann, 1997. p.1567–1576.

CASTELFRANCHI, C. Simulating with cognitive agents: the importance of cognitive emergence. In: SICHMAN, J. S.; CONTE, R.; GILBERT, N. (Ed.). **Multi-agent systems and agent-based simulation**. Berlin: Springer–Verlag, 1998. p.26–44. (Lecture Notes in Artificial Intelligence, v.1534).

CASTELFRANCHI, C.; CONTE, R. Emergent functionality among intelligent systems: cooperation within and without minds. **AI & Society**, London, v.6, n.1, p.78–87, January–March 1992.

CHAIB-DRAA, B.; LEVELSQUE, P. Hierarchical model and communication by signs, signals and symbols in multi-agent environments. In: EUROPEAN WORKSHOP ON MODELLING AUTONOMOUS AGENTS IN A MULTI-AGENT WORLD, MAAMAW, 6., 1994, Odense, Denmark. **Proceedings...** Berlin: Springer, 1996. p.151–165. (Lecture Notes in Computer Science, v.1069).

CHAMPION, A.; ESPIÉ, S.; MANDIAU, R.; KOLSKI, C. Multi-agent road traffic simulation: the coordination issue. In: EUROPEAN SIMULATION SYMPOSIUM, ESS, 13., 2001, Marseille, France. **Proceedings...** Ghent: SCS Europe, 2001.

CHAMPION, A.; MANDIAU, R.; ESPIÉ, S.; KOLSKI, C. Multi-agent road traffic simulation: towards coordination by game theory based mechanisms. In: WORKSHOP ON AGENTS IN TRAFFIC AND TRANSPORTATION, 2001, Sydney. **ITS World Congress**. [S.l.: s.n.], 2001.

CHATTERJEE, K.; MCDONALD, M. Modelling the impacts of transport telematics: current limitations and future developments. **Transport Reviews**, Abingdon, v.19, n.1, p.57–80, January 1999.

CHEN, H.; GRANT-MULLER, S. Use of sequential learning for short-term traffic flow forecasting. **Transportation Research Part C: Emerging Technologies**, Amsterdam, v.9, n.5, p.319–336, October 2001.

CHERRETT, T. J. **Congestion and driver response**. 1998. PhD Thesis — University of Southampton, Department of Civil and Environmental Engineering, Southampton, UK.

CODD, E. F. **Cellular Automata**. New York: Academic Press, Inc., 1968.

COHEN, P. R.; LEVESQUE, H. J. Persistence, intention and commitment. In: THE WORKSHOP ON REASONING ABOUT ACTIONS AND PLANS, 1986, Timberline, OR. **Proceedings...** San Mateo: Morgan Kaufmann, 1987. p.297–340.

COHEN, P. R.; LEVESQUE, H. J. Intention is choice with commitment. **Artificial Intelligence**, Amsterdam, v.42, n.2-3, p.213–261, March 1990.

CONTE, R.; GILBERT, N.; SICHMAN, J. S. MAS and social simulation: a suitable commitment. In: SICHMAN, J. S.; CONTE, R.; GILBERT, N. (Ed.). **Multi-agent systems and agent-based simulation**. Berlin: Springer, 1998. p.1–9. (Lecture Notes in Artificial Intelligence, v.1534).

COPSTEIN, B. **SIMOO**: plataforma orientada a objetos para simulação discreta multi-paradigma. 1997. Tese (Doutorado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

CORRÊA, M.; COELHO, H. Around the architectural agent approach to model conversations. In: EUROPEAN WORKSHOP ON MODELLING AUTONOMOUS AGENTS IN A MULTI-AGENT WORLD, MAAMAW, 5., 1993, Neuchatel, Switzerland. **Proceedings...** Berlin: Springer, 1995. p.172–185. (Lecture Notes in Computer Science, v.957).

CUENA, J.; HERNÁNDEZ, J. Z.; MOLINA, M. Knowledge oriented design of an application for real time traffic management: the TRYS system. In: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, ECAI, 12., 1996, Budapest, Hungary. **Proceedings...** Chichester: John Wiley & Sons, 1996. p.308–312.

CUENA, J.; OSSOWSKI, S. Distributed models for decision support. In: WEISS, G. (Ed.). **Multiagent systems**: a modern approach to distributed artificial intelligence. Cambridge, MA: The MIT Press, 1999. p.459–504.

DEMAZEAU, Y. Coordination patterns in multi-agent worlds: application to robotics and computer vision. In: IEE COLLOQUIUM ON INTELLIGENT AGENTS, 1991, London. **Proceedings...** London: IEE, 1991. p.5/1–5/9.

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY WUERZBURG. **SeSAm Multi-agent simulation environment**. Wuerzburg, Germany, 2002. Available at: ⟨http://ki.informatik.uni–wuerzburg.de/sesam/⟩. Visited on Feb. 28, 2002.

DIA, H. A conceptual framework for modelling dynamic driver behaviour using intelligent agents. In: INTERNATIONAL CONFERENCE ON APPLICATIONS OF ADVANCED TECHNOLOGIES IN TRANSPORTATION ENGINEERING, 6., 2000, Singapore. **Proceedings...** Reston: American Society of Civil Engineers, 2000.

DIA, H. An agent-based approach to modelling dynamic driver behaviour. In: WORKSHOP ON AGENTS IN TRAFFIC AND TRANSPORTATION, 2001, Sydney. **ITS World Congress**. [S.l.: s.n.], 2001.

DIA, H. An agent-based approach to modelling driver route choice behaviour under the influence of real-time information. **Transportation Research Part C: Emerging Technologies**, Amsterdam, v.10, n.5–6, p.331–349, October–December 2002.

DIA, H.; PURCHASE, H. Modelling the impacts of advanced traveller information systems using Intelligent Agents. **Road and Transport Research**, Vermont South, v.8, n.3, p.68–73, September 1999.

DIJKSTRA, J.; TIMMERMANS, H. J. P. A multi-agent systems approach for visualizing simulated behavior to support the assessment of design performance. In: WORKSHOP ON AGENTS IN TRAFFIC AND TRANSPORTATION, 2000, Barcelona. **Autonmous Agents**. [S.l.: s.n.], 2000.

D'INVERNO, M.; LUCK, M. Engineering AgentSpeak(L): a formal computational model. **Journal of Logic and Computation**, Oxford, v.8, n.3, p.233–260, June 1998.

DROGOUL, A. **De la simulation multi-agent à la résolution collective de problèmes**: une étude de l'émergence de structures d'organisation dans les systèmes multi-agent. 1993. Thèse de Doctorat, Informatique — Université Paris VI, Paris.

DROGOUL, A.; FERBER, J. Multi-agent simulation as a tool for studying emergent processes in societies. In: GILBERT, N.; DORAN, J. (Ed.). **Simulating societies**: the computer simulation of social phenomena. London: UCL Press, 1994.

EL HADOUAJ, S.; ESPIÉ, S.; DROGOUL, A. To combine reactivity and anticipation: the case of conflicts resolution in a simulated road traffic. In: WORKSHOP ON MULTI AGENT BASED SIMULATION, MABS, 2., 2001, Boston. **International Conference on MulitAgent Systems, ICMAS**. [S.l.: s.n.], 2000.

EMERSON, E. A.; SRINIVASAN, J. Branching time temporal logic. In: LINEAR TIME, BRANCHING TIME AND PARTIAL ORDER IN LOGICS AND MODELS FOR CONCURRENCY, 1988, Noordwijkerhout, The Netherlands. **Proceedings...** Berlin: Springer–Verlag, 1989. p.123–172. (Lecture Notes in Computer Science, v.354).

FERNANDES, J. M. A. **MAGOO, metodologia baseada em agentes e orientada por objectos para a resolução de problemas distribuídos**: uma aplicação à gestão de trânsito. 1998. Dissertação (Mestrado em Inteligência Artificial e Computação) — Departamento de Ciência dos Computadores, Faculdade de Ciências da Universidade do Porto, Porto.

FERNANDES, J. M. A.; OLIVEIRA, E. TraMas: traffic control through behaviour-based multi-agent system. In: INTERNATIONAL CONFERENCE ON THE PRACTICAL APPLICATION OF INTELLIGENT AGENTS AND MULTI-AGENT TECHNOLOGY, PAAM, 4., 1999, London. **Proceedings...** [S.l.: s.n.], 1999. p.457–458.

FISCHER, K.; MÜLLER, J. P.; PISCHEL, M. A pragmatic BDI architecture. In: HUHNS, M. N.; SINGH, M. P. (Ed.). **Readings in agents**. San Francisco: Morgan Kaufmann, 1998. p.217–224.

FREITAS, C. M. D. S. **Uma abordagem unificada para análise exploratória interativa visual**. 1994. Tese (Doutorado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

FROZZA, R. **SIMULA**: ambiente para desenvolvimento de sistemas multiagentes reativos. 1997. Dissertação (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

GABRIÉ, T.; HOWDEN, N.; NORLING, E.; TIDHAR, G.; SONENBERG, L. **Multi–agent design of a traffic–flow control system**. Melbourne, Australia: Department of Computer Science, The University of Melbourne, 1994. (Technical Report 94/24).

GARCIA-BULLE, M. **Multiagent network equilibrium with applications to transportation systems**. 1990. PhD Thesis — Stanford University, Stanford, California.

GARRETT, A. Intelligent Transport Systems: potential benefits and immediate issues. **Road and Transport Research**, Vermont South, v.7, n.2, p.61–69, June 1998.

GEORGEFF, M. P.; LANSKY, A. L. Reactive reasoning and planning. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, AAAI, 6., 1987, Seattle, WA. **Proceedings...** Menlo Park: AAAI Press, 1987. p.677–682.

GEORGEFF, M. P.; RAO, A. S. A profile of the Australian AI Institute. **IEEE Expert**, Los Alamitos, v.11, n.6, p.89–92, December 1996.

GIRAFFA, L. M. M. **Uma arquitetura de tutor utilizando estados mentais**. 1999. Tese (Doutorado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

HADDADI, A. A hybrid architecture for multi-agent systems. In: WORKSHOP OF THE SPECIAL INTEREST GROUP ON COOPERATING KNOWLEDGE BASED SYSTEMS, CKBS-SIG, 1993, Keele. **Proceedings...** Keele: DAKE Centre, 1994. p.13–25.

HADDADI, A. **Communication and cooperation in agent systems**: a pragmatic theory. Berlin: Springer–Verlag, 1996. (Lecture Notes in Artificial Intelligence, v.1056).

HALPERN, P. **Emergent behavior in structurally dynamic cellular automata**. Philadelphia, 1997. Available at: ⟨http://www.usip.edu/mpcs/halpern/emer-pap.html⟩. Visited on Oct. 31, 2002.

HAUGENEDER, H.; STEINER, D. A multi-agent approach to cooperation in urban traffic. In: WORKSHOP OF THE SPECIAL INTEREST GROUP ON COOPERATING KNOWLEDGE BASED SYSTEMS, CKBS-SIG, 1993, Keele. **Proceedings...** Keele: DAKE Centre, 1994. p.83–99.

HELLINGA, B. R. Requirements for the calibration of traffic simulation models. In: ANNUAL CONFERENCE OF THE CANADIAN SOCIETY FOR CIVIL ENGINEERING, 1998, Halifax. **Proceedings...** Montréal: CSCE, 1998. p.211–222.

HERNÁNDEZ, J. Z.; OSSOWSKI, S.; GARGÍA-SERRANO, A. Agent-based decision support for traffic management: experiences in a real-world scenario. In: WORKSHOP ON AGENTS IN TRAFFIC AND TRANSPORTATION, 2000, Barcelona. **Autonmous Agents**. [S.l.: s.n.], 2000.

HERTKORN, G.; WAGNER, P. Agent-based traffic simulation. In: WORKSHOP ON AGENT-BASED SIMULATION, 1., 2000, Passau, Germany. **Proceedings...** Ghent: SCS Europe, 2000.

HIDAS, P. How can autonomous agents help microscopic simulation? In: WORKSHOP ON AGENTS IN TRAFFIC AND TRANSPORTATION, 2000, Barcelona. **Autonmous Agents**. [S.l.: s.n.], 2000.

HIDAS, P. Modelling vehicle interactions in microscopic traffic simulation. In: WORKSHOP ON AGENTS IN TRAFFIC AND TRANSPORTATION, 2001, Sydney. **ITS World Congress**. [S.l.: s.n.], 2001.

HINDRIKS, K. V.; BOER, F. S. de; HOEK, W. van der; MEYER, J.-J. C. Formal semantics for an abstract agent programming language. In: INTERNATIONAL WORKSHOP ON AGENT THEORIES, ARCHITECTURES, AND LANGUAGES, ATAL, 4., 1997, Providence, Rhode Island. **Proceedings...** New York: Springer–Verlag, 1998. p.215–229. (Lecture Notes in Computer Science, v.1365).

HINDRIKS, K. V.; BOER, F. S. de; HOEK, W. van der; MEYER, J.-J. C. **A formal embedding of AgentSpeak(L) in 3APL**. Utrecht, The Netherlands: Institute of Information and Computing Sciences, Utrecht University, 1998. (Technical Report UU-CS-1998-07).

HINDRIKS, K. V.; BOER, F. S. de; HOEK, W. van der; MEYER, J.-J. C. A formal embedding of AgentSpeak(L) in 3APL. In: AUSTRALIAN JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, AI, 11., 1998, Brisbane, Australia. **Proceedings. . .** New York: Springer–Verlag, 1998. p.155–166. (Lecture Notes in Computer Science, v.1502).

HUBER, M. J. **UMPRS Agent**. Oceanside, CA: Intelligent Reasoning Systems, 1994. Manual.

HUBER, M. J. JAM: a BDI-theoretic mobile agent architecture. In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS, 3., 1999, Seattle, Washington. **Proceedings. . .** New York: ACM Press, 1999. p.236–243.

HUBER, M. J. Considerations for flexible autonomy within BDI intelligent agent architectures. In: MUSLINER, D.; PELL, B. (Ed.). **Agents with Adjustable Autonomy**. Menlo Park: The AAAI Press, 1999. p.65–72. (AAAI Spring Symposium Series, n.SS-99-06).

HUBER, M. J. **JAM Agents in a Nutshell**. Oceanside, CA: Intelligent Reasoning Systems, 2001. Manual. (Version 0.65+0.76i).

HÜBNER, J. F. **Migração de agentes em sistemas multi-agentes abertos**. 1995. Dissertação (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

HULSE, M. C.; DINGUS, T. A.; BARFIELD, W. Description and applications of advanced traveler information systems. In: BARFIELD, W.; DINGUS, T. A. (Ed.). **Human factors in intelligent transportation systems**. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1998. p.1–30.

INGRAND, F. F.; GEORGEFF, M. P.; RAO, A. S. An architecture for real-time reasoning and system control. **IEEE Expert**, Los Alamitos, v.7, n.6, p.34–44, December 1992.

INTELLIGENT REASONING SYSTEMS. **Research, development, and application of leading-edge architectures and coordination mechanisms for intelligent software agents**. Oceanside, California, 2002. Available at: ⟨http://www.marcush.net/IRS /index.html⟩. Visited on Sept. 3, 2002.

JACKSON, P. G. **Behavioural responses to dynamic route guidance systems**. London: University of London, Centre for Transport Studies, 1994. Working Paper.

JENNINGS, N. R. **Cooperation in industrial multi-agent systems**. Singapore: World Scientific, 1994. (World Scientific Series in Computer Science, v.43).

JIANG, B. Multi-agent simulations for pedestrian crowds. In: EUROPEAN SIMULATION SYMPOSIUM, ESS, 10., 1998, Nottingham, UK. **Proceedings. . .** Ghent: SCS Europe, 1998.

KANOH, H.; NAKAMURA, T. Knowledge based genetic algorithm for dynamic route selection. In: IEE INTERNATIONAL CONFERENCE ON KNOWLEDGE-BASED INTELLIGENT ENGINEERING SYSTEMS AND ALLIED TECHNOLOGIES, KES, 4., 2000, Brighton, UK. **Proceedings. . .** London: IEE, 2000. p.616–619.

KAWAMURA, T.; HASEGAWA, T.; OHSUGA, A.; HONIDEN, S. Bee-gent: bonding and encapsulation enhancement agent framework for development of distributed systems. In: ASIA PACIFIC SOFTWARE ENGINEERING CONFERENCE, APSEC, 6., 1999, Takamatsu, Japan. **Proceedings. . .** Los Alamitos: IEEE Computer Society, 1999. p.260–267.

KENDALL, E. A.; KRISHNA, P. V. M.; PATHAK, C. V.; SURESH, C. B. Patterns of intelligent and mobile agents. In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS, 2., 1998, Minneapolis, Minnesota. **Proceedings. . .** New York: ACM Press, 1998. p.92–99.

KHATTAK, A.; POLYDOROPOULOU, A.; BEN-AKIVA, M. Modeling revealed and stated pretrip travel response to ATIS. In: TRANSPORTATION RESEARCH BOARD ANNUAL MEETING, 75., 1996, Washington, DC. **Proceedings. . .** Washington: TRB, 1996. p.46–54. (Transportation Research Record, v.1537).

KINNY, D. **The distributed multi–agent reasoning system architecture and language specification**. Melbourne: Australian Artificial Intelligence Institute, 1993.

KLÜGL, F. **Multiagentensimulation**: konzepte, werkzeuge, anwendungen. München: Addison-Wesley, 2001.

KLÜGL, F.; PUPPE, F. The multi-agent simulation environment SeSAm. In: WORKSHOP ON SIMULATION IN KNOWLEDGE-BASED SYSTEMS, 1998, Paderborn, Germany. **Proceedings. . .** Paderborn: Universität Paderborn, 1998. p.98–194.

KLÜGL, F.; WAHLE, J.; BAZZAN, A. L. C.; SCHRECKENBERG, M. Towards anticipatory traffic forecast – modelling of route choice behaviour. In: WORKSHOP ON AGENTS IN TRAFFIC AND TRANSPORTATION, 2000, Barcelona. **Autonmous Agents**. [S.l.: s.n.], 2000.

LABIDI, S.; LEJOUAD, W. **De l'intelligence artificielle distribuée aux systèmes multi-agents**. Sophia-Antipolis, France: INRIA, 1993. (Rapport de recherche n.2004).

LEE, J.; HUBER, M. J.; KENNY, P. G.; DURFEE, E. H. UM-PRS: an implementation of the procedural reasoning system for multirobot applications. In: CONFERENCE ON INTELLIGENT ROBOTICS IN FIELD, FACTORY, SERVICE, AND SPACE, CIRFFSS, 1994, Houston, Texas. **Proceedings. . .** Washington: AIAA, 1994. p.842–849.

LINDSTAEDT, E. **Ambiente de simulação visual interativa**. 1995. Dissertação (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

LIU, R. **DRACULA traffic microsimulation model – a user guide**. Leeds, UK: Institute for Transport Studies, 2001. (Technical Note Draft 407).

LIU, R.; VAN VLIET, D.; WATLING, D. P. DRACULA: dynamic route assignment combining user learning and microsimulation. In: PTRC EUROPEAN TRANSPORT FORUM, 23., 1995, Coventry, UK. **Proceedings. . .** London: PTRC, 1995. v.E, p.143–152.

LIU, R.; VAN VLIET, D.; WATLING, D. P. **DRACULA**: a microscopic, day-to-day dynamic framework for modelling traffic networks. Leeds, UK: Institute for Transport

Studies, 1999. (Extract from paper submitted to Transportation Research Part A: Policy and Practice).

LYONS, G. D.; MCDONALD, M. Traveller information and the Internet. **Traffic Engineering and Control**, London, v.39, n.1, p.24–32, 1998.

MACHADO, R.; BORDINI, R. H. Running AgentSpeak(L) Agents on SIM_AGENT. In: INTERNATIONAL WORKSHOP ON AGENT THEORIES, ARCHITECTURES, AND LANGUAGES, ATAL, 8., 2001, Seattle, WA. **Proceedings...** Berlin: Springer, 2002. p.158–174. (Lecture Notes in Computer Science, v.2333).

MAHMASSANI, H. S.; JAYAKRISHNAN, R. System performance and user response under real-time information in a congested traffi corridor. **Transportation Research Part A: Policy and Practice**, Amsterdam, v.25, n.5, p.293–307, 1991.

MAK, V. W. DOSE: a modular and reusable object-oriented simulation environment. In: SCS MULTICONFERENCE ON OBJECT-ORIENTED SIMULATION, 1991, Anaheim, CA. **Proceedings...** San Diego: SCS International, 1991. p.3–11.

MAST, T. Introduction to ITS. In: BARFIELD, W.; DINGUS, T. A. (Ed.). **Human factors in intelligent transportation systems**. Mahwah, NJ: Lawrence Erlbaum Associates, 1998. p.xv–xxi.

MILAM, R. T.; CHOA, F. Recommended guidelines for the calibration and validation of traffic simulation models. In: TRB TRANSPORTATION PLANNING APPLICATIONS CONFERENCE, 8., 2001, Corpus Christi, TX. **Proceedings...** Washington: TRB, 2001.

MIT MEDIA LABORATORY. **Introduction to StarLogo**. Cambridge, MA, 2002. Available at: ⟨http://education.mit.edu/starlogo/⟩. Visited on Oct. 31, 2002.

MÓRA, M. C. **Um modelo formal e executavel de agentes BDI**. 1999. Tese (Doutorado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

MÓRA, M. C.; LOPES, J. G. P.; VICCARI, R. M.; COELHO, H. BDI models and systems: bridging the gap. In: INTERNATIONAL WORKSHOP ON AGENT THEORIES, ARCHITECTURES, AND LANGUAGES, ATAL, 5., 1998, Paris, France. **Proceedings...** Berlin: Springer, 1999. p.11–27. (Lecture Notes in Computer Science, v.1555).

MURCHLAND, J. D. Braess's paradox of traffic flow. **Transportation Research**, Amsterdam, v.4, n.4, p.391–394, December 1970.

NAGEL, K.; SCHRECKENBERG, M. A cellular automaton model for freeway traffic. **Journal de Physique I**, Les Ulis, v.2, n.12, p.2221–2228, December 1992.

NIITTYMÄKI, J. **Fuzzy traffic signal control**: principles and applications. 2002. PhD Thesis — Department of Civil and Environmental Engineering, Helsinki University of Technology, Espoo, Finland.

NIITTYMÄKI, J.; KIKUCHI, S. Application of fuzzy logic to the control of a pedestrian crossing signal. **Transportation Research Record**, Washington, DC, n.1651, p.30–38, 1998.

OLIVEIRA, F. M. Inteligência Artificial Distribuída. In: ESCOLA REGIONAL DE INFORMÁTICA, 4., 1996, Canoas, RS. **Anais. . .** Canoas: SBC–Sul, 1996.

OSSOWSKI, S.; GARCÍA-SERRANO, A. Social structure in artificial agent societies: implications for autonomous problem-solving agents. In: INTERNATIONAL WORKSHOP ON AGENT THEORIES, ARCHITECTURES, AND LANGUAGES, ATAL, 5., 1998, Paris, France. **Proceedings. . .** Berlin: Springer, 1999. p.133–148. (Lecture Notes in Computer Science, v.1555).

PARUNAK, H. V. D.; SAVIT, R.; RIOLO, R. L. Agent-based modeling vs. equation-based modelling a case study and users' guide. In: SICHMAN, J. S.; CONTE, R.; GILBERT, N. (Ed.). **Multi-agent systems and agent-based simulation**. Berlin: Springer, 1998. p.10–25. (Lecture Notes in Artificial Intelligence, v.1534).

PIRES, M.; DIAS, J.; BELO, O. Using multi–agent systems technology on cooperative traffic control simulation: a case study. In: IASTED INTERNATIONAL CONFERENCE ON MODELING, IDENTIFICATION AND CONTROL, 16., 1997, Innsbruck, Austria. **Proceedings. . .** Anaheim: ACTA Press, 1997. p.241–244.

POLYDOROPOULOU, A.; BEN-AKIVA, M.; KHATTAK, A.; LAUPRETE, G. Modeling revealed and stated enroute travel response to ATIS. In: ANNUAL TRANSPORTATION RESEARCH BOARD MEETING, 75., 1996, Washington, DC. **Proceedings. . .** Washington: TRB, 1996. p.38–45. (Transportation Research Record, v.1537).

POLYDOROPOULOU, A.; BEN-AKIVA, M.; KHATTAK, A.; LAUPRETE, G. **Modeling revealed and stated enroute travel response to ATIS**. Berkeley, CA: Institute of Transportation Studies, University of California at Berkeley, 1996. (Research Report UCB-ITS-PRR-96-21).

QUADSTONE LIMITED. **Paramics Online**. Edinburgh, 2002. Available at: ⟨http://www.paramics-online.com/⟩. Visited on March 31, 2002.

RAKHA, H.; HELLINGA, B.; AERDE, M. V.; PEREZ, W. Systematic verification, validation and calibration of traffic simulation models. In: TRANSPORTATION RESEARCH BOARD ANNUAL MEETING, 75., 1996, Washington, DC. **Proceedings. . .** Washington: TRB, 1996.

RAO, A. S. AgentSpeak(L): BDI agents speak out in a logical computable language. In: EUROPEAN WORKSHOP ON MODELLING AUTONOMOUS AGENTS IN A MULTI-AGENT WORLD, MAAMAW, 7., 1996, Eindhoven, The Netherlands. **Proceedings. . .** Berlin: Springer, 1996. p.42–55. (Lecture Notes in Computer Science, v.1038).

RAO, A. S.; GEORGEFF, M. P. Modeling rational agents within a BDI-architecture. In: INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING, KR, 2., 1991, Cambridge, MA. **Proceedings. . .** San Mateo: Morgan Kaufmann, 1991. p.473–484.

RAO, A. S.; GEORGEFF, M. P. BDI agents: from theory to practice. In: INTERNATIONAL CONFERENCE ON MULTIAGENT SYSTEMS, ICMAS, 1., 1995, San Francisco, CA. **Proceedings. . .** Cambridge: MIT Press, 1995. p.312–319.

RASMUSSEN, J. **Information processing and human-machine interaction**: an approach to cognitive engineering. Amsterdam: Elsevier Science, 1986. (North-Holland Series in System Science and Engineering, v.12).

ROSSETTI, R. J. F. **Um ambiente para suporte à simulação de sistemas de tráfego urbano**. 1998. Dissertação (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

ROSSETTI, R. J. F.; BAMPI, S. A software environment proposal to aid Intelligent Transportation System design. In: UFRGS MICROELECTRONICS SEMINAR, SIM, 13., 1998, Bento Gonçalves, RS. **Proceedings...** Porto Alegre: PPGC/UFRGS, 1998. p.181–184.

ROSSETTI, R. J. F.; BAMPI, S. A software environment to integrate urban traffic simulation tasks. In: EUROPEAN SIMULATION SYMPOSIUM, ESS, 10., 1998, Nottingham, UK. **Proceedings...** Ghent: SCS Europe, 1998. p.371–377.

ROSSETTI, R. J. F.; BAMPI, S. A Software environment to integrate urban traffic simulation tasks. **Journal of Geographic Information and Decision Analysis**, Ontario, v.3, n.1, p.56–63, 1999.

ROSSETTI, R. J. F.; BAMPI, S.; LIU, R.; VAN VLIET, D.; CYBIS, H. B. B. Representing drivers' behaviour using an agent-based approach. In: WORKSHOP ON AGENT-BASED SIMULATION, 1., 2000, Passau, Germany. **Proceedings...** Ghent: SCS Europe, 2000. p.45–50.

ROSSETTI, R. J. F.; BAMPI, S.; LIU, R.; VAN VLIET, D.; CYBIS, H. B. B. An agent-based framework for the assessment of drivers' decision-making. In: IEEE ANNUAL CONFERENCE ON INTELLIGENT TRANSPORTATION SYSTEMS, ITSC, 3., 2000, Dearborn, MI. **Proceedings...** Piscataway: IEEE, 2000. p.387–392.

ROSSETTI, R. J. F.; BORDINI, R. H.; BAZZAN, A. L. C.; BAMPI, S. Representing demand as consequence of BDI agents' decision-making. In: WORKSHOP ON AGENTS IN TRAFFIC AND TRANSPORTATION, 2001, Sydney. **ITS World Congress**. [S.l.: s.n.], 2001.

ROSSETTI, R. J. F.; BORDINI, R. H.; BAZZAN, A. L. C.; BAMPI, S.; LIU, R.; VAN VLIET, D. Using BDI agents to improve driver modelling in a commuter scenario. **Transportation Research Part C: Emerging Technologies**, Amsterdam, v.10, n.5–6, p.47–72, November 2002.

ROSSETTI, R. J. F.; LIU, R.; CYBIS, H. B. B.; BAMPI, S. A multi-agent demand model. In: MINI-EURO CONFERENCE AND THE 9TH MEETING OF THE EURO WORKING GROUP TRANSPORTATION, 13., 2002, Bari, Italy. **Proceedings...** Bari: Polytechnic University of Bari, 2002. p.193–198.

ROSSETTI, R. J. F.; LIU, R.; VAN VLIET, D.; BAMPI, S.; CYBIS, H. B. B. Applying an agent-based approach to traffic modelling. In: WORKSHOP ON AGENTS IN TRAFFIC AND TRANSPORTATION, 2000, Barcelona. **Autonmous Agents**. [S.l.: s.n.], 2000.

ROTH, B. H. **A blackboard model of control**. Stanford, CA: Stanford University, 1984. (Technical Report HPP83-38).

RUSSELL, S. J.; NORVIG, P. **Artificial intelligence**: a modern approach. Englewood Cliffs, NJ: Prentice-Hall, 1995.

SAAD, F.; SCHNETZLER, B. **Un modèle conceptuel du comportement des conducteurs en file sur autoroute**. Arcueil, France: Laboratoire de Psychologie de la Conduite, INRETS, 1994. Rapport Interne.

SCHACHTE, P. **A gentle introduction to static analysis and logic programming**. Victoria, Australia, 2002. Available at: ⟨http://www.cs.mu.oz.au/∼schachte/lpanalysis.html⟩. Visited on Sept. 4, 2002.

SCHLEIFFER, R. Traffic itself is simple – just analyzing it is not. In: ANNUAL HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, HICSS, 33., 2000, Maui, Hawaii. **Proceedings...** Los Alamitos: IEEE Computer Society, 2000.

SCHRIBER, T. J. **An Introduction to Simulation Using GPSS/H**. New York, NY: John Wiley and Sons, 1990.

SHANNON, R. E. **System simulation – the art and science**. Englewood Cliffs, NJ: Prentice Hall, 1975.

SHANNON, R. E. Introduction to simulation. In: WINTER SIMULATION CONFERENCE, WSC, 1992, Arlington, VA. **Proceedings...** New York: ACM, 1992. p.65–73.

SHEFFI, Y. **Urban transportation networks**: equilibrium analysis with mathematical programming methods. Englewood Cliffs, NJ: Prentice-Hall, 1985.

SHOHAM, Y. **Agent-oriented programming**. Stanford, CA: Computer Science Department, Stanford University, 1990. (Technical Report STAN-CS-1335-90).

SHOHAM, Y. Agent–oriented programming. **Artificial Intelligence**, Amsterdam, v.60, n.1, p.51–92, March 1993.

SICHMAN, J.; DEMAZEAU, Y.; BOISSIER, O. When can knowledge-based systems be called agents? In: BRAZILIAN SYMPOSIUM ON ARTIFICIAL INTELLIGENCE, SBIA, 9., 1992, Rio de Janeiro. **Proceedings...** [S.l.: s.n.], 1992. p.172–185.

SIMON, H. A. Rational choice and the structure of the environment. **Psychological Review**, Washington, DC, v.63, n.2, p.129–138, March 1956.

SLOMAN, A. **Teach primer – an overview of POP-11**. Birmingham, UK: School of Computer Science, The University of Birmingham, 1999.

SLOMAN, A.; POLI, R. SIM_AGENT: a toolkit for exploring agent designs. In: INTERNATIONAL WORKSHOP ON AGENT THEORIES, ARCHITECTURES, AND LANGUAGES, ATAL, 2., 1995, Montreal, Canada. **Proceedings...** Berlin: Springer–Verlag, 1996. p.392–407. (Lecture Notes in Computer Science, v.1037).

SMITH III, A. R. **Cellular automata theory**. Stanford, CA: Digital Systems Laboratory, Stanford Electronics Laboratories, Stanford University, 1969. (Technical Report n.2).

SMITH, R. G. The contract net protocol: high-level communication and control in a distributed problem solver. In: BOND, A. H.; GASSER, L. (Ed.). **Readings in distributed artificial intelligence**. San Mateo: Morgan Kaufmann, 1988. p.357–366.

STEELS, L. Cooperating between distributed agents through self-organisation. In: DEMAZEAU, Y.; MULLER, J. P. (Ed.). **Decentralized A.I.** Amsterdam: North-Holland, 1990. p.175–196.

STEINER, D.; BURT, A. D.; KOLB, M.; LERI, C. The conceptual framework of MAI^{2}L. In: EUROPEAN WORKSHOP ON MODELLING AUTONOMOUS AGENSTS IN A MULTI-AGENT WORLD, MAAMAW, 5., 1993, Neuchatel, Switzerland. **Proceedings...** Berlin: Springer, 1995. p.217–230. (Lecture Notes in Computer Science, v.957).

TANDAYYA, P.; ZOBEL, R. Agent-based simulation in DIS civil systems. In: WORKSHOP ON AGENT-BASED SIMULATION, 1., 2000, Passau, Germany. **Proceedings...** Ghent: SCS Europe, 2000. p.199–204.

TANDAYYA, P.; ZOBEL, R. The instant model for a DIS road traffic simulation system. In: EUROPEAN SIMULATION MULTICONFERENCE, ESM, 14., 2000, Ghent, Belgium. **Proceedings...** Ghent: SCS Europe, 2000. p.411–415.

TEDESCO, P. A.; SELF, J. A. **MArCO**: using conflict mediation strategies to support group planning interactions. Leeds, UK: Computer Based Learning Unit, University of Leeds, 2000. (Technical Report 00/4).

THOMAS, R. **PLACA, an agent oriented programming language**. 1993. PhD Thesis — Computer Science Department, Stanford University, Stanford, CA.

TOKORO, M. Agents: towards a society in which humans and computers cohabitate. In: EUROPEAN WORKSHOP ON MODELLING AUTONOMOUS AGENTS IN A MULTI-AGENT WORLD, MAAMAW, 6., 1994, Odense, Denmark. **Proceedings...** Berlin: Springer, 1996. p.1–10. (Lecture Notes in Computer Science, v.1069).

TOSHIBA CORPORATION. **Tutorial for Bee-gent**. Tokyo, Japan, 1999. Tutorial.

VAN KATWIJK, R. T. Co-ordination of traffic management instruments using agent technology. In: WORKSHOP ON AGENTS IN TRAFFIC AND TRANSPORTATION, 2000, Barcelona. **Autonmous Agents**. [S.l.: s.n.], 2000.

VAN VLIET, D. D'Esopo: a forgotten tree-building algorithm. **Traffic Engineering and Control**, London, v.18, n.7–8, p.372–373, July/August 1977.

WATLING, D. P. Urban traffic network models and dynamic driver information systems. **Transport Reviews**, Abingdon, v.14, n.3, p.219–246, 1994.

WATLING, D. P. **DRACULA 1.0**: user guide to the day-to-day model. Leeds, UK: Institute for Transport Studies, 1995. (Technical Note Draft 369).

WEISS, G. Adaptation and learning in multi-agent systems: some remarks and a bibliography. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, IJCAI, 14., 1995, Montréal, Canada. **Proceedings...** Berlin: Springer, 1996. p.1–21. (Lecture Notes in Computer Science, v.1042).

WEISS, G. Learning in multiagent systems. In: WEISS, G. (Ed.). **Multiagent systems**: a modern approach to distributed artificial intelligence. Cambridge, MA: The MIT Press, 1999. p.259–298.

WERNER, E. The design of multi-agent systems. In: EUROPEAN WORKSHOP ON MODELLING AUTONOMOUS AGENTS IN A MULTI-AGENT WORLD, MAAMAW, 3., 1991, Kaiserslautern, Germany. **Proceedings...** Amsterdam: North-Holland, 1992. p.3–28.

WOOLDRIDGE, M. Intelligent Agents. In: WEISS, G. (Ed.). **Multiagent systems**: a modern approach to distributed artificial intelligence. Cambridge, MA: The MIT Press, 1999. p.27–78.

WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent agents: theory and practice. **The Knowledge Engineering Review**, Cambridge, v.10, n.2, p.115–152, 1995.

WU, J.; MCDONALD, M.; BRACKSTONE, M. A fuzzy logic microscopic simulation model for interurban ATT assessment. In: EUROPEAN SIMULATION SYMPOSIUM, ESS, 10., 1998, Nottingham, UK. **Proceedings...** Ghent: SCS Europe, 1998.