

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Simulating Drivers' Decision-making Under Information Dissemination

Manuel Guilherme Figueiredo Soares



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Rosaldo Rossetti PhD (Assistant Professor)

Second Supervisor: Zafeiris Kokkinogenis

March 4, 2013

Simulating Drivers' Decision-making Under Information Dissemination

Manuel Guilherme Figueiredo Soares

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Doctor António Augusto de Sousa

External Examiner: Doctor Paulo Jorge Pinto Leitão

Supervisor: Zafeiris Kokkinogenis

March 4, 2013

Abstract

The rapid and ever-increasing population and urban activities have imposed a massive demand on Urban Transportation Systems (UTS). These systems were not prepared for such events, so traffic congestion and defective metropolitan systems were a direct consequence of this deficiency.

Nowadays, accurate and up-to-date traffic information is indeed very important for travellers to arrive faster and safely to their destinations. It is in this field that the Advanced Traveller Information Systems (ATIS) stand out in the Intelligent Transportation Systems (ITS). Taking advantage not only of new Web services and wireless communication technologies, but also of new pervasive computing paradigms, these systems provide valuable information before, during and even after an activity has been performed. ATIS collects surrounding information and generates knowledge, from its raw data, which will be then broadcast to travellers according to individual needs.

This information is used by the traveller in his decision-making process, as it increases his level of cognition: incident reports, weather conditions, optimal routes, among others. In addition to these advantages travellers can also benefit in terms of consumption, congestion, reduced environmental impacts, and prevent stress and hazards.

Designing a new information system to improve traffic solutions is a very difficult task, especially when the system involves socio-technical aspects, to which Simulation proved to be an effective approach to analysing and designing novel traffic solutions.

The aim of the thesis is to study the effects that knowledge dissemination by the ATIS can have in the drivers' decision making process. To gain insight into this process, a simulation of an artificial society of drivers will be devised and emergent mobility patterns will be observed and analysed. This work integrates the JADE (Java Agent DEvelopment) framework and SUMO (Simulation of Urban MObility) to build an artificial laboratory for simulating a society of driver agents and ATIS infrastructures immersed in an urban traffic scenario.

A scenario has been set up to provide a test-bed where an artificial society of driver agents and information infrastructures are defined. In this scenario, it is studied how ubiquitous knowledge influences the planning activities of each individual, and how an ATIS solution improves the system's throughput. Results have shown that indeed travellers' stopped time in a congestion situation was improved indicating that accurate and real-time information is in fact very important for the drivers' welfare.

Resumo

O rápido e constante crescimento das populações nos centros urbanos impôs um grande aumento na procura nos Sistemas Urbanos de Transporte (UTS). Estes sistemas não se encontravam preparados para tal revolução, por isso, os congestionamentos de tráfego e sistemas metropolitanos pouco eficientes foram uma consequência directa desta situação.

Hoje em dia, informação de tráfego precisa e em tempo real é muito importante para os viajantes, possibilitando-lhes chegar mais rápido e em segurança aos seus destinos. É neste âmbito que os Sistemas Avançados de Informação ao Viajante (ATIS) se inserem nos Sistemas Inteligentes de Transportes (ITS). Tirando partido não só das novas tecnologias web, e comunicações wireless, mas também dos novos paradigmas universais da computação, estes sistemas providenciam informação valiosa antes, durante e depois de qualquer actividade. Os ATIS recolhem vários tipos de informação e geram conhecimento a partir desses dados percebidos, que por sua vez vão ser transmitidos aos viajantes de acordo com as suas necessidades individuais.

Esta informação é usada pelo viajante no seu processo decisório, já que aumenta o seu estado de conhecimento, indicando: detalhes de acidentes, condições meteorológicas rotas óptimas, entre outros. Para além disto, os viajantes usufruem também de informações relativas a consumos, e congestionamentos, que levam a uma redução do impacto ambiental, situações de stress e perigos.

Projectar um novo sistema de informação para melhorar soluções de tráfego é uma tarefa bastante árdua, principalmente quando estão envolvidas aspectos tecno-sociais, para os quais a Simulação provou ser uma abordagem bastante eficiente na análise e desenvolvimento de novas soluções de tráfego.

O objectivo desta tese é estudar os efeitos que a disseminação de informação pelos ATIS pode ter no processo decisório dos condutores. Para atingir este objectivo a simulação de uma sociedade artificial de condutores vai ser implementada e padrões de mobilidade serão observados e analisados. Este trabalho integra a plataforma para desenvolvimento de Sistemas Multi-Agente JADE (Java Agent DEvelopment) com o simulador de tráfego microscópico SUMO (Simulation of Urban MObility), de forma a produzir um laboratório artificial para simular uma sociedade de condutores e infraestruturas ATIS imersos num cenário de tráfego urbano.

Foi construído um cenário de teste que serviu para apresentar e definir a sociedade artificial de condutores e as infraestruturas de informação. Neste cenário, é estudada a influência da informação no planeamento de cada indivíduo e como os ATIS podem ajudar na melhoria do rendimento do sistema. Os resultados mostraram que, de facto, o tempo que os condutores passam parados devido a congestionamentos foi melhorado com a influencia dos ATIS, o que evidencia a importância que informação actual e precisa tem no bem-estar dos condutores.

Acknowledgements

First of all I would like to thank to my friends and supervisors Rosaldo Rossetti and Zafeiris Kokkinogenis for all their help, supportive and pushing attitude since the beginning of this project, a year ago. They have always rested enthusiasm and excitement in me and in this thesis, which instilled me with a valuable ambition throughout this dissertation.

To my parents, sister and beloved girlfriend I would like to thank their patience and endless support since the day one.

To all my friends that were always there to cheer me up (and, kindly, down - it's their job), but especially to José Macedo, with whom I discussed, complained, laughed and bypassed some issues along the way.

Moreover I would like to thank the welcomeness to all the Artificial Intelligence and Computer Science Laboratory (LIACC) members, particularly to Professor Eugénio Oliveira.

Personally I would like to recall and thank Professor Raul Vidal for lecturing me and my colleagues on MIEIC, still in my high school, planting a growing seed that blossomed in these years of work.

And last but not least I want to thank to Professor Augusto de Sousa for his availability, kindness, positive energy and excellent humour, which made me feel several times in a family environment.

Guilherme Soares

*“Give a man a fish and you feed him for a day.
Teach a man how to fish and you feed him for a lifetime.”*

Ancient Chinese Proverb

Contents

1	Introduction	1
1.1	Context	1
1.2	Motivation and Objectives	2
1.3	Document Outline	3
2	Literature Review	5
2.1	Intelligent Transportation Systems	5
2.2	Future Urban Transports	8
2.3	Agent and Multi-Agent Systems Overview	9
2.3.1	Agent Architectures	11
2.3.2	Environment in Multi-Agent Systems	17
2.3.3	MAS deployment in Traffic Domain	19
2.4	Modelling & Simulation	21
2.4.1	Synergy Simulation & Agent Technology	21
2.4.2	Agents as a metaphor in M&S	22
2.5	Artificial Societies	23
2.6	Summary	25
3	Methodological Approach	27
3.1	Requirements Analysis	27
3.2	Proposed Design Solution	28
3.2.1	Software Selection	29
3.2.2	Proposed Architecture	29
3.3	Development Milestones	30
3.4	Summary	31
4	Development Software Overview	33
4.1	SUMO	33
4.1.1	Simulation of Urban Mobility	33
4.1.2	TraCI	35
4.2	JADE	36
4.3	TraSMAPI	38
4.4	Summary	40
5	Implementation	43
5.1	Driver Agent Development	43
5.2	ATIS Artefacts Development	47
5.3	Summary	49

CONTENTS

6 Experiments & Results Discussion	51
6.1 Braess Paradox Experimental Set-up	51
6.1.1 200 Vehicles Test	53
6.1.2 1900 Vehicles Test	54
6.2 En-Route Diversion Experimental Set-up	55
6.3 Summary	58
7 Conclusions & Future Work	59
7.1 Final Remarks	59
7.2 Contributions	60
7.3 Future Developments	61
7.4 Future Works	61
References	63

List of Figures

2.1	Intelligent Transportation Systems Modules	6
2.2	ITS's entities and their connections	7
2.3	Advanced Travellers Information Systems	8
2.4	Architectural frame of main agent characteristics	12
2.5	Simple Reflex Agent diagram	13
2.6	Reflex Agent with Internal State diagram	14
2.7	Goal Based Agent diagram	15
2.8	Utility Based Agent diagram	16
2.9	Learning Agent diagram	17
2.10	Agent Directed Simulation	22
2.11	The Coleman boat	24
3.1	ATIS and Society of Drivers synergy example	27
3.2	General proposed solution	28
3.3	Framework Architecture	29
4.1	SUMO's screenshot	34
4.2	TraCI's system architecture	35
4.3	JADE Middleware	36
4.4	JADE Architecture	37
4.5	JADE Graphical User Interface	38
4.6	TraSMAPI's general architecture	38
4.7	TraSMAPI usage in several microscopic traffic simulators	39
4.8	TraSMAPI's envisioned general architecture	40
5.1	Driver's Layers	44
5.2	Driver-Vehicle Association	45
5.3	Vehicle entity abstraction	46
5.4	Induced Traffic Jam	46
5.5	ATIS Interactions with Environment and Drivers	47
5.6	Implemented ATIS entities	48
5.7	Simplified class Diagram for all ATIS Artefacts	49
6.1	Two route network	52
6.2	Three route network	52
6.3	Experiment's States & Actions	52
6.4	Network configuration through time	53
6.5	200 Vehicles Test	54
6.6	1900 Vehicles Test	55

LIST OF FIGURES

6.7	ATIS Artefact in Action	56
6.8	ATIS Communication Sequence Diagram	56
6.9	En-route Diversion with simulated accident	57

List of Tables

2.1	Multiagent systems' attributes	18
2.2	Agent's applications in traffic systems	20
4.1	TraCI's general API	36
5.1	TraCI4J tool analysis	44
5.2	Some Vehicle-directed TraCI Commands	45
6.1	Experiment's Q-Table	52

LIST OF TABLES

Abbreviations

ABM	Agent-Based Modelling
ABMS	Agent-Based Modelling and Simulation
ABS	Agent-Based Simulation
ADS	Agent Directed Simulation
AMS	Agent Management System
API	Application Programming Interface
APTS	Advanced Public Transportation Systems
ARTS	Advanced Rural Transportation Systems
ARTSS	Autonomic Road Transport Support Systems
AS	Artificial Society
ATIS	Advanced Traveller Information Systems
ATMS	Advanced Traffic Management Systems
AVCS	Advanced Vehicle Control Systems
ATS	Artificial Transportation Systems
BDI	Belief Desire Intention
CVO	Commercial Vehicle Operations
DP	Dynamic Programming
DUA	Dynamic User Assignment
FIPA	Foundation for Intelligent Physical Agents
FUT	Future Urban Transport
GUI	Graphical User Interface
GPL	General Public License
ITS	Intelligent Transportation Systems
ITSUMO	Intelligent Transportation System for Urban Mobility
JADE	Java Agent DEvelopment
JVM	Java Virtual Machine
M&S	Modelling and Simulation
MAS	Multi-Agent System
NN	Neural Networks
OS	Operative Systems
OSM	OpenStreetMaps
POI	Points-of-Interest
RL	Reinforcement Learning
SL	Supervised Learning

ABBREVIATIONS

SUMO	Simulation of Urban MObility
TraCI	Traffic Control Interface
TraSMAPI	Traffic Simulation Manager Application Programming Interface
UTC	Urban Traffic Control
UTS	Urban Transportation Systems
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Vehicle and Infrastructure
V2G	Vehicle-to-Grid
VANETs	Vehicular Ad-hoc NETworks
VMS	Variable Message Signs

Chapter 1

Introduction

This chapter contains a section of brief contextualization of the problem of urban traffic congestion and some used technologies to counter it, a section of motivation and objectives where are presented the goals that guided this thesis, and finally a section where this document's structure will be detailed.

1.1 Context

The rapid and ever-increasing population and urban activities has imposed a massive demand on urban transportation systems. The main problem is that most of the urban areas weren't prepared for such hasty development which led to weak and defective metropolitan transportation systems [Rep].

Efficient transportation systems are crucial to an industrialized society and being its main communication infrastructure, therefore rapid and effective solutions for traffic congestion are needed to prevent its negative impact in the city's social and economic welfare. A way to address this issue is resorting to the use of modelling & simulation.

Transportation domain presents an inherent complexity. It involves diverse heterogeneous entities either in structure or in behaviour, e.g. vehicles, pedestrians, among others, that express interactions, reflecting social behaviours that go from collaboration to competition. Moreover, the transportation domain presents a high degree of stochasticity and dynamicity especially when considered in an urban context. Thereby, using simulation and taking advantage of its characteristics (e.g. time compressing) we can test several management solutions or even changes in the network more cheaply and faster. Such approach can provide us with the possibility of comparing studies between new infrastructures' designs or control algorithms without having to interfere in the real world.

Taking advantage of this simulation technologies a new generation of mobility systems, Intelligent Transportation Systems (ITS), arose and could be implemented and polished before being applied [MW06]. The ITS arise as the synergy between the Information and Communication Technologies and the Urban Transportation Systems, which include vehicles and networks that

move people and goods. Traditionally, Mathematical equations describe the drivers and pedestrians movements taking into account several flow restraints, is used to tackle traffic related issues and to model them. According to this approach, the traffic problem was handled as a whole, and the solution was a product of the fulfilment of all trips.

One of the crucial aspects in the concept & development of ITS solutions is the importance of Traffic Information. The development in wireless communication networks and web services enabled the development of the Advanced Traveller Information Systems (ATIS). These systems aim at providing travellers traffic information, which will help them make their pre-trip or en-route decisions, targeting to a quicker, safer and efficient trip plan [CMV06, ZWSF07].

The formalization of the ITS concept is to be considered a great achievement by the transportation engineering community of practitioners and researchers. However, in the last few years the traffic and transportation domain has made a breakthrough in the way is conceived. The explosion of the computing technology in terms of applications we are experimenting the last decade brought together expertise from different scientific and technical disciplines giving birth to new computing and communication paradigms. New type of systems called socio-technical arose from such mutual conjunctions where people and technology live in symbiosis. The transportation and, generally speaking urban domain, could not be impermeable to such revolution. Indeed proves to be a valid test-bed where such new social and technological paradigms can be applied.

A new concept has been coined to deal with this revolution, Future Urban Transport (FUT) Systems. Within FUT the notion of mobility system overcomes the ITS limitations, instead of focusing only the simple processes of transportation of good and persons it becomes more conscious in terms of environment, accessibility, equality, security, and sustainability of resources [PRK11]. People are placed as a central aspect, as well their preferences, of the urban systems, forcing architectures to become more adaptable and accessible to their needs. Therefore, new technologies and methodologies are necessary to track these new models.

1.2 Motivation and Objectives

Normally, in traffic solutions development, the use of a simulator is very straightforward related to traffic flow and junction management. Albeit many attempts and published papers, the solutions presented do not make full use of the concept of intelligent agents.

Additionally, multi-agent systems approach has become recognized as a useful approach for modelling and simulating complex systems [MT07].

Keeping the above mentioned revolution in urban transportation in mind and guided by the need to design more human-centric economic and environmental solutions, a framework that generates an urban context, meaning a traffic network and its population, is necessary so that analysts and designer can study, develop and evaluate their policies and strategies.

The aim of this thesis is to provide community of researchers and practitioners with a tool that can instantiate a heterogeneous Artificial Society (AS) of drivers immersed into a realistic traffic environment. Experimentations with such AS can help us study emergent mobility patterns and

how information or knowledge can affect drivers' decision-making process. The concept of AS can be used by traffic managers or government institutions as a test-bed for strategies or policies analysis towards social awareness use of resources .

There are other scenarios that can be explored with this tool, like Vehicle-to-Vehicle (V2V) or Vehicle-to-Infrastructure (V2X) applications, demand and supply analysis in Vehicle-to-Grid (V2G) networks, or even ubiquitous transportation services & ambient intelligence.

In order to achieve the coupling of AS with traffic urban environment we will devise an integration of a microscopic traffic simulator for large-scale networks with an agent development platform framework, in order to support all the necessary elements of the Artificial Society of Drivers.

1.3 Document Outline

This thesis is structured as follows:

- In this chapter, an introduction to the subject of work and contextualization is made, presenting also the main goals and motivations to the thesis fulfilment;
- In Chapter 2, a literature review is made, introducing background concepts as well some related works;
- In Chapter 3, we gather the system requirements and propose an architectural design;
- Chapter 4, introduces the software for the development, highlighting its utility to our goals;
- Chapter 5, describes in detail the implementation procedure;
- In Chapter 6, we present the various experimental set-ups analysing the gathered data;
- Chapter 7 summarises the information gathered, namely the expected outcomes and contributions of the present thesis, concluding the document depicting the future work.

Introduction

Chapter 2

Literature Review

In this chapter, we explore some concepts that are necessary for a better understanding of this project, as well as a discussion of some existing related works, highlighting its benefits and shortcomings.

First, we present a general background on some topics, where is exploited the notion of Agent and Multi-Agent Systems (MAS) in traffic domain, Agent-Based Modelling (ABM) and lastly Artificial Transportation Systems (ATS).

2.1 Intelligent Transportation Systems

Transportation is an indispensable activity in an urban area's activities. It has turned into an essential part of any economy based, since it is the main infrastructure for people and goods trade.

Transport systems are intended to take both people and other goods from certain points to one or more destinations fulfilling their needs in efficiency and costs. These systems have become rather complex and extremely large, which led to their geographic and functional distribution.

Communication networks, digital systems and computers played a major revolution in the transportation systems, improving efficiency and allowing low costs.

Centralised traffic coordination and planning started playing a major role in network design and management. Although centralised decision systems can be very efficient in theory, their performance is quite dependent on scale. In other words, their efficiency is related to the number of unities processed by the system, becoming slower and therefore lacking productivity. As transportation systems are becoming very large, both in terms of structure and dimension. The whole process of acquiring information from all sources, and providing adequate responses timely is a very arduous task, which is further worsened with real-time constraints and the presence of heterogeneous entities.

In the mid-80's the ITS concept arose when a United States (US) group of transportation professionals recognized the impact that the computing and communications revolutions of the

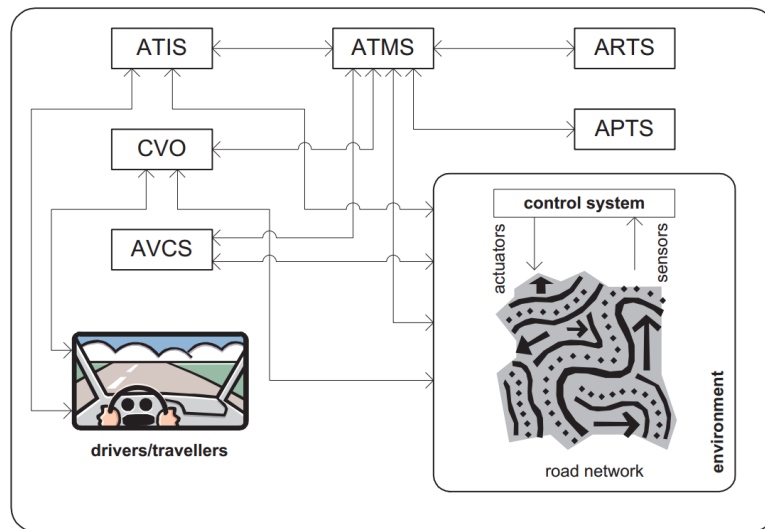


Figure 2.1: Intelligent Transportation Systems Modules [Ros]

Information Age could have on urban transportation. So, instead of focusing on physical intervention, it aims to create intelligent systems that are able to communicate and cooperate toward the improvement capacity usage just making better use of existing resources [HDB98].

ITS use a distributed perspective to divide specific issues of traffic and transportation, proposing modules to treat them. These modules, working together, exchanging knowledge, seek to maximize the overall efficiency [BCSH89]. According to Mast [Mas98], a basis structure for ITS would comprise the following modules, whose interactions are illustrated in Figure 2.1.

- **Advanced Traveller Information Systems (ATIS)** - provide a set of information for drivers, such as navigation, route guidance, and hazard warning, all this adapted to user's necessities;
- **Advanced Vehicle Control Systems (AVCS)** - these mechanisms assists in driving tasks, especially in dangerous and uncommon situations. Autonomous driving is future vision of these systems;
- **Commercial Vehicle Operations (CVO)** - address special needs from commercial transportation, having services as vehicle identification, loading tracking, record keeping, and others. This part aims to decrease the high costs from commercial sector;
- **Advanced Traffic Management Systems (ATMS)** - they play an important role in ITS, monitoring, controlling, and managing traffic in every road level. Some techniques as automated traffic signal timing, variable message signs (VMS), and virtual traffic lights, can be used for reduce congestions;
- **Advanced Rural Transportation Systems (ARTS)** - focus on traffic and transportation in rural areas, which has some special features. These are critical systems in large countries that can help countryside communities;

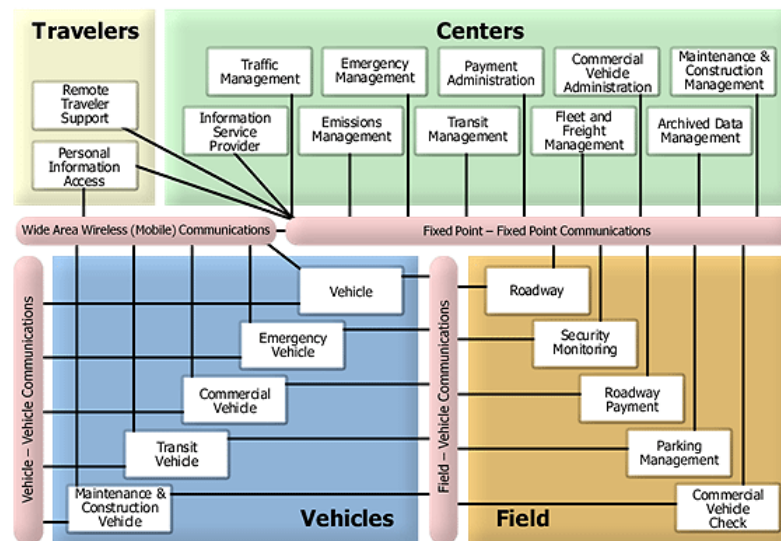


Figure 2.2: ITS's entities and their connections¹

- **Advanced Public Transportation Systems (APTS)** - enhance effectiveness, economics, and, consequently, attractiveness of public transportation. Also, count with fleet management and real-time system information.

Communication plays an important role in ITS and it exists between all entities known from an Intelligent Transportation System's architecture, as we can see in Figure 2.2¹, and Vehicle-vehicle (V2V) and vehicle-infrastructure (V2I) communications are extensively treated in the literature. Back to Figure 2.2, we can observe the ITS's main subsystems. The travellers represent pedestrians, but also people inside vehicle, for instance a person inside a public transport or drivers. Centres are the controller stations of all services, the field is the infrastructure of traffic and transportation system and lastly the vehicles that cover all vehicle types e.g. emergency vehicles or personal ones.

For the purpose of this thesis we will only consider the subsystem from the holistic view of ITS related to the Traveller Information, that is, **ATIS**.

Advanced Travellers Information Systems

Advanced Travellers Information Systems (ATIS) play an important role in intelligent transportation systems. It assists travellers with pre-trip and en route travel information to improve the convenience, safety and efficiency of the trip.

In a highly mobile society, precise and well-timed traffic information can help travellers reach their destinations quickly and safely. To serve this information need, Advanced Traveller Information Systems (ATIS) are introduced to help drivers avoid congestion and choose efficient and safer routes.

¹High level architecture diagram description, 2010. Available in http://www.its.dot.gov/arch/arch_longdesc.htm

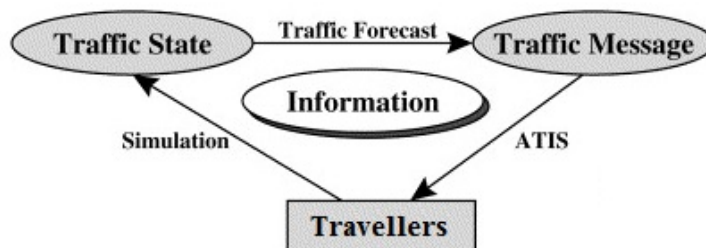


Figure 2.3: Advanced Travellers Information Systems (adapted from [WBKS00])

Initially ATIS was designed to assist travellers making pre-trip travel planning. With the rapid progress in the development of computer and communication technologies, traveller information will not only be able to benefit travellers, but also can be utilized by service providers. We can expect more enriched and better traveller information will be available, and ATIS providing en route information will also get popular soon.

2.2 Future Urban Transports

With the late revolution in transportation systems the user is a central aspect of transportation systems, forcing architectures to become adaptable and accessible by different means so as to meet different requirements and a wide range of purposes.

This novel scenario has been motivating and challenging practitioners and the scientific community and suggested a completely new decentralised perspective of processes. On the other hand, discussions are still fostered by current ambitions to the Future Urban Transport (FUT) systems, even more conscious in terms of environment, accessibility, equality, security, and sustainability of resources. Some of the main features of today's intelligent transportation, to mention a few, are as follows:

- **Automated Computation** - Future transport systems must make decisions automatically, analysing input information and acting accordingly, triggering coordinated actions to improve system performance;
- **Flexibility and Freedom** - the lack of flexibility in transportation systems limits their potential to users. FUT should provide flexibility, different options and driver choices, as well as personalised services;
- **Accuracy** - precise and up-to-date information must be delivered in useful time. In transport systems, which are generally classified as dynamic and stochastic, errors and other failure situations are proportional to the accuracy of system reactions and responses to requested services;

- **Intelligent Infrastructures** - transportation systems are greatly dependent on the network topology and infrastructures. New communication technologies, including mobile, wireless and ad-hoc networks are improving infrastructures;
- **Distributed Architecture** - There are several requirements that must be satisfied, from user-centred to service-based functionalities, turning intelligent transportation into a complex, heterogeneous and sophisticated **artificial society**.

Accounting for the characteristics above-mentioned, multi-agent systems have been greatly studied and applied in developing the concept of future transportation. Especially in the past couple of decades, a great deal of research work has been carried out in this area. Besides MAS, other concepts are also being devised and improved e.g. such as ubiquitous computing, ambient intelligence and service-oriented architectures.

Today's technology already allows us to experience new properties and services that enhance traditional transportation. For instance, mobile communication allows a user to access and consult traffic conditions while he is still on the way to his vehicle. Also, pedestrians may use multi-modal travel planners to find better itineraries, i.e. including various times of transports and also walking paths. They can select the best alternatives according to specific requirements, such as cost, expected travel time, or presence of given points of interest (POI), like, museums, restaurants and pharmacies [ZA08].

As we can see, some examples are already in use, however much of the expected potentials of such systems are still utopias. Nevertheless, with today's technologies they are quite feasible:

- **emergency situations** - emergency vehicles, e.g. fire trucks, need to get to a particular place in a congested urban area, and with intelligent control systems, which are aware of the emergency, collaborate by setting all traffic lights across the due itinerary to green [Mit84];
- **itinerary planning** - as soon as a driver gets in the vehicle, the system prepares the best itinerary according to the driver's to-do list or agenda;
- **ambient intelligence** - accident handling. In the event of an accident, the environment should be able to perceive and while calling the emergency services, could redirect other vehicles to avoid the accident site [JS03].

2.3 Agent and Multi-Agent Systems Overview

In this section, we aim to give a broad view of MAS and focus on its application on transportation system.

The definition of **Agent** stimulates a very interesting discussion in research community. Nevertheless, this interest in finding a formal definition decreased and emphasis is placed on application domains of the agent paradigm.

Literature Review

Tolk and Uhrmacher [TU09] present a brief, yet interesting review over the agent definitions found in the literature. Among these definitions, is the one provided by Russell and Norvig ² suggesting:

" agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors "

Other definitions take the type of environment into account, that is, whether it is dynamic and complex or assuming some autonomy of the agent.

" autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed." [Mae95]

Frezza [Fro97] indicates some features related to the level of perceptual, cognitive and social skills which differentiate an agent from a simple object:

- the capability to perceive other agents as well the environment, and actuate according to its seeking goal;
- the communication, adaptability and autonomy, which are fundamental for social integration;

In the latter we are faced with probably the main feature that differentiates an agent from a simple computer program: **autonomy**.

In fact, we find this feature in several definitions. Mostly it is explicitly stated as in Frezza's statement or in Shoham work [Sho93], but sometimes, authors' definitions imply autonomy, like in Russell and Norvig's one where autonomy is implicit.

Nevertheless, one of the most wide-ranging and cited definition of agent [NRW04, FG97, WJO95], is the one given by Wooldridge and Jennings [WJ95] in which an agent is:

"a hardware or (more usually) a software-based computer system that enjoys the following properties:

- **autonomy** - agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state;
- **social ability** - agents interact with other agents (and possibly humans) via some kind of agent-communication language;
- **reactivity** - agents perceive their environment (...) and respond in a timely fashion to changes that occur in it;
- **pro-activeness** - agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking initiative."

²Russell, S. J. and P. Norvig (2009). Artificial Intelligence: A Modern Approach, 3rd Edition

In short, there isn't a single definition of an intelligent agent; there are several characteristics that can be captured in the form of taxonomies that help to contribute to a consensus, among practitioners.

Although some researchers find an agent to be describable as an independent and isolated entity, we find the **agent's social ability** a **very important** characteristic, since interaction with other autonomous intelligent entities, i.e. agents, will enrich the one's knowledge.

Furthermore, we need to emphasize this agent's feature, as we want to build a **society** of agents, where direct or indirect interactions are crucial in social complex systems, as is the driver society in traffic domains, therefore the **Multi-Agent System** (MAS) paradigm.

These systems are characterized by asynchronous computing and no global system control. Agents have a limited point of view and data is decentralized. The application domains for agent populations are many and reach from simulation of populations and the influence of individual behaviour in complex system analyses to agent mediated decision support systems [MT07].

2.3.1 Agent Architectures

The internal architectures of agents are subdivided into two main approaches: the first approach is a reactive architecture, where it is only considered a set of perception-action pairs; the second one is centred in a more cognitive based on the agent's representation of the environment [UW09].

An agent model can reach different levels of complexity and functionality, going from a simple reactive to a more complex cognitive structure.

Moya & Tolk [MT07] propose an architectural frame in Figure 2.4 addressing main agent characteristic. Its intention is to be a guideline for system engineers, simulation and agent developers. It is neither complete or exclusive. The authors identify three external and four internal architectural domains/areas. The external areas comprise those functions needed within an agent to interact with its environment:

- **Perception** functions watch the environment. Using its sensors, the agent receives stimuli (signals) from its environment and sends this information to the internal sense-making functions;
- **Action** area includes the effectors. If the agent acts in its environment, the necessary functions are placed here. Its mission is to execute tasks from the internal decision making area;
- **Communication** area exchanges information with other agents or humans. If it receives information, it is sent to the internal sense-making area. Its task is to send information from the internal decision-making centre to action area.

The internal areas classify the functions needed for the agent to act and adapt as an autonomous object. The four areas identified here are as follows:

Literature Review

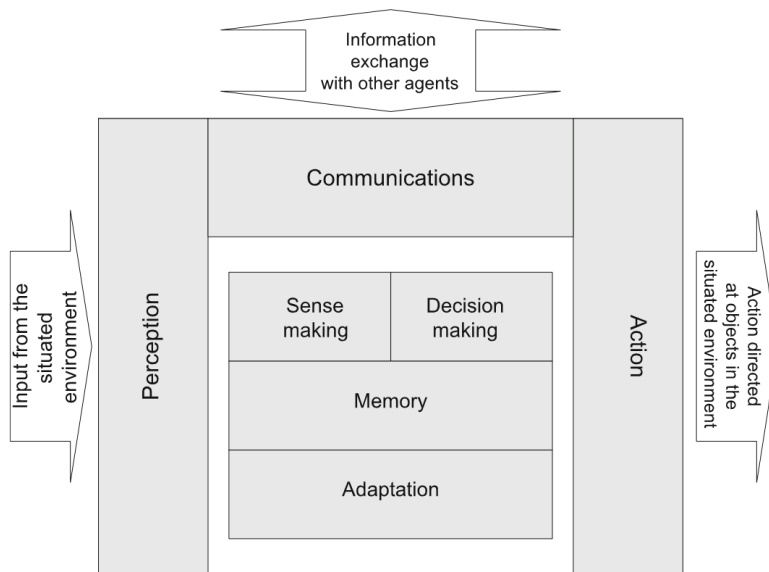


Figure 2.4: Architectural frame of main agent characteristics [MT07]

- **Sense-making** area receives input (sensors and communication) and maps this information to the internal representation, which is part of the memory domain. This domain potentially comprises data correlation and data fusion methods, data mediation capabilities, methods to cope with un- certain, incomplete, and contradictory data, and so on;
- **Decision-making** area supports reactive as well as deliberative methods. It uses the information stored in the memory area and triggers communications and actions;
- **Adaptation** area may be connected with perception and action. The function group updates the information in the memory area to reflect current goals, tasks, and desires;
- **Memory** area stores all information needed for the agent to perform its tasks. It is possible to distinguish between long-term and short-term memory, different methods to represent knowledge can be used alternatively or in hybrid modes, and so on.

Russel and Norvig [RNC⁺95] classify the agent's basic model in four categories: simple reflex, model-based reflex agents, goal-based, and utility-based.

Simple reflex

The simplest kind of agent is the simple reflex agent, which is shown in the Figure 2.5. Its condition-action rules are strictly related with scenario modelling.

These agents select actions on the basis of the current percept, ignoring the rest of the percept history. Although its simplicity, simple reflex behaviours occur even in the more complex environment. It relies on the condition-action rules, i.e. the agent's reasoning analyses an input from its perception, which verify if some condition is satisfied and triggers a specific action from agent

program. Its model is composed by: **sensors**, which allow the agent to perceive its surroundings; **world-state representation**, which represents the actual state perceived; **decision module**, where, based on condition-action rules it is decided what should be done; and **effectors** to perform the agent's behaviour.

The simple reflex considers the environment as simple as possible. To tackle the environment's complexity and dynamics, which prevented a proper and complete characterization of the scenario, the **reflex with internal state** model was introduced.

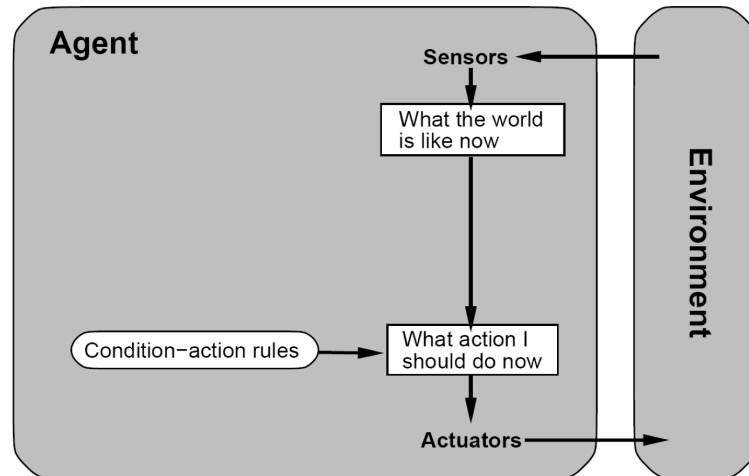


Figure 2.5: Simple Reflex Agent diagram

Model-based reflex agents

The most effective way to handle partial observability is for the agent to keep track of the part of the world it can't perceive. Model-based agent, Figure 2.6, mainly differs from the *Simple Reflex* in the way that it has internal memory to save the world's last state, and has a model of the world's dynamics in order to predict the next state.

Thus, the main differences from the previous model are the state memory, world dynamics and the agent's action's effects.

Goal Based

Knowing something about the current state of the environment is not always enough to decide what to do. Goal-based agent combines this information with its desire to achieve a certain goal, hence this agent model, Figure 2.7, is goal oriented.

The agent must search for possible actions in order to achieve its goals, taking into account their, i.e. the actions', future effect in the world state. As seen in Figure 2.7 a new component was added, which evaluates the result of an action to achieve a certain goal.

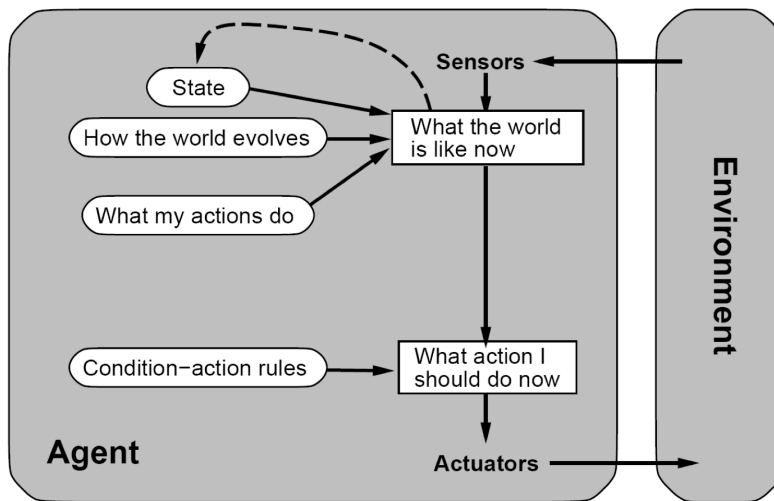


Figure 2.6: Reflex Agent with Internal State diagram

Utility Based

Goals alone are not enough to generate high-quality behaviour in most environments. A comparison of different world states is needed to measure the agent's amount of "happiness" - **Utility**.

Utility-based model, Figure 2.8, uses the concept of utility, which thus represents the degree of satisfaction of an agent. An agent can face the problem to decide among several ways to reach its goal, so a function can be implemented to map the best solution and increase its degree of satisfaction. Therefore, an utility-based agent must also have a model of the environment in order to make decisions, because it must know the states to which its actions will lead.

As we can observe in Figure 2.8 the utility function, which calculates the satisfaction level, decides the action that should be performed next.

Learning Agents

None of the previous agent structures include the agent's capability to learn with its experience. Learning allows the agent to operate in initially unknown environment and become more competent than its initial knowledge would allow.

A learning agent can be divided into four conceptual components, as shown in Figure 2.9. The most important distinction is between the *learning element* and the *performance element*, which are respectively responsible for making improvements, and selecting external actions. The performance element is what we have previously considered to be the entire agent: it takes in percepts and decides on actions. The learning element uses feedback from the critic on how the agent is doing and determines how the performance element should be modified to perform better in the future.

Russell and Norvig consider the learning process in intelligent agents as a:

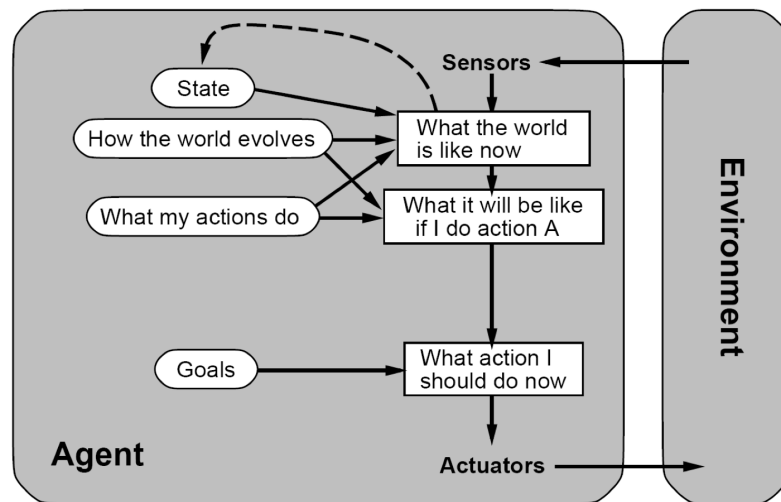


Figure 2.7: Goal Based Agent diagram

"process of modification of each component of the agent, (...) thereby improving the overall performance of the agent"

There are extreme cases in which the environment is completely known a priori. However, generally the environment modifies and augments which requires a rational agent not only to gather information but also to learn as much as possible from what it perceives, keeping its world state up-to-date.

A way to implement a learning agent is by using a Reinforcement Learning (RL) mechanism, in which agent learn from a series of reinforcements - rewards or punishments. The mostly known algorithm is the Q-Learning which we will present in the next section.

Architectures Summary

In short, we can view the world of agents as being categorized this way:

Intelligent Agents	cognitive agents reactive agents
--------------------	-------------------------------------

Reactive agents are simple processing units that perceive and react to changes in their environment. Such agents do not have an internal representation of the environment or other agents and do not use complex symbolic reasoning. Its actions are only based in a set of predefined rules that trigger them. There are some well-known architectures in this type of domain, like the Simple Reflex or the Subsumption architecture, where each task has its own predefined priority [BC86].

Cognitive agents are considered to be characterized by a symbolic level of representing knowledge and by mentalistic notions. An agent with a cognitive architecture makes its reasoning based in his knowledge of world, i.e. the environment's state and properties, as well the dynamicity of other agents in the environment. There are numerous of architectures that characterize this kind

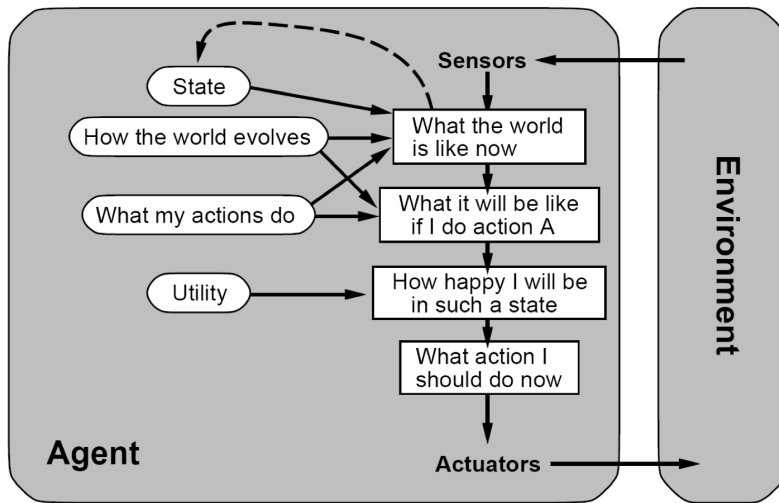


Figure 2.8: Utility Based Agent diagram

of agent. Among them is the BDI (Belief-Desire-Intention) architecture, which states that the agent's beliefs, desires and intentions are enough to characterize an agent [RG92]; Other one is the Competitive Task architecture, which is similar to the Subsumption one but the tasks' priority changes in time through a Reinforcement Learning (RL) process [DF94]. Other studies include Neural Networks (NN) and a combination of some agent architectures [Tyr93, SF00].

However, in the literature we meet also Hybrid Architectures that combines features from the above mentioned ones, building thus more, flexible agent designs. One of those is the Delegate Agent concept used by Joachim Wahle, et al in [WBKS02], where they define a two-layer agent architecture. A basic Tactical Layer which is responsible for the driving task i.e., accelerating, braking, or changing the lane; And a Strategic Layer to solve more sophisticated problems, like route choice behaviour and navigation.

Reinforcement Learning

Reinforcement Learning (RL) is an approach to machine intelligence that combines two disciplines to successfully solve problems that neither discipline can address individually: the Dynamic Programming (DP) for optimization and problem control, Supervised Learning (SL) is a general method for training a parametrized function;

In RL, the agent is given a goal to achieve. It then learns how to achieve that goal by trial-and-error interactions with its environment, receiving negative or positive input (reinforcement) associated with its level of success. Hence it encourages exploration, since the agent does not know the result of its action before its execution.

Q-learning [WD92] is probably the best-understood reinforcement learning algorithm. In Q-learning, the agent learns a mapping from states and actions to their utilities. An important assumption of Q-learning is the Markovian environment assumption, meaning that any information needed to determine the optimal actions is reflected in the agent's state representation, i.e. when

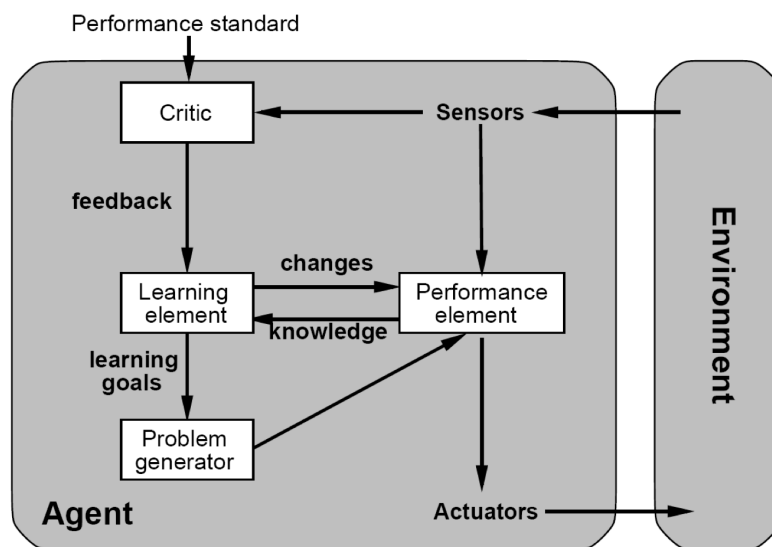


Figure 2.9: Learning Agent diagram

its sensors are not able to make essential distinctions among world states, the Markov assumption is violated.

An agent tries an action at a particular state, and evaluates its consequences in terms of the immediate reward or penalty, giving the final state a value - **utility** (Equation 2.1). By trying all actions in all states repeatedly, it learns which are the best ones.

$$Q(\text{state}, \text{action}) \rightarrow \text{utility} \quad (2.1)$$

In a certain state s , given the a Q-function, the control policy is simply choose the action a for which $Q(s, a)$ is maximal. The Q-function is incrementally built and updated along time, taking into account the previously stated rewards or penalties, according to the Equation 2.2. Given a state transition (s, a, s', r) meaning that an action a in response to a state s results in a new state s' and pay-off r (reward or penalty):

$$Q(s, a) = (1 - \eta) Q(s, a) + \eta \cdot (r + \gamma \max_{a'} Q(s', a')) \quad (2.2)$$

where η is the learning rate, γ is a discount factor and $\max_{a'} Q(s', a')$ is the maximum Q-value in the new state.

Covering all actions in all states, the agent learns which actions should result in a better utility in each state, by analysing the Q-table values.

2.3.2 Environment in Multi-Agent Systems

As we said previously MAS is formed by several agents interacting with each other and with an **environment**, in which they live. Thus to model such system we should take into account not only the single agent entity but also the **environment** where the agents live and share information with each other. The multi-agent systems can differ in numerous ways. They can contrast in the

agents they host, their interaction, and the environment in which the agents act. Table 2.1 presents a summary of some multi-agent systems' attributes, allied with their potential range [Wei00].

	Attribute	Range
Agents	number	≥ 2
	uniformity	homogeneous...heterogeneous
	goals	contradicting...complementary
	architecture	reactive...deliberative
	abilities (sensors, effectors)	simple...advanced
Interactions	frequency	low...high
	persistence	short-term...long-term
	level	signal-passing...knowledge-intensive
	pattern	decentralized...hierarchical
	variability	fixed...changeable
	purpose	competitive...cooperative
Environment	predictability	foreseeable...unforeseeable
	accessibility	unlimited...limited
	dynamics	fixed...variable
	diversity	poor...rich
	resources	restricted...ample

Table 2.1: Multiagent systems' attributes and associated range (adaptation from [Wei00])

Danny Weyns models the environment as an explicit part of a Multi-Agent System. He advocates [WOO07] that the environment is a *first-class abstraction* in a MAS with two main roles: it must provide the surrounding conditions for agents to exist and also allow an *exploitable design abstraction* for building MAS applications. This implies that environment is an essential part of every MAS becoming a building block for MAS that engineers can use creatively in the design of a MAS.

Such an environment could consist not only of other agents, but also of resources, infrastructures, obstacles and other entities. Within the environment, agents will interact in order to accomplish their individual goals and the overall systems goals, even though this is not strictly necessary.

As reported in [Klu] following a general multi-agent system architecture, discussed under the FIPA³ framework, can consist of the following components that can be usually found in a MAS's environment:

- Mediators, facilitators, yellow/white page server and other entities that serve as infra-structure supporting the functioning of the particular, productive agent system by offering services needed for identifying and finding others;

³Foundation for Intelligent Physical Agents Website - www.fipa.org

- General infra-structure beyond helpful entities, namely bus systems or other elements that carry and transport information;
- Resources in the widest sense that may be information sources, databases, service provider, i.e. entities that are not mere infrastructure but form the part of the environment the agents are working with.

Furthermore, besides considering the environment as a facilitator, Oliveira E. [Oli12] defends a "*social-based*" environment, promoting a **proactive** environment in MAS. A "social Environment" should also reason at a higher level, thinking about agents' collective behaviour and decide how to influence autonomous agents, which are aware of external conventions, norms and authorities. This way, the environment can actively decide to punish or reward behaviours that may be considered maleficent or beneficial for that particular society.

As Oliveira E. [Oli12] stated, the environment can also have a policy-maker role in a MAS, however, in our case we will consider the environment as the ambient where our society lives and its entities act.

2.3.3 MAS deployment in Traffic Domain

Agent-based computing is one of the powerful technologies for the development of distributed complex systems [ZV03]. The growing interest in agent technology results from its inherent ability to decompose a system into multiple agents to achieve a global objective.

In this particular case, we are going to study agents related and directed to traffic simulation, specifically the Driver-Vehicle Agents. This term refers to the idea that a single entity, i.e. the agent, incorporates the complex human driving behaviour as well as the properties of a vehicle. The behavioural part of this entity reasoning is based on the decision-making process in a short or long term basis taking into account the characteristics of human behaviour in a given situation e.g. the quick overtaking decision or the re-routing in a network.

These variations in the complex human behaviour are modelled using mathematical models whereas the external part of the agent is related to the physical properties of the vehicle, like fundamental laws of kinematics.

There a great number of examples reported in the literature which use this powerful approach mainly with traffic and transportation management, as well as with microscopic representation of human entities behaviour e.g. especially drivers.

In Table 2.2 we present some real works encountered in the literature.

The TRACK-R agents project [GSTCM03] is focused in joining the development of FIPA compliant agents with knowledge-based techniques, successfully providing traffic routes recommendations for humans and other agents.

The Mobile-C platform [CCP06] is a FIPA compliant framework for mobile and heterogeneous agents. Mobile-C has been used to simulate highway traffic detection and management,

Literature Review

Project	Domain	Features
TRACK-R [GSTCM03]	Road Traffic Management	FIPA-compliant MAS development to provide traffic route recommendations
Mobile-C [CCP06]	Traffic Detetion and Management System	Supports stationary and mobile agents in a hybrid control architecture
InTRYS & TRYSA2 [HOGS02]	Decision support for Urban Traffic Control	MAS that perform decision support for real-time traffic management in Barcelona
CARTESIUS [LR02]	Traffic Congestion Management	Two interacting knowledge-based agents for inter-jurisdictional traffic congestion management
IDTMIS/UTS [Roo99]	Urban Intersection Control	theoretical UTC model based on several intelligent intersection traffic signalling agents (ITSA), road segment agent (RSA), and some authority agents

Table 2.2: Agent’s applications in traffic systems [CC10]

which agents have been used for dynamic code deployment and unanticipated actions performance. This platform has shown to have a great potential to enhance the ITS capabilities, like flexibility and interoperability.

Integrated TRYS (InTRYS) and TRYS Autonomous Agents (TRYSA2) [HOGS02] are two multi-agent systems that perform decision support for real-time traffic management in the urban motorway network around Barcelona. They deal with local traffic problems by resorting to agents’ knowledge-based reasoning techniques.

CARTESIUS [LR02] is a MAS architecture for traffic congestion management on free-ways and surface street (arterial) networks, providing real-time decision support to Traffic Operations Centre personnel. CARTESIUS is composed of two interacting knowledge-based agents that perform cooperative reasoning and resolve conflicts on congestion and formulate integrated control plans.

Integrated Dynamic Traffic Management and Information System (IDTMIS) [Roo99] is a multi-user, multi-disciplinary traffic management and control system directed to the applicability of autonomous intelligent agents in Urban Traffic Control, namely in Urban Intersections. It incorporates several intelligent entities like Traffic Signalling Agents (TSA), or Road Segment Agents (RSA) in order to adapt and respond to traffic conditions in real-time, making better use of the intersection’s capacity.

Other projects are related to traffic-light management systems, Macedo J. et al [MSTR12]

present an integrated a centralised MAS architecture for intelligent traffic-light control, aiming at the overall improvement of the network's performance under the cooperative and adaptive assumptions. It proposes a hierarchical architecture for micro and macro perspectives on the network's performance.

2.4 Modelling & Simulation

In the literature there is not a exclusive definition for Modelling and Simulation (M&S). Each research domain within or related to the M&S discipline, tends to define with its own perspective.

Pragmatically, one could say that **simulation** is the imitation of some real entity, object, or process over time, which embodies certain behaviours or features of the selected physical or abstract system.

A **model** is a representation of a system; it is a way to reproduce the actual system's performance and state, during its life-cycle.

Therefore, **Modelling** is essential, for helping the modeller to predict the effects that changes in the system can provoke in its being. Furthermore, **Simulation** is mainly a execution of a model of a system. This action is very valuable, since most experiments are very expensive and impractical to perform in the real-world.

2.4.1 Synergy Simulation & Agent Technology

For more than two decades, the field of MAS and the field of simulation have been combined in very active aspects of research. The **synergy** between these two areas of research is very fascinating, bringing valuable benefits and developments between these two strands of study.

On the one hand, simulation has often been used as a support to analyse and design MAS in several applications domains, **M& for MAS**. MAS are used as a *programming paradigm* and are very useful in application domains that are characterized by inherent distribution of resources, dynamic conditions in its environment. Thus due to its, the application's domain, complex and dynamic nature, **simulation** is imperative for software testing and agent's decision making experimenting. Some examples of these kinds of ambient are the electronic-markets, supply chain managements, transportation systems or military systems.

On the other hand, we have the other way around synergy, coined **MAS for M&S**. Modelling systems, that is, MAS is used as an approach of *modelling paradigm* to build artificial laboratories for *in vitro* experiments, controlled, monitored. This approach exploits the MAS characteristics to model a **simulation** scenario. A typical example where it happens is for simulating traffic situations, in which drivers' behaviours can be naturally modelled as agents.

A more detailed view on this synergy is made by Ören [ONU⁺00], who coined the term "*agent-directed simulation*" to introduce the different merging of agent technology and simulation adding a new dimension to the existing synergies.

Ören and Yilmaz [YO07] formalize the Agent-directed simulation (ADS), for system engineering, as a unified and comprehensive framework that extends the narrow view of using agents

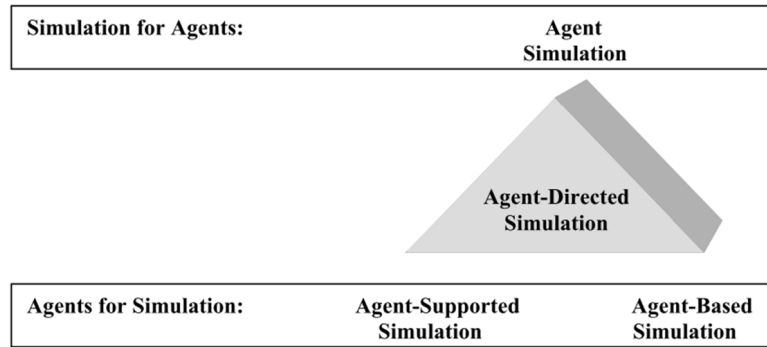


Figure 2.10: Agent Directed Simulation

simply as system or model specification metaphors. Now, the integration of agent technology and simulation is to be considered truly complete. The unified paradigm of ADS, as seen in Figure 2.10, consists of two categories as follows [OY12]:

- **simulation for agents** (agent simulation), that is, simulation of systems that can be modelled by agents, like in engineering, human and social dynamics, military applications, ..);
- **agents for simulation** that can be grouped under two groups:
 - **agent-supported simulation**, is the use of agents as a support facility to enable computer assistance in problem solving or enhancing cognitive capabilities of simulation systems, providing front-end or back-end user system interface functions, such as problem specification, data compression, explanation, problem and/or solution documentation, and solution selection;
 - **agent-based simulation**, focuses on the use of agents for the generation of model behaviour in a simulation study, such as dynamic model composition while in simulation run-time.

Many researcher who don't take into account the contribution of agents to simulation often associate the terms *agent simulation* and *agent-based simulation* as the same principle.

In our work, we will adopt the same perspective, since we will consider agents as a design metaphor, as a *programming* or *modelling paradigm*, not considering their contribution to simulation.

2.4.2 Agents as a metaphor in M&S

Agent-Based Modelling and Simulation (ABMS) is a relatively new technique for computer simulation that aims on modelling & simulating complex systems composed of autonomous, intelligent agents taking also into account their behaviours and interactions.

Macal and North [MN10] state that agent-based modelling is a way to model the dynamics of complex systems and complex adaptive systems. Agent-based models also include models of

behaviour (human or other) and are used to observe the collective effects of agent behaviours and interactions. Although complex, these systems often self-organize themselves and create emergent order.

An agent-based model is one that uses a conceptualization of the reference system as a multi-agent system i.e. a system consisting of interaction “agents” as basis for its model. These can be seen as active autonomous entities that are situated and persistent in an environment where they sense and act accordingly to their constraints. The second element of an agent-based simulation model is the environmental model that defines the frame for the agents to live.

As we have seen previously, using agent metaphor in simulation models is based on the idea that it is possible to represent the behaviour of active entities in the world in terms of the interactions of a society of agents with their own operational autonomy.

With the development of agent modelling tools, advances in computation, thus the availability of micro-data, have made possible a growing number of agent-based applications across a variety of domains and disciplines, from manufacturing control systems reviewed in [La09] to macroeconomic policies in [DN11].

A typical example where MAS models are used in simulation is for simulating traffic situations. In [RBB⁺02] is addressed the complex task of M&S drivers behaviour through the use of agent-based techniques, which is concerned with the reasoning mechanism of drivers modelled by means of a Beliefs, Desires, and Intentions (BDI) architecture.

Another example is the domain of on-line simulations that are used to optimize industrial processes. Such systems apply MAS to exploit their inherent adaptability to cope with the fast evolving distributed business environment. A simulator gathers data from sensors that are connected to the real world; it simulates possible future states, and adapts the system based on the outcome of the simulations. Such systems have been built, for example, for on-line optimization of semiconductor manufacturing [LLL⁺05] where quick decision making is required in order to improve the operational performance or to develop “what-if” analysis to respond to abrupt changes in these field of industry.

2.5 Artificial Societies

The great potential of ABMS one can find it in its application to simulate human societies. Traditional social simulation models were quite naive; they only supported homogeneous populations and sub-populations and thus, were unable to couple different social science disciplines.

First Thomas Schelling [Sch71] with the segregation model and then Epstein and Axtell with “*Growing Artificial Societies*” [EAP96] made a breakthrough about how can we use agents for exploration.

Epstein and Axtell [EAP96] said:

*“Fundamental social structures and group behaviours emerge from the **interaction** of **individual agents** operating on artificial **environments** under rules that place only bounded demands on each agent’s information and computational capacity”*

Literature Review

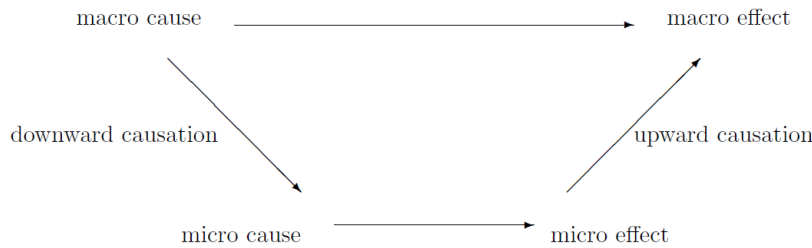


Figure 2.11: The Coleman boat describing downward and upward causation and the link between the micro and the macro level. [UW09]

Not far, traffic systems are inherently linked to an urban context, which can be represented as a society, allowing agents technologies and Multi-agent simulations to cope in this field.

This artificial society is composed by an environment in which a set of heterogeneous agents operate in, obeying to defined behavioural rules (social rules) manifesting global dynamics.

Social systems are among the most complex systems in our world. Coleman [Col94] introduced the “Coleman boat” ,see Figure 2.11, a representation describing the process of human actions and social changes, meaning that a macroscopic actions do not just cause change of the macro state of a society, but at the same time causes a change in the micro state of most or all of the individual beings.

Urban tissue can be represented as a society that presents organizational issues. Thus, the concept of Agents and Agent-based Social Simulation can cope with transportation domain approaches to optimize and design systems.

As known, Metropolitan Systems are spatially spread and are composed by a great number of interacting entities in its domain, each of which with its individual behaviour. Herewith, the combination of all these individual actions have great influence in the overall system’s condition. So, a good abstraction for these environments is agent-based traffic simulations, where each entity of the domain is modelled by an agent, e.g. vehicles, infrastructures, as well its behaviours.

Artificial Transportation System

Fei-Yue Wang in the mid 2000, has used for the first time the concept of Artificial Societies within the context of generalized simulation framework of urban mobility systems coined with the term Artificial Transportation System (ATS). According to this, an ATS:

“is a generalization of the traffic simulation, which integrates the transportation system with other urban systems, such as logistic systems, social and economic systems, etc., to behave as a coordinate tool for transportation analysis, evaluation, decision-making and training.” [LTWW07]

In this situation Wang, describes the use of synthetic populations that live and perform on a given urban context, where members its members - we can call **agents**, organize their activities and their trips according to their needs. Taking all this into account ABM as a useful mind-set where the population is heterogeneous i.e. each individual is a different entity with different and complex behaviours. [Bon02, CCM10]

Due to the high complexity and dynamics of the transportation systems, traditional traffic simulation is not able to capture the uncertainty that illustrates them. Travellers can choose whether to travel or not, can change in any moment their planned routes, their choice can be affected by social or economic or environmental phenomena.

Simulation is a key component in this new step of mobility systems, due to the increased complexity in the test and validation task, which is especially more complex due to real-time constraints and the presence of heterogeneous participating entities: vehicles, urban and traffic infrastructures, pedestrians, among others.

This heterogeneity allows a more complete and expanded evaluation or analysis in a transportation system since it is enables the distribution of resources and a multi-domain analysis.

In short, it is extremely important to understand this concept, since we can have in our hands a tool that can helps us understand how people make choices and can give us possibility to explore the most various situations.

Traffic domain is not only an engineering problem, it goes beyond this idea and has become an interdisciplinary problem. As we have spoken in first chapter, within the FUT concept, the entities play a central and important role in the system, having socio-economical or environmental concerns and preferences.

In this context, the sharing of different domain's tools is very important, and one of the latest trends is the Social Simulation area which has found in the ABMS a very strong, coining the **Artificial Society** concept.

2.6 Summary

Placing ourselves within the new trending of Future Urban Transports (FUT), where people are positioned as a central aspect, as well their preferences, we traced various important concepts to the project, namely the synergies between agent technologies and modelling and simulations i.e. Agent simulation and Agent-based simulation and we presented relevant works that have affected this research bounds.

Making use of the synergy between agent technologies and M&S brings valuable benefits and developments to these two study areas.

Agent-based computing proved to be a very powerful technology to tackle complex systems, namely the traffic domain simulation. It is also a very good abstraction for designing exploratory simulation of information and its effects on the system. Not only, but also to be used to analyse and evaluate strategies and policies to guide a system to the desired outcome.

Literature Review

Chapter 3

Methodological Approach

The use of simulation in solving traffic problems is, in fact, a more practical and intelligent approach. However, when it comes to drivers' decision making in an Advanced Travellers Information System it lacks a proper simulation framework.

This chapter gathers the main requirements for this project, details the proposed solution and architecture and finally describes the milestones for the framework development.

3.1 Requirements Analysis

To develop the desired framework we need a society of entities and an environment where their relationships will occur, Figure 3.1.



Figure 3.1: ATIS and Society of Drivers synergy example ¹

¹Adapted from: <http://automotive.tomtom.com/services/hd-traffic.html>

Having to deal with atomic entities in the transportation domain and the detail level comes down to the vehicle or better to the driver resolution, an Agent-Based Modelling approach seems to be the appropriate way to represent the road traffic environment and the driver entities that live and interact in it. A traffic simulator that can implement the agent paradigm must provide a good API for accessing the vehicles' and other traffic variables, so that we can create, control and monitor its states from an external application. Moreover, it will also allow us to create and artefacts, such as ATIS. We will also need a intuitive MAS development framework, which will be used to implement the drivers' artificial society and various environment artefact such as ATIS. Thus it should be sufficiently portable, to match the simulator's execution environment and also be flexible enough to allow communication with the simulator's API.

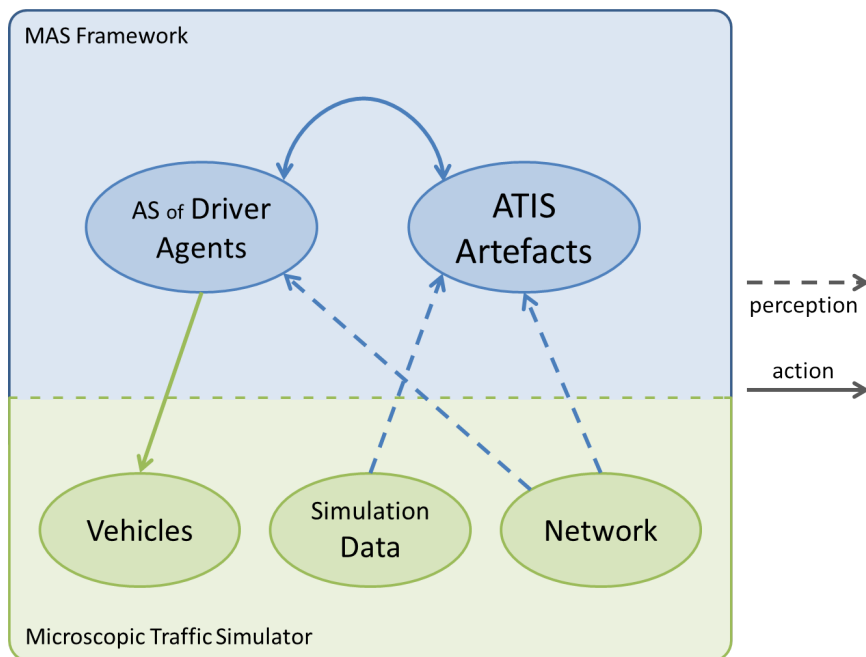


Figure 3.2: General proposed solution

In this artificial laboratory environment, Figure 3.2, the agents should be able to perform communications between to, request ou deliver valuable information. The ATIS infrastructures will gather their knowledge from its network and simulation data perception, e.g. the simulation's elapsed time, among others. The Drivers will perceive its surrounding network and allied to the ATIS's assistance will perform driving decisions.

In the next sections we will present the software selection and proposed architecture, as well the milestones for the solution implementation.

3.2 Proposed Design Solution

Taking into account all the aforementioned requirements we will now present the software choices and decisions and detail their importance to the project's ideals.

3.2.1 Software Selection

The microscopic traffic simulator chosen to provide the environment and traffic entities simulation is **SUMO**, which is an open source, highly portable, microscopic traffic simulator. It provides real-time interoperability with other applications through TraCI's API. This microscopic traffic simulator will provide us with the individual vehicular entities and the traffic environment .

We want to use a standardized and mature MAS development tool, so we chose **JADE** (**J**ava **A**gent **D**evelopment Framework). This free framework is highly portable since it has interoperating versions for every profile of Java Machines. JADE aims to simplify the development of MAS while guaranteeing standard compliance, thus being, we will use JADE's capabilities to create our Artificial Society of drivers and our ATIS artefacts.

Our goal is not just to produce a framework for a single microscopic traffic simulator, but to any that fulfils the aforementioned requirements. So, instead of linking the MAS framework directly to the simulator's API, we will use **TraSMAPI** (**T**raffic **S**imulation **M**anager **A**pplication **P**rogramming **I**nterface) as a generic microscopic traffic simulator API. This tool will allow a higher level of abstraction and transparency, which enables us to use any microscopic traffic simulator. Allowing result comparison among other benefits.

In the next chapter these tools and applications are presented in detail, highlighting their strengths which will be useful for this dissertation.

3.2.2 Proposed Architecture

Taking into consideration all the general requirements and goals we've devised the following architecture, depicted in Figure 3.3.

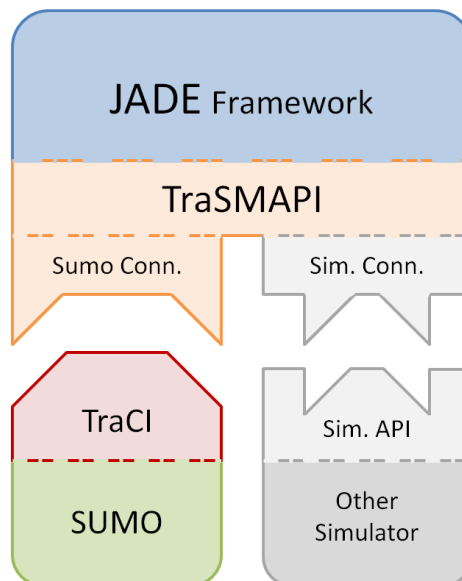


Figure 3.3: Framework Architecture

In this picture we can observe the main contribution of TraSMAPI in our framework. TraSMAPI provides an abstraction over different possible microscopic simulators, which completely makes our platform independent from a microscopic traffic simulator. Besides, it makes possible further studies on simulation results comparisons, since it is possible to test the same solution, i.e. source code, in various microscopic simulators, hence demonstrating TraSMAPI's transparency contribution to the researchers.

In addition to the TraSMAPI block, we can also observe that JADE is directly connected to the microscopic traffic API (TraSMAPI), which has a communication model for SUMO Simulator that reflects the basic API for the interaction with the simulator.

The SUMO Simulator offers an API for access to its simulation state - TraCI. For an external application to communicate with this software it must obey TraCI's communication protocol and messages types. The Sumo Communication Module attached to TraSMAPI, converts this low-level simulator's API to a higher-level one, which will be then used by our artificial society of drivers and ATIS Artefacts. These will be implemented in JADE's MAS development framework coupled to TraSMAPI, Figure 3.3.

3.3 Development Milestones

In this section it will be presented the practical methodology to follow in order to achieve the proposed framework.

The main tasks will be presented and briefly described as their details are specified in the Implementation chapter.

Task 01 : Extend TraSMAPI's SUMO's entity coverage

In this first task we want to extend SUMO's Communication Model, in order to build an interface to all entities simulated in the microscopic traffic simulator. With this task we'll be able to instantiate a Driver entity associated to a vehicle in the simulation, which can be controlled through TraSMAPI, meaning route defining, speed changing, among others.

Task 02 : Artificial Society of Drivers - Experiment

Create an artificial society of drivers and build an experiment, in which we tried to replicate the Braess Paradox ² to validate the learning process of the agent.

Task 03 : Build ATIS entities into TraSMAPI

Use agents technologies programming paradigm to built ATIS artefacts and couple them with the existing ones (Drivers, Traffic Lights, and other). This entity will perceive the microscopic traffic simulator's network state and inform the drivers about desired and applicable relating aspects.

²Original Paper [Bra68] translated to English in [BNW05]

Task 04 : Artificial Society of Drivers in an Informative Environment - Experiment

In this last task, our goal was to gather all the work produced and combine them in an in-vitro experiment, where ATIS infrastructures and an artificial society of drivers cohabit in the same environment.

3.4 Summary

Carefully defining the requirements is fundamental for identifying the required steps to develop an integrated framework, which will act as a guide line through the development of the thesis. In the following chapters, the development tools are going to be described in more detail and the defined milestones will be scrutinized.

Methodological Approach

Chapter 4

Development Software Overview

In this chapter, we will describe and detail the selected software to develop our framework.

First the microscopic traffic simulator SUMO is presented, detailing its API with external applications. Secondly an overview in the multi-agent development framework JADE is given, and finally TraSMAPI framework is also introduced.

4.1 SUMO

Simulation of Urban Mobility (SUMO) [BBEK11] is an open source, highly portable, microscopic and multi-model traffic simulation package designed to handle large road networks and to establish the a common test-bed for algorithms and models from traffic research.

The simulator was developed in the Institute of Transportation Systems at the German Aerospace Centre – with collaboration of the Centre for Applied Informatics in Cologne and it is licensed under the GPL

4.1.1 Simulation of Urban Mobility

SUMO simulator is possibly the most studied microscopic traffic simulator in the research community, with a high number of scientific papers referring to it. In Figure 4.1 is presented a screenshot of SUMO's Graphical User Interface (GUI).

This project started in the year 2000 [KHRW02] with a need of an open-source tool into which several algorithms could be implemented and evaluated, such as road networks, demand or traffic controls. Some of it's main features are mentioned below (adapted from: SUMO user documentation ¹):

- High portability
 - implemented in standard c++;
 - Windows and Linux distributions;

¹Sumo User Documentation: http://sumo.sourceforge.net/doc/current/docs/userdoc/SUMO_User_Documentation.html

Development Software Overview

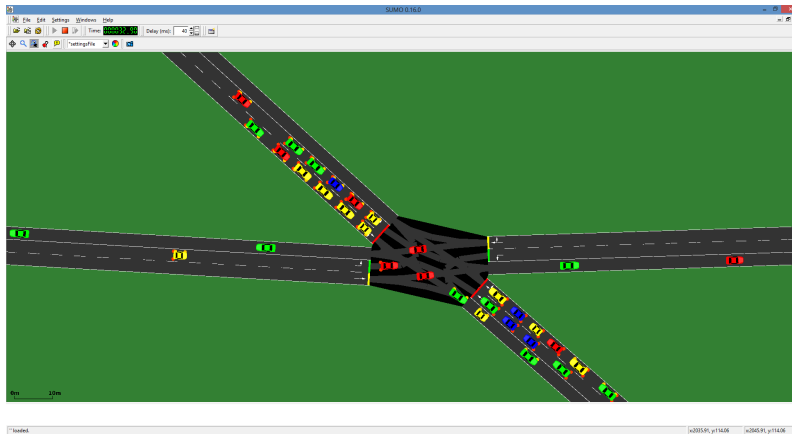


Figure 4.1: SUMO's screenshot

- Simulation
 - time-discrete, space-continuous;
 - vehicles are modelled explicitly (microscopic simulation);
 - car-following model by S.Krauß [Kra98];
 - collision free vehicle movement;
 - different vehicle types;
 - multi-lane streets with lane changing;
 - fast OpenGL implemented Graphical User Interface;
 - manages networks with several 10.000 edges (streets);
 - fast execution speed (up to 100.000 vehicle updates/s on a 1GHz machine);
 - Interoperability with other application on run time using TraCI ² [WPR⁺08];
 - network-wide, edge-based, vehicle-based, and detector-based outputs;
- Routing
 - microscopic routes, thus each vehicle has its own;
 - Dynamic User Assignment [Gaw99];
- Network Import
 - imports network from a large variety of sources : VISUM, Vissim, Shapefiles, OSM, Tiger, RoboCup or XML-Descriptions;
 - missing values are determined via heuristics;
- Includes all applications needed to prepare and perform a traffic simulation (network and routes import, dynamic user assignment (DUA), simulation);

²TraCI User Documentation: <http://sourceforge.net/apps/mediawiki/sumo/index.php?title=TraCI>

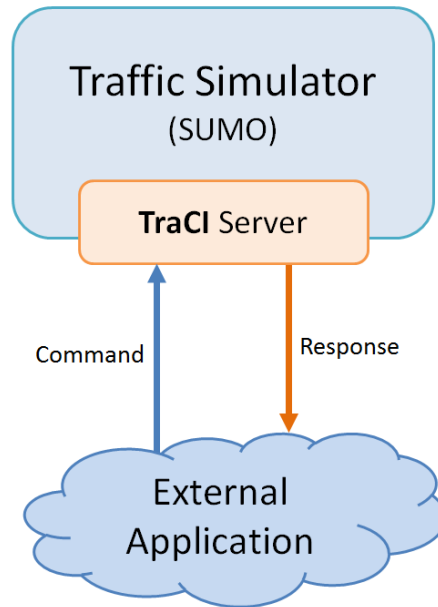


Figure 4.2: TraCI's system architecture

SUMO is a complex project with several contributors [Flo09, PD09], and consists of hundreds lines of code, still growing. Its real-time interoperability is granted by TraCI's interface, which allows us to access a running road traffic simulation.

4.1.2 TraCI

TraCI's (**Traffic Control Interface**) goal is to give access to a traffic simulation in run-time.

TraCI uses a TCP based client/server architecture providing access to SUMO. It opens a port in SUMO's simulation and waits for outbound well defined commands. Its system architecture is depicted in Figure 4.2, where we can observe the basic communication between the Traffic Simulator i.e. SUMO, and other external application using a TCP connection.

It offers us a wide range of features to use during the simulation, in Table 4.1 is presented the API's general structure, however the full table can be consulted in TraCI's documentation website.

SUMO is used in different projects in the literature. It helps to investigate several research topics like simulating realistic Vehicular Ad-hoc NETWORKS (VANETs) [RSRM10, PRL⁺08], to simulate automatic driving [PR12] or traffic management strategies, e.g. intelligent traffic lights [KBM⁺, MSTR12], route choice among others. It was recently use for surveillance purposes, using highway and inner-city induction loops, traffic visualization providing important visual information to the police forces.

Command	Description
Control-related	perform a simulation step, close the connection
Mobility-related	set a maximum speed of a vehicle, stop a vehicle, slow a vehicle down, force lane changing, set a new route, set a new destination edge
Subscription-related	subscribe for retrieval of an object's values
Environment-related	convert positions between different formats, compute routes, ask for values of objects within the scenario
Traffic Lights-related	retrieve information about the status of a traffic light

Table 4.1: TraCI's general API

4.2 JADE

JADE (Java Agent DEvelopment Framework) is a free software framework to develop agent-based applications. Its goal is to **simplify** the development while ensuring standard compliance through a comprehensive set of system services and agents. JADE is fully implemented in Java language and is compliant with the Foundation for Intelligent Physical Agents ³ (FIPA) specifications for **interoperable** multi-agent systems, besides this agent platform can be distributed across several machines, which not even need to share the same Operative System (OS).

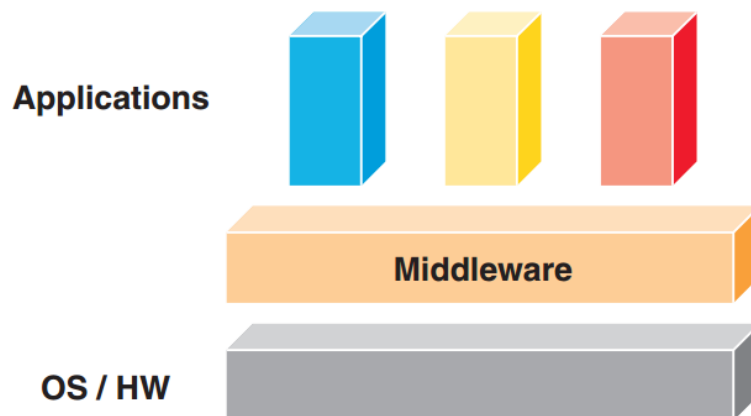


Figure 4.3: Role of the middleware (adaptation from [BCPR03])

This framework can be considered an agent middleware, Figure 4.3 that implements an Agent Platform and a development framework. It deals with all those aspects that are not peculiar of the agent internals and that are independent of the applications, such as message transport, encoding and parsing, or agent life-cycle - Agent Management System (AMS).

³FIPA website: <http://www.fipa.org/>

Development Software Overview

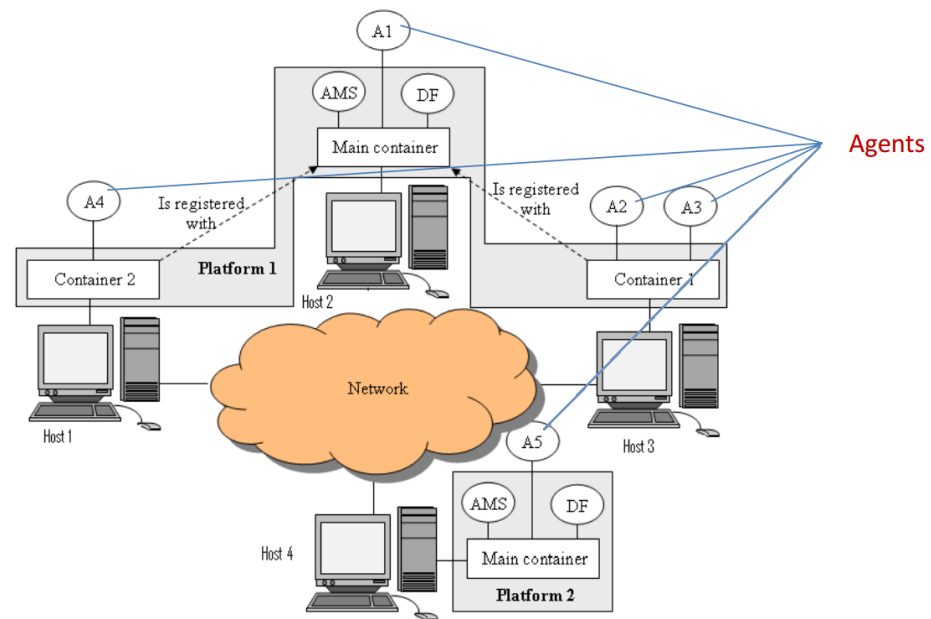


Figure 4.4: JADE Architecture [Car]

JADE's aim is to simplify the development of multi-agent systems while guaranteeing standard compliance with the FIPA specifications: naming service and yellow-page service (Directory Facilitator - DF), message transport and parsing service, and a library of FIPA interaction protocols ready to be used. All agent communication is performed through message passing, where FIPA ACL is the language to represent messages.

The agent platform can be dispersed on several computers, where each runs a single Java Virtual Machine (JVM). Each JVM is basically a container of agents that provides a complete run time environment for agent execution and allows several agents to concurrently execute on the same host, as we can see in Figure 4.4.

Each agent is implemented as one thread, however agents often need to execute parallel tasks. As JAVA language offers multi-threading solutions, JADE also inherits these characteristics moreover it supports scheduling of cooperative behaviours, storing these tasks in a light and effective way. The run-time includes also some ready to use behaviours for the most common tasks in agent programming, such as FIPA interaction protocols [BPR99].

The agent platform provides a Graphical User Interface (GUI) for the remote management, monitoring and controlling of the status of agents, allowing, for example, to stop and restart them. The GUI allows also to create and start the execution of an agent on a remote host, provided that an agent container is already running (Figure 4.5).

One of the main goals in developing a middleware is to maintain a high degree of openness and flexibility, making it applicable in as many different domains as possible. JADE spread is very vast being employed in many different contexts ranging from academic studies to substantial

Development Software Overview

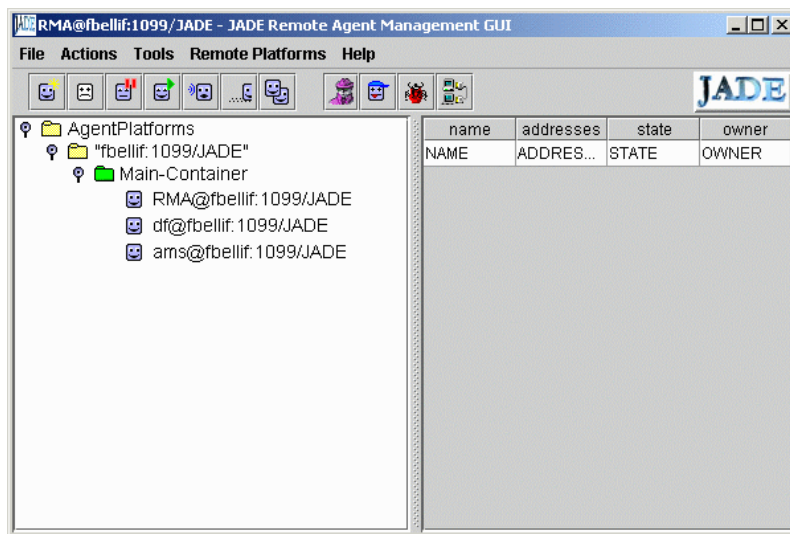


Figure 4.5: JADE Graphical User Interface

business applications around the world.

Numerous R&D projects, where an interaction between numerous elements is required, and in which an autonomous and dynamic adaptation to complex relations is needed, have used JADE as a developing tool. In traffic domain, there are several works that profit from JADE's platform for developing Multi-Agent System traffic management solutions [SR10, VMvBvK03].

4.3 TraSMAPI

TraSMAPI (Traffic Simulation Manager Application Programming Interface) is a synergy between three main components (Figure 4.6): an Application Programming Interface (API), a Statistics module and a Multi-Agent System framework.

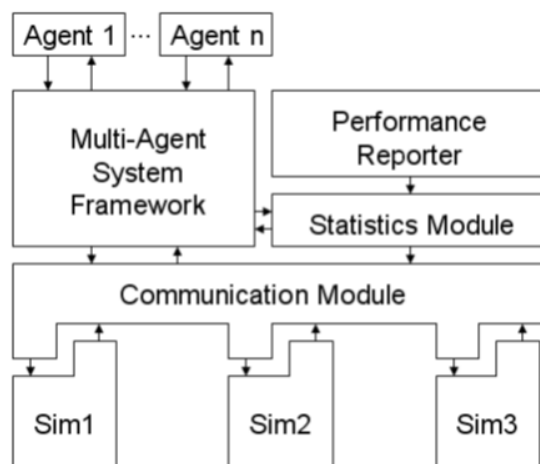


Figure 4.6: TraSMAPI's general architecture

The TraSMAPI allows real-time communication between traffic-management agents and the environment created by the simulator.

This interaction enables the simulation of dynamic control systems which can adapt according to their environments and offer the most appropriate, solution to the specific traffic situation that is being analysed.

The API was built in an abstraction level higher than most common Microscopic Traffic Simulators so that, ideally, the solution should be independent of the microscopic simulator choice. This is guaranteed as far as the chosen simulators allow it, and provided that their communication interface differs and they do not implement the same set of features. In Figure 4.7 is presented a usage of this tool for testing traffic-light management algorithms in two different microscopic traffic simulators [TARO12].

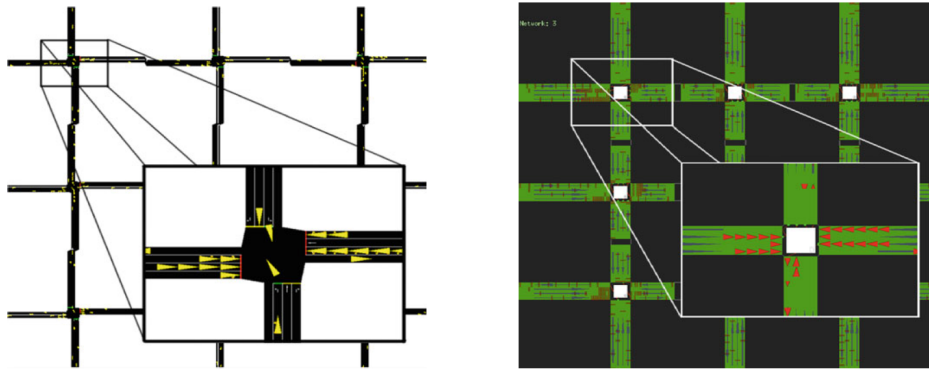


Figure 4.7: TraSMAPI usage in several microscopic traffic simulators, SUMO (left) and ITSUMO (right)

This feature allows the comparison of results from different simulators using exactly the same traffic management solution. To this it is used the statistics module in order to assess the performance of the solution and to provide dated statistics which could be essential for learning agents [TARO10].

The general architecture of TraSMAPI is based on modules each of which with a well-defined function in the whole system, Figure 4.6.

The application is organized in three main Modules: the Communication Module, the Statistics Module, and the Multi-Agent System Framework.

- **Communication Module** provides the basic API for the interaction with the simulator to gain knowledge on the environment state, and orders, to change the state of the environment. The exchange of information is usually done by remote communication technologies dependent on the simulator over which the abstraction is being built;

- **Statistics Module** is a passive module that does not interfere directly with the simulation. However, it is very important to develop real-world applications. The Statistics Module records all the simulation data acting as a historical archive;
- **Multi-Agent System Framework** is a module that is meant to serve as a starting point for the creation of Multi-Agent Systems. Its Framework allows the creation of new agents by following a common interface. The agents themselves are created with a reference to one or multiple objects in the simulation gaining direct access to these simulation artefacts or entities.

Concerning our work, we aim to substitute this MAS Framework Module for a more widely distributed MAS frameworks - JADE. With this, we orient TraSMAPI also to real-world solutions implementations, since it will have a more mature, generic and FIPA compliant MAS development framework, Figure 4.8.

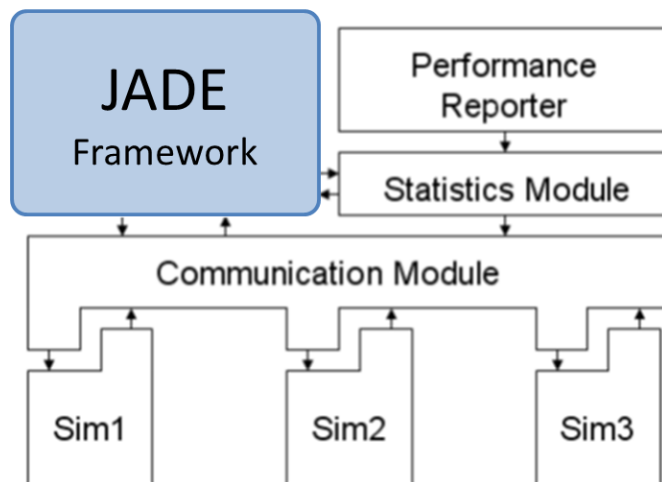


Figure 4.8: TraSMAPI's envisioned general architecture

We have already made some experiments regarding traffic management solutions using TraSMAPI and SUMO Simulator [MSTR12].

Besides that, we have also extended TraSMAPI to other microscopic traffic simulators, IT-SUMO [BdSB10] and AIMSUN⁴.

4.4 Summary

A practical solution for the integration of a microscopic traffic simulator with a framework to develop agent-based applications was presented.

⁴ Aimsun Website - <http://www.aimsun.com/wp/>

Development Software Overview

We have chosen free and open source software that are in constant development by research community. These activities and constant improvements demonstrate all the value and control that these frameworks offer to the research community.

Our tool will consist in a coupling of two independent systems, therefore, their original documentation can still be used. Moreover, in the following chapters, a more technical and detailed overview of the tool usage will be presented.

Development Software Overview

Chapter 5

Implementation

After creating the concept of the application and development software analysis according to its description in the previous chapters, several features were developed. In this chapter we will focus on the implementation of such features, detailing each one of the previously identified tasks, refer to Section 3.3.

First we will focus on the Artificial Society of drivers and its implementation and linkage to a microscopic traffic simulator. The general architecture of each agent will be detailed and related to the delegate-agent concept [WBKS02].

Secondly, the ATIS Artefact development is detailed, presenting the three types of implemented infrastructures.

In the next Chapter 6 more implementations details will be specified, though related to the experimental set-ups.

5.1 Driver Agent Development

Our goal is to have an heterogeneous artificial society of drivers in JADE's agent platform, each of its entities responsible for one vehicle in the SUMO's traffic environment.

However it would be very computationally expensive to simulate hundreds or thousands of vehicles and driver's decision-making in JADE. Hence, we have adopted the delegated-agent concept, which has been used in [WBKS02], to separate the tactical from the strategic layer of the agent, and execute them in parallel, thus improving performance, Figure 5.1.

The tactic-reactive layer is assured by SUMO's microscopic traffic simulation model, like car-following [KWG97], leaving the cognitive-strategical layer to the agent implemented in JADE's agent platform. Therefore, for this first milestone we must make possible the association between a Driver Agent, instantiated in JADE's agent platform and a vehicle simulated by the microscopic traffic simulator SUMO, Figure 5.2.

In order to achieve these ideals, we need to extend the scope of TrasMAPI, enabling it to build an abstraction over a vehicle entity, in addition to the already implemented Traffic Light concept, refer Figure 5.3 .

Implementation

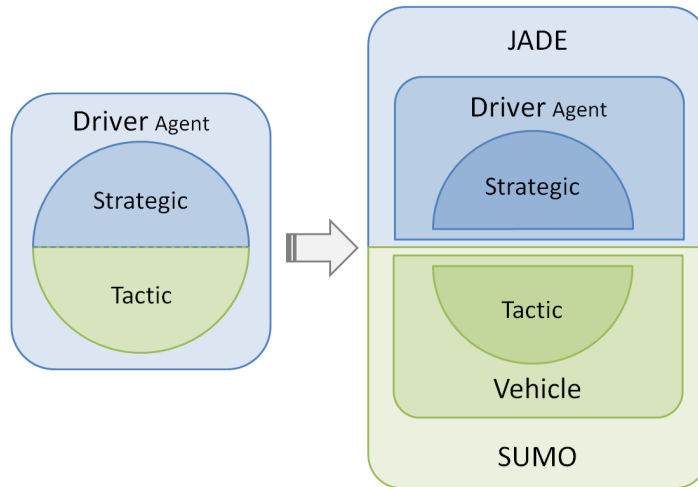


Figure 5.1: Driver's Layers: tactic-reactive in SUMO; strategic-cognitive layer in JADE.

As mentioned in the previous chapter, Section 4.1, the simulator offers real-time interoperability with other external applications by implementing a TraCI Server. Therefore, we need to create the module to communicate with this tool access microscopic simulation data.

There are already some implementations of applications with the same purpose. The developers of SUMO use a Client-Library¹ written in Python when they interact with the simulator, however, this client can be written in any programming language that supports TCP sockets for remote connections.

TraCI4J² is a Java library, developed at Politecnico di Torino, for interfacing SUMO simulator with a Java program. Since TraSMAPI is also written in Java programming language this could be an excellent tool to add to our framework however, it presents some shortcomings, which we present below in Table 5.1.

	Advantages	Disadvantages
TraCI4J	Communication protocol is stable	Implementation and Documentation still in <i>alpha</i> version.

Table 5.1: TraCI4J tool analysis

Since it is a tool still in development, *alpha* version, and taking into account that its development has been decreasing over time, it is unwise to use it in our implementation since its maturity isn't guaranteed .

¹TraCI-Python : http://sourceforge.net/apps/mediawiki/sumo/index.php?title=TraCI/Interfacing_TraCI_from_Python

²TraCI4J Documentation : http://sourceforge.net/apps/mediawiki/traci4j/index.php?title=Main_Page

Implementation

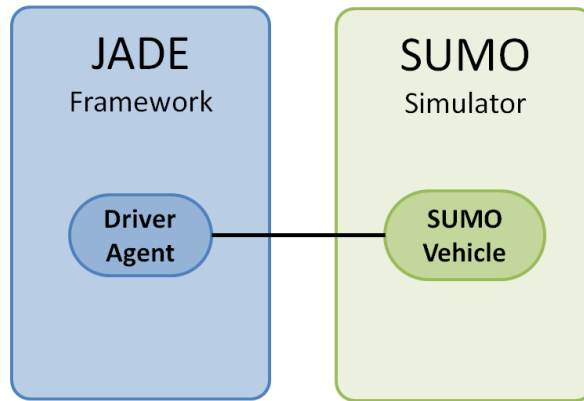


Figure 5.2: Driver-Vehicle Association

Nevertheless its connection protocol to TraCI is quite stable. So we withdrew some concepts of TraCI4J's protocol implementation and took that as a starting point to extend and improve the TraSMAPI's communication module with the traffic simulator SUMO.

Command	Variable
Vehicle Value Retrieval	speed
	road Id
	lane Id
	route Id
	edges
Change Vehicle State	max speed
	speed
	stop
	change lane
	change target
	change route by id
reroute by travel time	
reroute by effort	

Table 5.2: Some Vehicle-directed TraCI Commands (adapted from TraCI's Documentation)

Thus, we implemented the communication protocol that complies with well-defined TraCI's instructions, for more detailed information see TraCI's documentation³ and reference [WPR⁺08], where the protocol and message flow are presented and explained.

Regarding the vehicle entity we have implemented several functions for variable retrieval or state change. The following ones, Table 5.2 are the directly related ones to our project.

³TraCI's documentation : <http://sourceforge.net/apps/mediawiki/sumo/index.php?title=TraCI>

Implementation

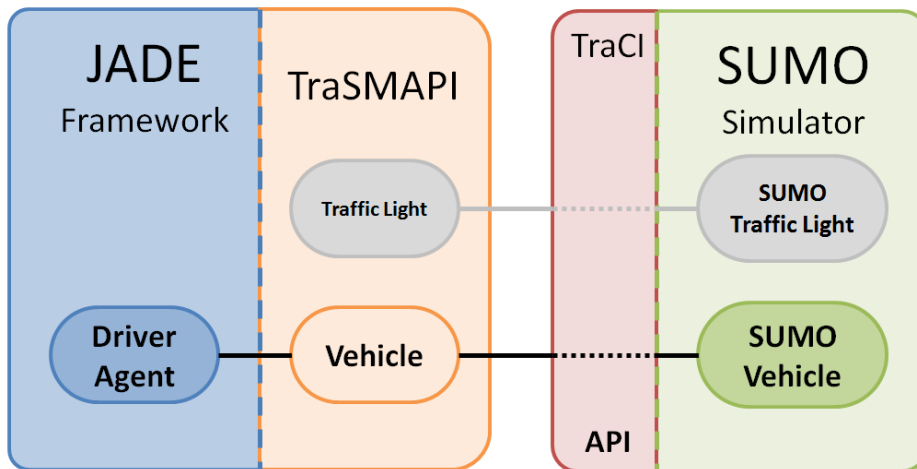


Figure 5.3: Vehicle entity abstraction

To test this implementation we have developed a very simple experiment, where a driver stops for an undefined time in the middle of an edge, causing a traffic jam.

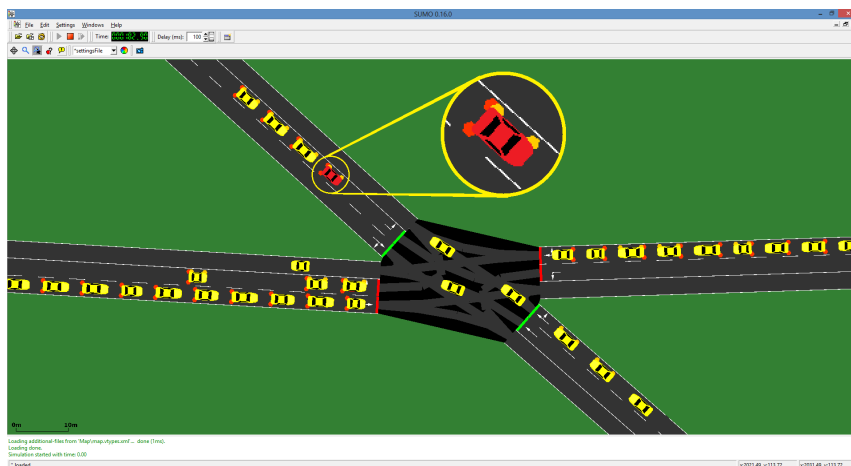


Figure 5.4: Induced Traffic Jam

As we can see in Figure 5.4, the red car stays still, despite having green light to carry on its journey, causing the following cars to stop behind him.

With this accomplishment we are able to instantiate a vehicle entity in a driver agent's state, making it responsible for high-level actions, like the ones stated in Table 5.2.

5.2 ATIS Artefacts Development

Using agent technologies programming paradigm to build and model the ATIS artefacts we were able to blend them in JADE's agent platform. Thus, providing them the ability to communicate with the other agents, namely the drivers. Moreover this artefact will perceive information from the environment simulated in SUMO. This way, we can consider this infrastructure as a perceiver and a service provider, as we observe in Figure 5.5.

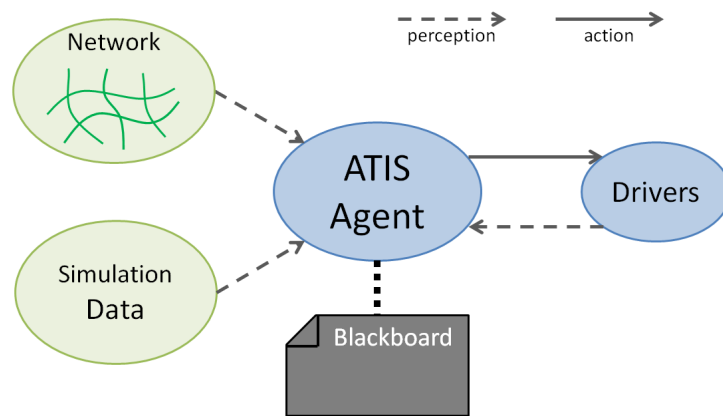


Figure 5.5: ATIS Interactions with Environment and Drivers

The set of perception variables from which the ATIS infrastructures gather information are quite vast. It varies from network data, simulation information and drivers' accident notification, building a common knowledge base on traffic information - the Blackboard.

An ATIS Agents is responsible for a certain type of work and field of action. We have defined three types of ATIS Agents, the Radio Broadcast (RB), the Variable Message Sign (VMS), and the Informative Traffic Lights (ITL), see Figure 5.6.

Each ATIS entity uses the microscopic traffic simulator SUMO as sensory environment. To represent this infrastructure inside the simulator we used the SUMO's Polygon⁴ abstraction, which does not have an active role in the simulation.

To gather this sensory data API's calls must be executed. However, the mere fact of asking a variable and get its value can be quite time consuming, since it demands the exchange of two messages, i.e. the request and the variable value reading.

Considering this performance issue, SUMO's API (TraCI) provides two subscription commands that showed to be very useful to retrieve information on network data: variable subscription and context subscription. With these commands, one can register a request for a value retrieval for a defined amount of time, which eliminates the request phase, thus reducing the execution time in half.

⁴A sequence of 2D points, representing a polygon shape.

Implementation

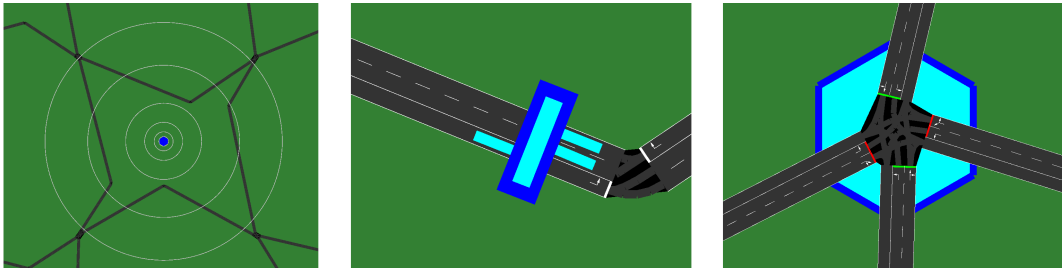


Figure 5.6: All three implemented ATIS entities : the **left** one is the Radio Broadcast, identified by the concentric circumferences that represent information diffusion; the **middle** one is a VMS that indicates the lanes to which it transmits information; the one on the **right** is the Informative Traffic Light that is responsible for a specific junction.

Variable subscriptions provides a periodical update on a structure's variable.

Context subscriptions allows the obtainment of specific values from surrounding objects of a certain object within a defined range. This is the reason why we have represented the ATIS entity as a polygon object in the simulated environment.

Radio Broadcast (RB)

This element in our set of ATIS entities is the one that covers more area. It benefits from the context subscription command in TraCI's API, which enables the gathering of specific values from surrounding objects within a certain range.

The Radio Broadcast is responsible for a confined area, from which it can identify the vehicles ids and send them the desired and information.

Moreover, this entity can be aware of any variable from the ranging edges, junctions or induction loops, using them to enrich the global information knowledge.

Variable Message Signs (VMS)

VMS entities are responsible for only one road. Its offered information is only visible by passing by travellers. It knows at each moment which vehicles are in the present edge and can inform them about some incoming traffic jam, roadblocks or even advise prudence due to roadworks.

This entity can be combined with the range capability, since drivers can not read its indications from a far distance, which means that the VMS only offers its knowledge when the Driver is close enough.

Informative Traffic Lights (ITL)

Informative Traffic Lights are an extension of the VMS ATIS. Instead of being responsible for collecting and sending data from a single edge, it monitors a whole junction of edges.

Typically it is associated to a traffic light management intelligence, however this idea is beyond this project's scope. ITL have access to incoming lanes data, including vehicles ids and occupancy.

Implementation

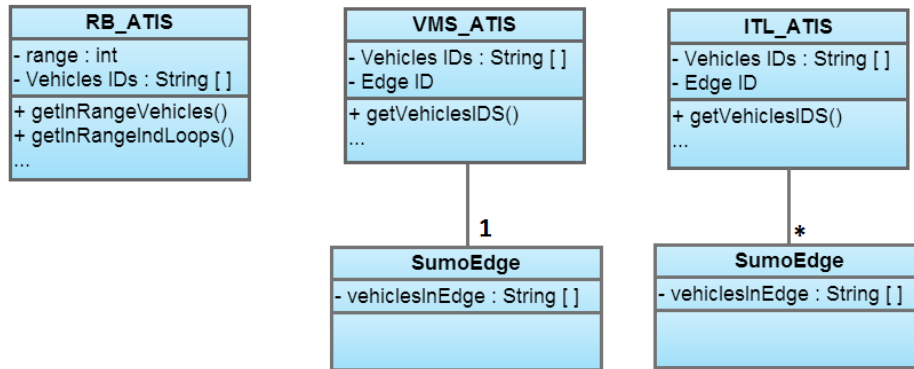


Figure 5.7: Simplified class Diagram for all ATIS Artefacts

In short, with this implementation we were able to create an entity that gathers facts from the network and simulation data, and provides them to the driver agents through JADE’s messaging network.

5.3 Summary

As known, Metropolitan Systems are spatially spread and are composed by a great number of interacting entities in its domain, each of which with its individual behaviour. Herewith, the combination of all these individual actions have great influence in the overall system’s condition. So, a good abstraction for these environments is agent-based traffic simulations, where each entity of the domain is modelled by an agent, e.g. vehicles, infrastructures, as well its behaviours, creating an Artificial Society.

Therefore, we have modelled this AS in JADE’s agent platform, being each driver an agent. We have adopted the delegate agent concept to model our agent’s architecture [WBKS02]. The tactic-reactive layer was entrusted to the microscopic traffic simulator, taking care of reactive tasks related to the driving itself (breaking, change lane, slow down, among others). The strategical layer was kept in JADE’s side. Here, the researcher can implement his own strategical architecture. In the next chapter we will implement a learning architecture with a reinforcement learning approach.

Concerning the ATIS Artefacts we have implemented three times of entities, however, many others can be invented with much ease, since they are on the programmer’s side, meaning outside the simulator’s domain, being independent of the simulation.

The ATIS concept is considered an Artefact since it has a reactive architecture, providing the services and functions that make individual agents work together in a MAS.

However, it was implemented as an agent, in order to envisage a future application in the futuristic Autonomous Road Transport Support Systems (ARTSS) concept [McC]. These ARTSS, using autonomic approaches, should have the potential to deliver savings in the cost of system

Implementation

configuration, maintenance, and infrastructure, while potentially improving network efficiency and reducing the chances of human error.

Chapter 6

Experiments & Results Discussion

In this chapter we are going to present two experimental set-ups.

In the first one we replicate the Braes Paradox in a route choice scenario. Here we can notice the Artificial Society capability of learning and adaptation in adverse and dynamic situations.

In the latest, we present a en-route diversion due to an induced accident during a simulation. The vehicles receive the accident notification and change their route if the accident directly affects their journey. This attitude argues in favour with the drivers' awareness and decision-making capabilities.

6.1 Braess Paradox Experimental Set-up

There are generally two types of travel behaviour: **user-optimizing behaviour**, in which travellers select their optimal route, whom are generally characterized as “selfish”; and **system-optimizing behaviour**, in which a central controller directs traffic. Our work focuses on the first one, the driver behaviour, and the Braess paradox occurs only for user-optimizing behaviours.

In an urban area with a lot of traffic, adding a new road to distribute and facilitate traffic may seem like a sensible idea. However, according to the Braess paradox, just the opposite occurs: a new route added in a transportation network actually increases the travel times of all individual travellers[Bra68, BNW05].

The Braess Paradox is a good illustration of how easily our intuitions about collective interaction can be fooled.

Car drivers seek to minimise the time to get from O (Origin) to D (Destiny), however, car drivers may not be able to act independently of each other: collective interactions may influence individual behaviour.

In this experiment we try to replicate this paradox by setting up an artificial society of “selfish” learning drivers, in a well defined scenario. Their goal is to get from point A to point B the fastest way possible.

Experiments & Results Discussion

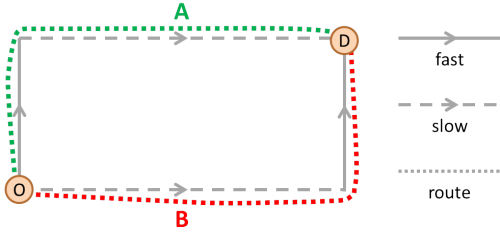


Figure 6.1: Two route network

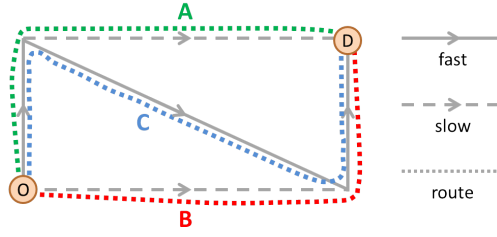


Figure 6.2: Three route network

The network, sketched in Figure 6.1, starts by being composed by two symmetrical routes, each of which consists of a fast section and a slow one. Then at a certain time, a new road is added, Figure 6.2, providing drivers more and better road resources.

We have built an artificial society of Q-learner drivers, which will "live" for 500 days and perform, each day, a trip from point O to D. When arrived, each driver registers his Travel Time (TT):

$$TT = arrivalTime - departureTime \quad (6.1)$$

Taking the environment into account we have modelled it in a finite-state automaton, depicted in Figure 6.3, building the correspondent Q-table (Table 6.1), where each route choice in state s generates an *utility*.

Since our problem is scalar, depending only on the route choice, we can simplify it to $Q(r)$, being r the route chosen. Hence our update-function is:

$$Q(r) = (1 - \eta)Q(r) + \eta.R \quad (6.2)$$

being η the learning rate and R the Reward function:

$$R = \frac{aTT}{TT} - 1 \quad (6.3)$$

while TT is current TT and aTT is the average Travel Time of all trips:

$$aTT = \frac{\sum TT}{\#trips} \quad (6.4)$$

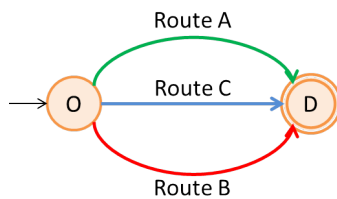


Figure 6.3: Experiment's States & Actions

Q-Table	route A	route B	route C
state O (s)	$Q(s,A)$	$Q(s,B)$	$Q(s,C)$

Table 6.1: Experiment's Q-Table

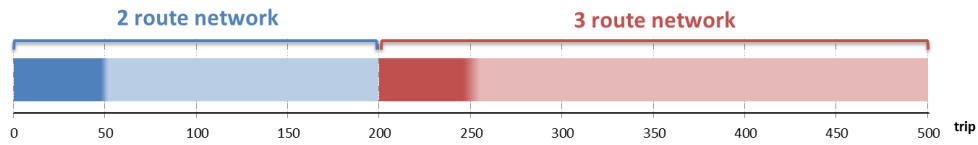


Figure 6.4: Network configuration through time

For our test-bed we have defined an exploring and exploited time in each network configuration for all 500 days, as shown in Figure 6.4.

Each network configuration, meaning different route arrangement, is **explored** by the driver agents during 50 days, in which the drivers are randomly assigned to a route so to retrieve knowledge from its trip time, thus updating his Q-table. The remaining days are **exploited** by the driver according to its utility values. The drivers' departure time is equally distributed along the first hour of the day.

We have performed several tests with various number of drivers to observe their learning process in a route-choice environment. Next, we will analyse two of these tests, one with low density and another with a high density of vehicles. The first one will have a demand of 200 vehicles in an hour, and the second one will have 1900 vehicles in an hour.

For each analysis will be presented two graphs. The top one represents the number of vehicles in each route for each day of the simulation. The bottom one shows the average travel time of each route and also the average total travel time of all travellers, for each day of the simulation .

6.1.1 200 Vehicles Test

The first test we present here has a demand of 200 vehicles, which departure times were equally distributed in the 3600 first seconds of the day. Therefore, each 18 seconds a driver started its journey from point O to point D ($\frac{3600}{200} = 18$).

In Figure 6.5 are shown two plots: on the top is depicted the number of vehicles that travelled through a certain route per day; on the bottom is represented the average travel time for each route and the average travel time computed on all the routes.

Concerning the vehicular dispersion in all routes we can distinguish the **exploration** intervals $[0,50]$ and $[200,250]$ due to their similar distribution in all routes.

In the first exploitation phase we clearly note a preference in route B with respect to the route A. This choice is primarily made due to the segment sequence in each route. In route B the driver does not need to decrease its momentum since the maximum speed allowed is always increasing, whilst in route A the driver needs to break when changing from a faster edge to a slower one.

As route C is faster to get from point O to point D and since we are dealing with a very temporally spaced departure times, it is obvious that this route won't cause any congestion and thereby is chosen by the majority of drivers.

By taking a look in the average travel times we can observe that the drivers surely chose the fastest route, C plot, since their travel time is lower and thus beneficial.

Experiments & Results Discussion

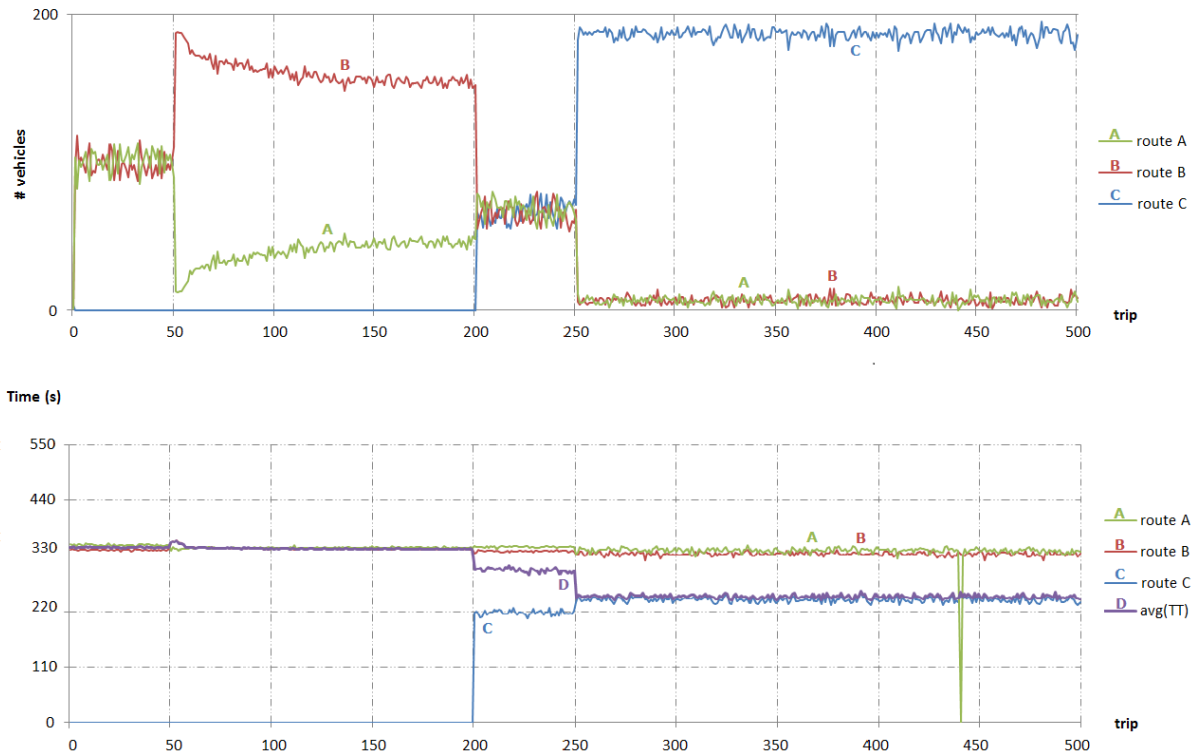


Figure 6.5: Occupation and Travel Time in the 200 Vehicle Test.
Top: Occupation(trip); Bottom: Travel-Time(trip)

6.1.2 1900 Vehicles Test

Another test that we have made was with 1900 vehicle, increasing the flow to approximately one vehicle per each two seconds. The same analysis was made in this test-bed and it is depicted in Figure 6.6

Comparing to the 200's test, in the previous section, the main set-up difference is the flow increment.

This time, the route choice discrepancy, in the first **exploitation** phase, does not occur. The number of vehicles that chose route A or route B is nearly the same, without fluctuations, which establishes a constant average travel time (observed in the bottom graph).

During the second **exploration** phase [200,250], we can state that the average travel time in the new route C is a bit smaller. Hence, in the beginning of the second **exploitation** period the drivers should have a great utility in the choice of C. In fact, we can observe that almost every 1900 vehicles chose to travel through it i.e. route C, **overpassing the initial average travel time**, recorded when there were only 2 routes available. With this insertion, the average travel time increased from approximately 1000 seconds to a staggering 3000 seconds.

The learning Drivers, encountering such a scenario, quickly change their opinion on the utility of route C. They return to their previous choice avoiding the overpopulated route and improving their travel time. We can observe this event in the quick variation in upper graph's peaks in just

Experiments & Results Discussion

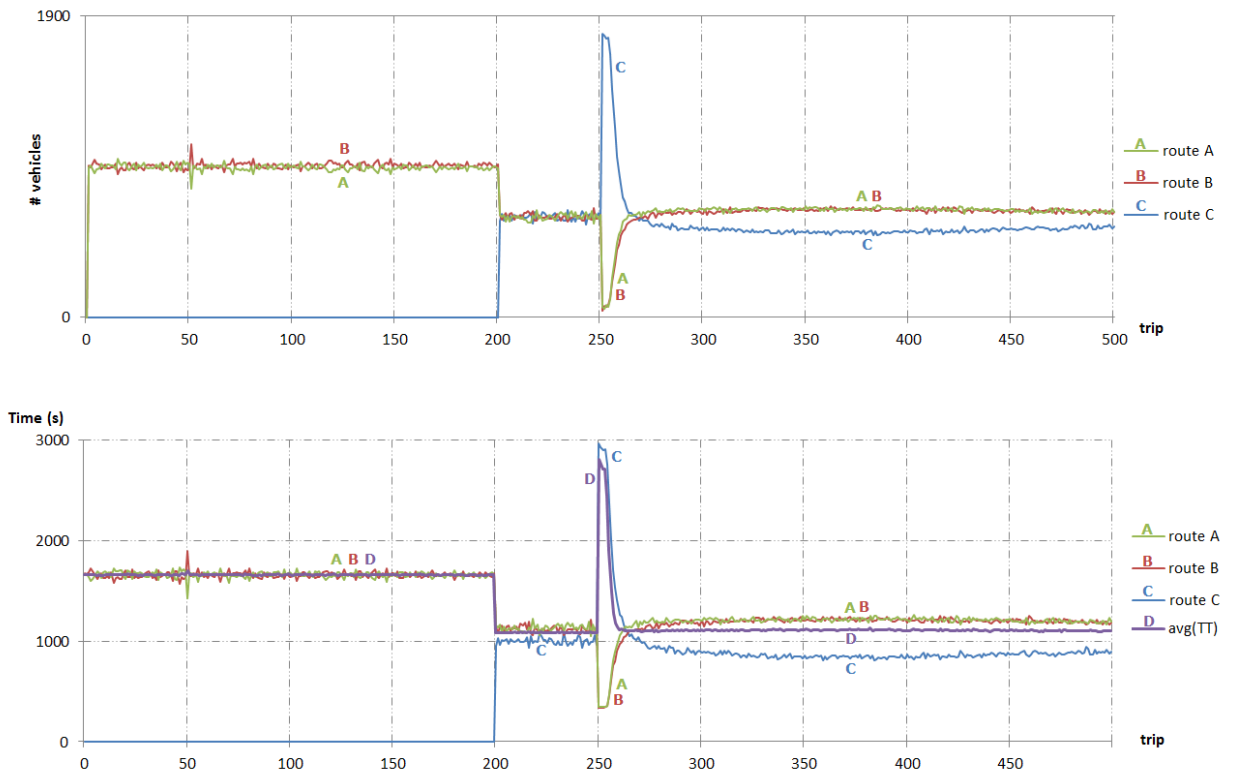


Figure 6.6: Occupation and Travel Time in the 1900 Vehicle Test.
Top: Occupation(trip); Bottom: Travel-Time(trip)

approximately 20 days.

With this learning process the overall travel time diminishes alluding to the **underutilization** of route C, which becomes the less used route, despite being the fastest one.

6.2 En-Route Diversion Experimental Set-up

In this experimental set-up we intend to reproduce the drivers' decision-making process in route choice during their trip under information dissemination.

We have built a small grid network and instantiated in this urban environment 500 vehicles, thus an Artificial Society of drivers constituted by 500 agents, and several ATIS Artefacts distributed through the network.

Moreover, we have implemented a simple accident emulator where the user can choose a vehicle and stimulate its stoppage.

In the beginning of the simulation it is given to each entity of the AS a random origin edge and orientation and a different random destiny. With this, the agents' reactive layer in SUMO can make use of its shortest path algorithms and calculate the best route to accomplish each driver's desire, Figure 6.9 (Top).

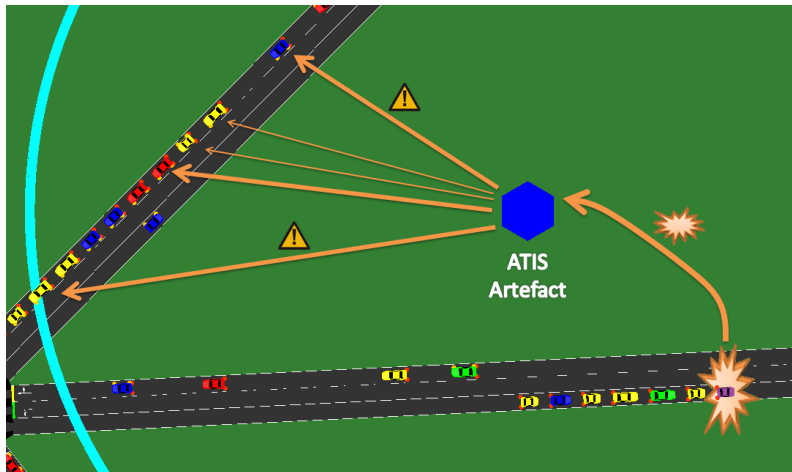


Figure 6.7: ATIS Artefact in Action

As a title of example we are going to consider the vehicle B as a representative of each one of the AS entities involved in this experimentation.

Whilst the driver of vehicle B performs his trip an accident happens in an edge that is going to be used by this driver, meaning, the accident's location belongs to the future edges of the driver's route Figure 6.9 (Middle).

The vehicle A crashes and its driver sends a warning message to one ATIS artefact informing the accident's location.

Afterwards, the several ATIS artefacts inform the driver agents, each of whom will give a special weight to that edge in their own weight table and recalculate the route with this new restriction.

The previously described sequence is depicted in Figure 6.7.

As we can see in the diagram in Figure 6.8 there are several messages exchanged between the various entities involved in this environment.

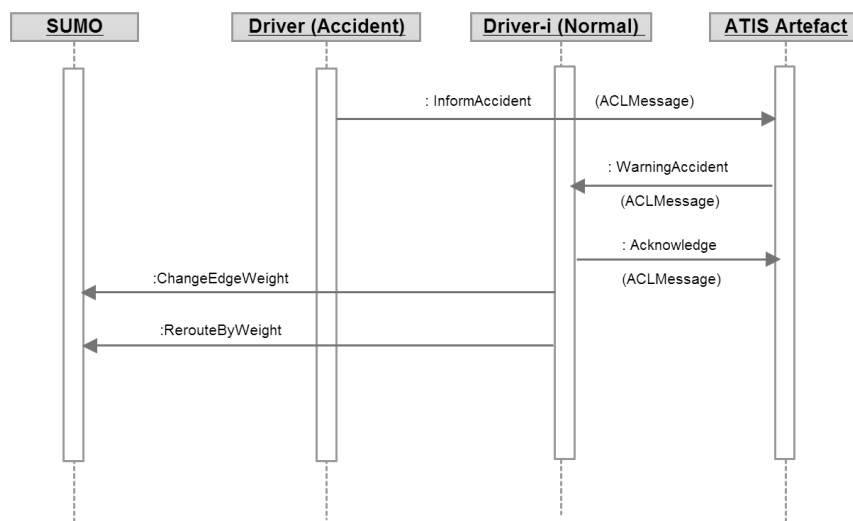


Figure 6.8: ATIS Communication Sequence Diagram

Experiments & Results Discussion

Whenever a traffic setback happens, accident or breakdown, the Driver Agent to whom it occurred, informs the ATIS entities sending an Informative ACL Message with the blocked edge identification.

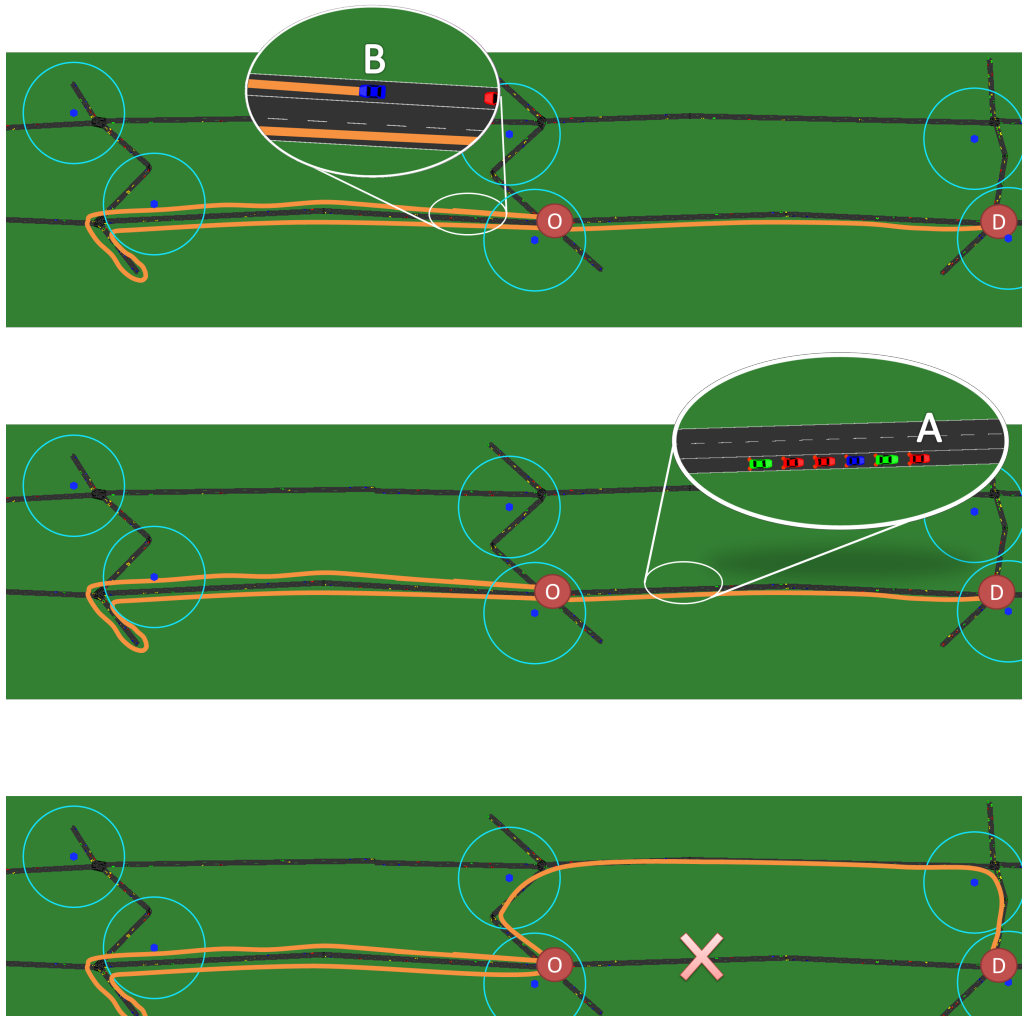


Figure 6.9: En-route Diversion with emulated accident:
Top: SUMO's route computation for vehicle B between Origin O and Destiny D;
Middle: Accident simulated in SUMO - vehicle A stops in a free lane;
Bottom: Re-routing taking into consideration the edges' weights.

Then, the ATIS Artefacts spread the accident information throughout all the drivers that intend to travel that lane, which is acknowledge also through FIPA ACLMessage.

Now, it is up to each Driver's strategical layer to decide if the received information is valuable, up-to-date, or in advance. For example, if the warning edge is still too far from the self, one can hope the congestion to fade away before the driver arrives to the warned place.

Each vehicle, meaning the traffic entity inside the microscopic traffic simulator SUMO, has a edge-weight table in its state. If an edge is too much time or effort consuming the Driver can set

its weight to a higher value.

When called, the route finder shortest path algorithm will take into consideration any value existent in this table. Therefore, if an edge's value is too high, the route will not contain the congested edge in its course.

In Figure 6.9 (Bottom) we can observe the newly computed route after the incident notification to the driver. He avoids the congested edge and circumvents the block.

6.3 Summary

To summarize, we have presented two experimental set-ups using our tool to develop and implement an Artificial Society of Drivers and ATIS Artefacts.

In the first one we replicate the Braes Paradox in a route choice scenario. Here we can notice the capability of the agents in the artificial society of learning and adaptation in adverse and dynamic situations.

The Braess Paradox is a fine example of how easily our intuitions about collective interaction can be deceived. Despite adding a new road to the transportation network it does not grant an improvement in the travellers travel times.

In our tests we've managed to reproduce this idea, using an Artificial Society of Learning Drivers, which had to fulfil a trip choosing from a set of available routes for this purpose.

On one hand we have presented a first experimentation scenario where the vehicle density was too small to evidence the paradox scenario (200 vehicles per hour). Since the departure times were very temporally spaced, the route was not jammed and, thereby chosen by the majority of drivers.

On the other hand we have explored a test-bed with a high density of vehicles, approximately 1 vehicles per each two seconds. In this case, we have noticed the increase of the travel time and the **underutilization** of the newly added route, despite being the fastest one.

In the end we were able to replicate the phenomenon using our integrated framework.

*"Whether one street is preferable to another depends not only on the quality of the road, but also on the **density of the flow**" [BNW05]*

In the latest, we present an en-route diversion due to an induced accident during a simulation.

All the vehicles have a route connecting an Origin to a Destination randomly generated.

The ATIS Artefacts are dispersed in the network infrastructure providing the agent drivers of valuable information concerning vehicle accidents.

The agents receive the accident notification and change their route if it directly affects their journey. This attitude argues in favour with the drivers' awareness and decision-making capabilities.

Chapter 7

Conclusions & Future Work

One of the crucial aspects in the concept & development of ITS/FUT solutions is the importance of Traffic Information.

Accurate and up-to-date traffic information is indeed very important for travellers, in order to arrive faster and safely to their destinations. It is in this field that the ATIS stand out in the ITS. ATIS collect surrounding information and generates knowledge, from its raw data, which will be then broadcasted to travellers, according to individual needs, increasing their level of cognition. This way, it can be used by the traveller in his decision-making process, regarding optimal routes, weather conditions, reduced environmental impacts, among others.

Simulation proved to be an effective approach to analysing and designing novel traffic solutions in socio-technical aspect systems.

Traffic systems have been subject to a lot of improvement last decades and travellers have, in general, witnessed a revolution in the way a trip is planned in urban networks. Hence, facing the current traffic situation in most developed countries it is now imperative to foster new transportation solutions using state-of-the-art technologies towards Future Urban Transport (FUT). User is now central figure in the new vision of urban systems. In this new scene, traditional traffic simulation packages fail to model and represent all aspects of human behaviour in a detailed way.

In general, most simulators follow a macro/microscopic approach as an attempt to improve the representation of traffic flow and management rather the traveller behaviour.

The following sections will raise a critical analysis of the designed and implemented solution towards the initial goals of this dissertation, along with its main results. Lastly, some future developments and perspectives are suggested as well.

7.1 Final Remarks

The main objective of this thesis was to devise a framework of an Artificial Society of Drivers immersed into a realistic urban traffic environment. So that, experimentations with such AS could

provide the community of researchers and practitioners with insights about the formation of emergent mobility patterns and how information or knowledge can affect drivers' decision-making process.

In the literature review, we placed ourselves within the new trending of Future Urban Transports (FUT), where people are positioned as a central aspect, as well their preferences, of the future urban cities. We traced various concepts that are related to the project, namely the synergies between agent technologies and modelling and simulations i.e. Agent simulation and Agent-based simulation and we presented relevant works that have affected this research bounds.

We have devised a conceptual architecture and we built a prototype of Artificial Societies in traffic scenario, where we experimented the knowledge representation of the network using Reinforcement Learning techniques and we instantiated the way artefacts, that generate information, can affect the drivers' decision making process, whether in route choice or in planning activities. We used the concept of delegate agent, where the agent uses a tactic and strategic layers for reasoning .

Although we have managed to build a flexible tool to instantiate a synthetic society over a traffic domain, we cannot claim that we yielded to produce a large-scale real Agent Based Simulation. That is, using a large population of agents controlling an equally large population of vehicles.

The drawback is due to the limitation of the simulation software, which despite of having a solid API for external access it evidences in this interface a bottleneck in the communication channel with SUMO. This is an architectural limitation of the software package and not of the proposed architectural design.

We believe that, the use of a mature commercial traffic simulator solution we are able to deploy a truly large-scale Agent-Based Simulation (ABS), having the Artificial Society interfaced with the population of drivers as its main goal.

7.2 Contributions

We proposed a framework where MAS of different nature can be instantiated over the traffic domain, meaning socio-technical systems, embedded intelligent artefacts, aiming to design more human-centric, economic and environmental solutions.

This tool also reveals great flexibility for multi-agent systems development in traffic domains, since one can easily develop his own synthetic population, where each agents is presented with its own preferences and beliefs. Such AS can thus be used to design solutions based in individual or collective intelligence and participation (social-awareness) or as a test-bed for policy evaluation by governmental institutions.

We have extended TraSMAPI's communication module to SUMO, providing world-wide community a fully capable high-level API for traffic domain control and management in this microscopic traffic simulator.

A powerful MAS development platform, JADE, was coupled to TraSMAPI, simplifying the development of multi-agent systems while guaranteeing standard compliance with the FIPA specifications.

Besides externally attaching the ATIS Artefact to the SUMO microscopic traffic simulator, we have also managed to develop and deploy the concept of delegate agent in SUMO, deploying the reactive layer of the agent in SUMO's traffic simulation, and the strategical, cognitive layer in the MAS developing framework.

In a general view we provide a tool very useful both to the multi-agent simulation and transportation community, mainly because this framework allows different types of studies, namely: control algorithms, service design, but also studies in policy supervising evaluation, incentive mechanisms appraisal and Vehicle-to-Vehicle communication applications.

7.3 Future Developments

Giving the complexity of this framework, the next step is to solidify its structure, primarily in SUMO's communication module. This microscopic traffic simulator only provides a single communication port, thus bottlenecking the communication channel. One major improvement is to extend the SUMO's API to support multiple port connections, this way distributing control and management queries to the simulation.

Related to the previous topic is the scalability. Further and deeper tests should be made about the number of entities that the Artificial Society can manage in a simulation scenario, in one or several machines, taking advantage of the distribution capability of JADE. However, these test should only be performed when the aforementioned issue is resolved, since it is a known bottleneck situation.

Other improvements are related to the further expansion of TraSMAPI's simulator range. This way, testing this framework approach in other microscopic simulators, like AIMSUN.

7.4 Future Works

Taking into account the resulting framework, numerous of applications can emerge from this.

Aiming realistic simulations, new and more truthful visualization frameworks could be coupled.

Moreover, a possible integration with a Serious Game driver simulator will be used, to provide a Human-in-the-loop simulation, to appraise and validate the user's decision-making process when receiving information.

Interface our framework with a data-base that provides demographic information about the population for a better representation of the AS is also a possible future work. Furthermore, a data-base of additional networks, edge or nodes, could provide an agent delegate of the simulation information, in order to expand or modify the network topology on the fly and continuing the

Conclusions & Future Work

simulation from the state it was stopped. Thus adding to our simulation another dimension under the umbrella of agent-directed simulation (ADS).

Using the MAS development framework, researchers can study and create better and more complex algorithms either for traffic management or route planning.

Researcher and practitioners can, not only, build and appraise Vehicle-to-X scenarios, but also, study new policies development and incentive mechanisms for improving aspects of the urban system.

Concerning the architectural design, future developments in trending cloud based architecture could be performed using simulation as a service. Hence, enabling large scale simulations with thousands of hybrid agents also allowing designers of different systems to participate on the same scenario.

References

- [BBEK11] M Behrisch, L Bieker, J Erdmann, and D Krajzewicz. SUMO - Simulation of Urban MObility: An Overview. In *SIMUL 2011, The Third International Conference on Advances in System Simulation*, pages 63–68, Barcelona, Spain, 2011.
- [BC86] R A Brooks and J H Connell. Asynchronous distributed control system for a mobile robot. In *Proceedings SPIE*, pages 77–84. Cambridge, 1986.
- [BCPR03] F Bellifemine, G Caire, A Poggi, and G Rimassa. JADE-A White Paper, Sept. 2003. *Online document accessible under <http://jade.tilab.com/papers/2003/WhitePaperJADEEXP.pdf> (last access: March 22, 2008)*, 2003.
- [BCSH89] W Barfield, L Conquest, J Spyridakis, and M Haselkorn. Information requirements for real-time motorist information systems. In *Vehicle Navigation and Information Systems Conference, 1989. Conference Record*, pages 101–104. IEEE, 1989.
- [BdSB10] Ana L C Bazzan, Maicon de Brito do Amarante, Tiago Sommer, and Alexander J Benavides. {ITSUMO}: an Agent-Based Simulator for {ITS} Applications. In Rosaldo Rossetti, Henry Liu, and Shuming Tang, editors, *Proc. of the 4th Workshop on Artificial Transportation Systems and Simulation*. IEEE, 2010.
- [BNW05] D Braess, A Nagurney, and T Wakolbinger. On a paradox of traffic planning. *Transportation science*, 39(4):446–450, 2005.
- [Bon02] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, 99(Suppl 3):7280–7287, 2002.
- [BPR99] F Bellifemine, A Poggi, and G Rimassa. JADE-A FIPA-compliant agent framework. In *Proceedings of PAAM*, volume 99, page 33. London, 1999.
- [Bra68] D Braess. Über ein Paradoxon aus der Verkehrsplanung. *Mathematical Methods of Operations Research*, 12(1):258–268, 1968.
- [Car] Henrique Lopes Cardoso. In readings: Crash Course Jade 4.0.1 - <https://oldmoodles.fe.up.pt/1112/file.php/3595/JADECrashCourse.pdf>.
- [CC10] B Chen and H H Cheng. A review of the applications of agent technology in traffic and transportation systems. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2):485–497, 2010.
- [CCM10] Bo Chen, Harry H Cheng, and Senior Member. A Review of the Applications of Agent Technology in Traffic and Transportation Systems. 11(2):485–497, 2010.

REFERENCES

- [CCP06] B Chen, H H Cheng, and J Palen. Mobile-C: a mobile agent platform for mobile C/C++ agents. *Software: Practice and Experience*, 36(15):1711–1733, 2006.
- [CMV06] Caspar G. Chorus, Eric J. E. Molin, and Bert Van Wee. Use and effects of Advanced Traveller Information Services (ATIS): a review of the literature. *Transport Reviews*, 26(2):127–149, March 2006.
- [Col94] J S Coleman. *Foundations of social theory*. Belknap Press, 1994.
- [DF94] A Drogoul and J Ferber. Multi-agent simulation as a tool for modeling societies: Application to social differentiation in ant colonies. *Artificial Social Systems*, pages 2–23, 1994.
- [DN11] H Dawid and M Neugart. Agent-based models for economic policy design. *Eastern Economic Journal*, 37(1):44–50, 2011.
- [EAP96] J M Epstein, R Axtell, and 2050 Project. *Growing Artificial Societies: Social Science from the Bottom Up*. Complex Adaptive Systems. Brookings Institution Press, 1996.
- [FG97] S Franklin and A Graesser. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. *Intelligent Agents III Agent Theories, Architectures, and Languages*, pages 21–35, 1997.
- [Flo09] G Flotterod. Cadyts—a free calibration tool for dynamic traffic simulations. In *Proceedings of the 9th Swiss Transport Research Conference*, 2009.
- [Fro97] Rejane Frozza. *Simula : ambiente para desenvolvimento de sistemas multiagentes reativos*. PhD thesis, Universidade Federal do Rio Grande do Sul, 1997.
- [Gaw99] C Gawron. *Simulation-based traffic assignment—computing user equilibria in large street networks*. PhD thesis, PhD Thesis, University of Cologne, 1999.
- [GSTCM03] A M Garcia-Serrano, D Teruel Vioque, F Carbone, and V D Méndez. FIPA-compliant MAS development for road traffic management with a knowledge-based approach: the TRACK-R agents. In *Challenges in Open Agent Systems {Ö}03 Workshop, Melbourne, Australia*, 2003.
- [HDB98] M C Hulse, T A Dingus, and W Barfield. Description and applications of advanced traveler information systems. *Human factors in intelligent transportation systems*, pages 1–30, 1998.
- [HOGS02] J Z Hernández, S Ossowski, and A Garcia-Serrano. Multiagent architectures for intelligent traffic management systems. *Transportation Research Part C: Emerging Technologies*, 10(5):473–506, 2002.
- [JS03] J Jijina and J M Stefan. Method and system for mobile vehicle re-routing, 2003.
- [KBM⁺] Daniel Krajzewicz, Elmar Brockfeld, Jürgen Mikat, Julia Ringel, Christian Rössel, Wolfram Tuchscheerer, Peter Wagner, and Richard Wösler. Simulation of modern Traffic Lights Control Systems using the open source Traffic Simulation SUMO.
- [KHRW02] D Krajzewicz, G Hertkorn, C Rössel, and P Wagner. Sumo (simulation of urban mobility). In *Proc. of the 4th Middle East Symposium on Simulation and Modelling*, pages 183–187, 2002.

REFERENCES

- [Klu] Franziska Klugl. *Agent-Based Simulation Engineering*. PhD thesis, Würzburg University, Germany.
- [Kra98] Stefan Krauß. Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics. *thesis and dissertation*, (319):115, 1998.
- [KWG97] S Krauss, P Wagner, and C Gawron. Metastable states in a microscopic model of traffic flow. *Physical Review E*, 55(5):5597, 1997.
- [La09] Paulo Leitão. Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence*, 22(7):979–991, 2009.
- [LLL⁺05] Malcolm Yoke Hean Low, Kong Wei Lye, Peter Lendermann, Stephen John Turner, Reman Tat Wee Chim, and Surya Hadisaputra Leo. An agent-based approach for managing symbiotic simulation of semiconductor assembly and test operation. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, AAMAS '05, pages 85–92, New York, NY, USA, 2005. ACM.
- [LR02] F Logi and S G Ritchie. A multi-agent architecture for cooperative inter-jurisdictional traffic congestion management. *Transportation Research Part C: Emerging Technologies*, 10(5):507–527, 2002.
- [LTWW07] Jinyuan Li, Shuming Tang, Xiqin Wang, and Fei-Yue Wang. A software architecture for artificial transportation systems - principles and framework, 2007.
- [Mae95] Pattie Maes. Artificial life meets entertainment: lifelike autonomous agents. *Commun. ACM*, 38(11):108–114, 1995.
- [Mas98] T Mast. *Introduction to ITS*. Erlbaum, Mahwah, NJ, 1998.
- [McC] Thomas McCluskey. ARTS Towards Autonomic Road Transport Support Systems.
- [Mit84] W L Mitchell. Traffic light control for emergency vehicles, 1984.
- [MN10] C M Macal and M J North. Tutorial on agent-based modelling and simulation. *Journal of Simulation*, 4(3):151–162, 2010.
- [MSTR12] J Macedo, M Soares, I Timoteo, and R J F Rossetti. An approach to advisory-based traffic control, 2012.
- [MT07] Lisa Jean Moya and Andreas Tolk. Towards a taxonomy of agents and multi-agent systems. In *Proceedings of the 2007 spring simulation multiconference - Volume 2*, SpringSim '07, pages 11–18, San Diego, CA, USA, 2007. Society for Computer Simulation International.
- [MW06] J C Miles and A J Walker. The potential application of artificial intelligence in transport, 2006.
- [NRW04] M Nickles, M Rovatsos, and G Weiss. *Agents and computational autonomy: potential, risks, and solutions*, volume 2969. Springer, 2004.

REFERENCES

- [Oli12] Eugénio Oliveira. A Structured Environment to Facilitate Agreements. In G Vouros S. Ossowski, F. Toni, editor, *Proceedings of the First International Conference on Agreement Technologies (AT 2012), CEUR Workshop Proceedings*, pages 351–352, Dubrovnik, Croatia, October 15-16, 2012, 2012.
- [ONU⁺00] T I Ören, S K Numrich, A M Uhrmacher, L F Wilson, and E Gelenbe. Agent-directed simulation: challenges to meet defense and civilian requirements. In *Proceedings of the 32nd conference on Winter simulation*, pages 1757–1762. Society for Computer Simulation International, 2000.
- [OY12] Tuncer Ören and Levent Yilmaz. Synergies of simulation, agents, and systems engineering. *Expert Systems with Applications*, 39(1):81–88, 2012.
- [PD09] L G Papaleondiou and M D Dikaiakos. Trafficmodeler: A graphical tool for programming microscopic traffic simulators through high-level abstractions. In *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, pages 1–5. IEEE, 2009.
- [PR12] José L F Pereira and Rosaldo J F Rossetti. An integrated architecture for autonomous vehicles simulation. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing, SAC '12*, pages 286–292, New York, NY, USA, 2012. ACM.
- [PRK11] L S Passos, R J F Rossetti, and Z Kokkinogenis. Towards the next-generation traffic simulation tools: a first appraisal, 2011.
- [PRL⁺08] M Piórkowski, M Raya, A Lezama Lugo, P Papadimitratos, M Grossglauser, and J.-P. Hubaux. TraNS: realistic joint traffic and network simulator for VANETs. *SIGMOBILE Mob. Comput. Commun. Rev.*, 12(1):31–33, 2008.
- [RBB⁺02] R J F Rossetti, R H Bordini, A L C Bazzan, S Bampi, R Liu, and D V Vliet. Using BDI agents to improve driver modelling in a commuter scenario. *Transportation Research Part C: Emerging Technologies*, 10(5):373–398, 2002.
- [Rep] F I G Report. *The Need for Spatial Information Management Rapid Urbanization and Mega Cities* .: Number 48.
- [RG92] A S Rao and M P Georgeff. An abstract architecture for rational agents. In B Nebel, C Rich, and R Swartout, editors, *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR&R-92)*, pages 439–449. M. Kaufmann Publishers, 1992.
- [RNC⁺95] S J Russell, P Norvig, J F Canny, J M Malik, and D D Edwards. *Artificial intelligence: a modern approach*, volume 2. Prentice hall Englewood Cliffs, NJ, 1995.
- [Roo99] D A Roozmond. Using autonomous intelligent agents for urban traffic control systems. *Proc. 6th World Congr. Intell. Transp. Syst.*, 1999.
- [Ros] Rosaldo J.F. Rossetti. *A bdi-based approach for the assessment of drivers' decision-making in commuter scenarios*. Doctoral thesis, Universidade Federal do Rio Grande do Sul.

REFERENCES

- [RSRM10] David Rieck, Björn Schünemann, Ilja Radusch, and Christoph Meinel. Efficient traffic simulator coupling in a distributed V2X simulation environment. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, SIMUTools '10*, pages 72:1—72:9, ICST, Brussels, Belgium, Belgium, 2010. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [Sch71] T C Schelling. Dynamic models of segregation†. *Journal of mathematical sociology*, 1(2):143–186, 1971.
- [SF00] O Simonin and J Ferber. Modeling self satisfaction and altruism to handle action selection and reactive cooperation. In *The Sixth International Conference on the Simulation of Adaptive Behavior FROM ANIMALS TO ANIMATS*, volume 6, pages 314–323, 2000.
- [Sho93] Yoav Shoham. Agent-oriented programming. *Artif. Intell.*, 60(1):51–92, 1993.
- [SR10] L Sanchez Passos and R Rossetti. Traffic light control using reactive agents. In *Information Systems and Technologies (CISTI), 2010 5th Iberian Conference on*, pages 1–6. IEEE, 2010.
- [TARO10] Ivo J.P.M. Timoteo, Miguel R. Araujo, Rosaldo J.F. Rossetti, and Eugenio C. Oliveira. TraSMAPI: An API oriented towards Multi-Agent Systems real-time interaction with multiple Traffic Simulators. *13th International IEEE Conference on Intelligent Transportation Systems*, pages 1183–1188, September 2010.
- [TARO12] IvoJ.P.M. Timóteo, MiguelR. Araújo, RosaldoJ.F. Rossetti, and EugénioC. Oliveira. Using TraSMAPI for the assessment of multi-agent traffic management solutions. *Progress in Artificial Intelligence*, 1(2):157–164, 2012.
- [TU09] Andreas Tolk and Adelinde M Uhrmacher. Agents: Agenthood, Agent Architectures, and Agent Taxonomies. In *Agent-Directed Simulation and Systems Engineering*, pages 73–109. Wiley-VCH Verlag GmbH & Co. KGaA, 2009.
- [Tyr93] T Tyrrell. The use of hierarchies for action selection. *Adaptive Behavior*, 1(4):387–420, 1993.
- [UW09] A M Uhrmacher and D Weyns. Multi-agent systems: simulation and applications. *Computational analysis synthesis and design of dynamic models series (4)*, 2009.
- [VMvBvK03] A T Van den Bosch, M R Menken, M van Breukelen, and R T van Katwijk. A test bed for multi-agent systems and road traffic management. In *Proceedings of the 15th Belgian-Netherlands Conference on Artificial Intelligence (BNAIC'03)*, pages 43–50, 2003.
- [WBKS00] Joachim Wahle, Ana Lúcia C Bazzan, Franziska Klügl, and Michael Schreckenberg. Decision dynamics in a traffic scenario. *Physica A: Statistical Mechanics and its Applications*, 287(3-4):669–681, December 2000.
- [WBKS02] Joachim Wahle, Ana Lúcia C Bazzan, Franziska Klügl, and Michael Schreckenberg. The impact of real-time information in a two-route scenario using agent-based simulation. *Transportation Research Part C: Emerging Technologies*, 10(5–6):399–417, October 2002.

REFERENCES

- [WD92] C.J.C.H. Watkins and P Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.
- [Wei00] G Weiss. *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press, 2000.
- [WJ95] Michael Wooldridge and Nicholas Jennings. Agent theories, architectures, and languages: A survey. In Michael Wooldridge and Nicholas Jennings, editors, *Intelligent Agents*, volume 890 of *Lecture Notes in Computer Science*, pages 1–39. Springer Berlin / Heidelberg, 1995.
- [WJO95] M Wooldridge, N R Jennings, and Others. Intelligent agents: Theory and practice. *Knowledge engineering review*, 10(2):115–152, 1995.
- [WOO07] D Weyns, A Omicini, and J Odell. Environment as a first class abstraction in multiagent systems. *Autonomous agents and multi-agent systems*, 14(1):5–30, 2007.
- [WPR⁺08] Axel Wegener, Michał Piórkowski, Maxim Raya, Horst Hellbrück, Stefan Fischer, and Jean-Pierre Hubaux. TraCI: an interface for coupling road traffic and network simulators. In *Proceedings of the 11th communications and networking simulation symposium, CNS '08*, pages 155–163, New York, NY, USA, 2008. ACM.
- [YO07] Levent Yilmaz and Tuncer I Ören. Agent-directed simulation systems engineering. In *Proceedings of the 2007 Summer Computer Simulation Conference, SCSC '07*, pages 897–904, San Diego, CA, USA, 2007. Society for Computer Simulation International.
- [ZA08] K G Zografos and K N Androustopoulos. Algorithms for Itinerary Planning in Multimodal Transportation Networks. *Intelligent Transportation Systems, IEEE Transactions on*, 9(1):175–184, 2008.
- [ZV03] F Zambonelli and H Van Dyke Parunak. Signs of a revolution in computer science and software engineering. *Engineering Societies in the Agents World III*, pages 120–125, 2003.
- [ZWSF07] Kun Zhang, Jeremy Woolley, Branko Stazic, and C A Forster. An effective tool for advanced traveller’s information systems development. In *State of Australian Cities National Conference, 2007, Adelaide, South Australia*, pages 434–443, 2007.