Location based services and location based behavior in a smart city

Chen Wang

N^o d'ordre NNT : 2016LYSEC017

THESE de DOCTORAT DE L'UNIVERSITE DE LYON

opérée au sein de l'Ecole Centrale de Lyon

Ecole Doctorale INFOMATHS ED N^o 512

Spécialité: Informatique

Soutenue publiquement le 27/05/2016, par:

Chen WANG

Location Based Services and Location Based Behavior in a Smart City

Devant le jury composé de :

Franck TARPIN-BERNARD	Professeur - Université de Grenoble	Président
Christophe KOLSKI	Professeur - Université de Valenciennes	Rapporteur
Ahmed SEFFAH	Professeur - Lappeenranta University of Technology	Rapporteur
Hélène TATTEGRAIN	Directrice - Laboratoire LESCOT	Examinatrice
René CHALON	Maître de Conférences - Ecole Centrale de Lyon	Examinateur
Chuantao YIN	Associate Professor - Beihang University	Co-directeur de thèse
Bertrand DAVID	Professeur - Ecole Centrale de Lyon	Directeur de thèse

Acknowledgements

Firstly, I want to express my gratitude to Professor Bertrand DAVID, my supervisor of thesis. He is open minded, thoughtful, humorous. Doing research under his guidance gives me the feeling of "free to fly". He is good at using heuristic methods, when giving comments and suggestions on my research work, so that he is not only an expert in Human-Computer-Interaction domain, but also adept in "Human-Human-Interaction". He sent me to Web of Things summer school, Internet of Things exhibition, HCI international conference, *etc.*, which expanded my knowledge and understanding of related topics of research. I am lucky and happy to be the last PhD student of Bertrand before his retirement.

Then I would like to thank Chuantao YIN, my co-supervisor, for his recommendation of me to Bertrand. During the application stage of the thesis proposal, I almost missed the deadline and it was Chuantao who helped me a lot in finalizing the discussion with Bertrand for the proposal of the thesis. He also shared with me his experience of doing research, and encouraged me with advices on the methodologies.

I also want to thank René CHALON, who works together with Bertrand in Ecole Centrale de Lyon, for reading and modifying our papers for conference or journal submission. He is serious and careful, and also helped me with user acceptability tests and analysis in my research work.

Support and help from colleagues and friends are something I also appreciate a lot. Ivan DAL PIO LUOGO, Gaël DUPERREY, Loïc MARTINEZ, Bertin NAYO, Thai-An GUYON have done their final internship of CNAM in the laboratory and since they have already several years of real working experience, I learned a lot from them through discussions about life and work as well as other cultural stuff (also got chance to practice my French). Other former members of the laboratory such as Tao XU, Yun ZHOU, Bingxue Zhang, Huiliang JIN, have also contributed to the friendly and vivid ambiance of the lab.

Thanks to my family and my wife, Ying, who always support me during the hard times.

Thanks to China Scholarship Council (CSC) and Behihang University that provide financial support and selected me to participate in the CSC-Centrale program for PhD study in France.

Contents

List of Tables

List of Figures

Abstract

The concept of "Smart Cities" has emerged during the last few years to describe how investments in human and social capital and modern Information and Communication Technologies (ICT) infrastructure and e-services fuel sustainable growth and quality of life, enabled by a wise management of natural resources and through participative government. To us, Smart City is a real augmented environment allowing ubiquitous computing, with up-to-date web 2.0, which is collaborative, mobile and contextual, human actors as well as different things (connected objects) are now an integral part of internet. In the international France-China project on Smart City we used the MOCOCO approach (Mobility, Contextualization, Collaboration) to conduct research work with multiple applications in working, learning and social situations; professional and home working situations, professional and teenager contextual mobile learning situations as well as Smart City applications are taken into account – transportation, goods distribution and local sport and cultural activities.

This dissertation focuses on Location Based Services, and Internet of Things, which are both important aspects of Smart City. The choice of dynamic management of road lanes as a case study in this thesis, is also a good practice of integrating new technologies to make the city smarter and to make our life more comfortable. According to Schiller and Voisard (2004), Location Based Services (LBS) can be defined as services that integrate a mobile device's location or position with other information so as to provide added value to a user. During recent years, LBS has evolved from simple GIS applications and positioning of emergent phone callers to more complicated, proactive, application-oriented services adapted to different users. However, heterogeneity of devices, data management and analysis, and HCI aspects are always main challenges for LBS. Our goal is to make the LBS meet the requirements of Smart City, with use of Internet of Things (IoT), integrating a certain degree of ambient intelligence.

The theoretical aspect of our contributions is that we examine at component level the possibility and feasibility of using IoT to better support LBS. The ability of IoT architecture of integrating various objects gives LBS a better management of location-aware devices; the sensors can also enrich the data source of LBS. The middleware of IoT, good at objects abstraction and service composition, provides possibilities to deploy more intelligent and customized service components, thus can enhance the middleware of LBS.

The practical aspect of our contributions is that we choose a dynamic lane management problem as a use case study demonstrating our approach in regard to combining LBS with IoT for a Smart City application. The goal of the dynamic lane management system is to make a better use of road lanes by dynamic allocation of lanes to different types of transportation. We provide the system architecture, user interfaces and a simulation environment to validate the solution design. We also develop a proof of concept to validate the technological aspects of the lane management system.

The simulation environment of the lane management system is another important part of our contributions, it includes a core simulator to simulate the function of the system and the behaviors of the vehicles, and an editor of scenario and a generator of traffic as initialization tools. Different visualization methods of simulation results are also taken into consideration. In addition, we develop an evaluation tool which allows for real time user interaction, based on the visualization of the results of the simulator to conduct user tests for HCI aspects, as human factors should always be considered in the context of Smart City.

The dissertation is organized as follows: Chapter 2 surveys related works and major references related to our topic of research; Chapter 3 describes the solution design to the dynamic lane management system and an overview of the simulation environment; Chapter 4 introduces the core simulator as well as the proof of concept; Chapter 5 shows different forms of visualization and presents our user tests based on an evaluation tool; Chapter 6 concludes the dissertation and shows our perspectives of future work.

Keywords: smart city, location-based services, internet of things, traffic simulation

Résumé

Le concept de "Smart Cities" a émergé au cours des dernières années pour décrire comment les investissements dans le capital humain et social, et dans les technologies de la communication (TIC) infrastructures et services électroniques peuvent maintenir la croissance durable et la qualité de vie, par une gestion judicieuse des ressources naturelles et par un gouvernement participatif. Pour nous, Smart City est un environnement réel augmenté permettant l'informatique ubiquitaire, avec web 2.0, qui est collaborative, mobile et contextuelle, les acteurs humains, ainsi que des objets connectés faisant désormais partie intégrante de l'Internet. Dans le context de notre projet international France-Chine sur Smart City, nous avons utilisé une approche MOCOCO (Mobilité, Contextualisation, Collaboration) à mener des travaux de recherche avec de multiples applications dans des situations de travail professionnels et à domicile, des situations d'apprentissage mobile contextuelles, ainsi que des applications de Smart City sont prises en compte - le transport, la distribution des marchandises, et des activités sportives et culturelles.

Cette thèse se concentre sur les services basés sur la localisation (LBS), et Internet des Objets (IdO), qui sont deux aspects importants de Smart City. Le choix de la Gestion Dynamique des Voies de Circulation comme une étude de cas dans cette thèse est également une bonne pratique d'intégration de nouvelles technologies pour rendre la ville plus intelligente et pour rendre notre vie plus confortable. Selon Schiller et Voisard (2004), les services basés sur la localisation peuvent être définis comme des services qui intègrent l'emplacement ou la position d'un dispositif mobile avec d'autres informations afin de fournir une valeur ajoutée à un utilisateur. L'objectif est d'utiliser la mise en œuvre IdO pour améliorer LBS, fournissant l'intelligence ambiante et d'assurer la facilité d'utilisation pour usagers dans des situations dynamiques.

L'aspect théorique de nos contributions est que nous examinons la possiblilité et la faisabilité de l'utilisation de l'IdO pour augmenter LBS. L'architecture de l'IdO a une capacité d'intégrer divers objets, ce qui fournit à LBS une meilleure gestion des dispositifs de géolocalisation; l'intergiciel de l'IdO, capable de faire l'abstraction des objets et la composition de services, donne la possibilité de déployer des composants de service plus intelligents et personnalisés, ainsi peut améliorer l'intergiciel de LBS.

L'aspect pratique de nos contributions est que nous avons choisi une problématique de gestion dynamique des voies comme une étude de cas, validant notre approche d'utiliser l'IdO pour augmenter LBS dans une application de Smart City. L'objectif du système de gestion dynamique des voies est d'assurer une meilleure utilisation de voie de circulation par l'allocation dynamique de voies à différents types de transport. Nous avons fourni l'architecture du système du point de vue de TIC, et un environnement de simulation pour valider la conception de la solution. Nous avons également développé une preuve de concept pour valider les aspects technologiques du système.

L'environnement de simulation comprend un simulateur pour simuler la fonction du système et les comportements des véhicules, un éditeur de scénario, et un générateur de trafic en tant qu'outils d'initialisation. Différentes formes de visualisation de résultats de simulation sont également prises en compte. En outre, nous avons développé un outil d'évaluation basé sur la visualisation en 3D, qui permet l'interation entre l'utilisateur et l'outil en temps réel, pour effectuer des tests d'utilisation comme l'étude des aspects IHM, puisque les facteurs humains devraient toujours être mis en premiers dans le contexte de Smart City.

Cette thèse est organisée comme le suivant: Chapitre 2 présente l'état de l'art; Chapitre 3 décrit la conception du système de gestion dynamique des voies ansi qu'un aperçu de l'environnement de simulation; Chapitre 4 présente le simulateur et la maquette; Chapitre 5 montre de différentes formes de visualisation et présente nos tests utilisateurs; Chapitre 6 conclut la thèse et donne nos perspectives des travaux futurs.

Mots clés: smart city, services basés sur localisation, internet des objets, simulation de trafic

Chapter 1

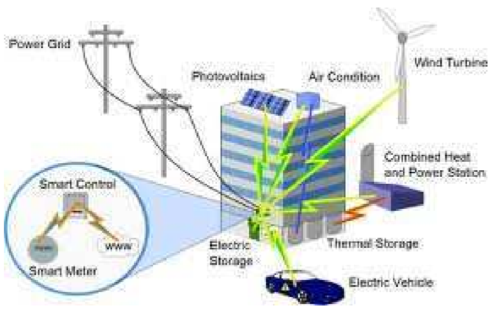
Introduction

1.1 Smart City and Ubiquitous Computing

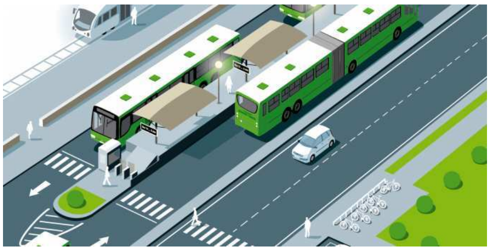
The concept of "Smart Cities" has emerged in the last few years to describe how investments in human and social capital and modern Information and Communication Technologies (ICT) infrastructure and e-services fuel sustainable growth and quality of life, enabled by wise management of natural resources and through participative government [David *et al.* 2012].

According to [Caragliu *et al.* 2011], a smart city can be defined along six dimensions: smart economy, smart mobility, smart environment, smart people, smart living and smart government. Each dimension includes some factors that can further describe the idea of them. For instance, smart mobility could comprise international or national accessibility, availability of ICT infrastructure, and sustainable, innovative and intelligent transportation systems. The smart city originates from the concept of "Smarter Planet" which was put forward by IBM in November 2008 ([IBM 2008]). They attempt to inject the new generation of information technologies into the business, government and civil society of the city. They would also like to install sensors in objects in a complicated system (*e.g.* a grid network), to monitor its status and connect all the sensors together to the internet. Then, the central super computer or cloud computing resources could manage activities, living standard and production in a finer way.

At present, the Smart City is still a concept undergoing evolution and experimentation. It aims at highlighting the role of ICT in a modern city, and at integrating and optimizing the resources of a city, to make city life more efficient, energy-economic and intelligent ([Jin 2014]). Rather than focusing only on how



(a) Smart buildings[1]



(b) Smart transportation[2]

(c) Smart home[3]

Figure 1.1: Examples of Smart City application area

technologies can change the city, which is the case of "digital cities" or "intelligent cities", the smart city also pays attention to non-technological aspects, such as social activity, environment, energy, *etc.*

David et al. described a smart city bus stop named "ABRI+Bus Shelter" ([David & Chalon 2010]) supported by Location-Based Services (LBS). This is a hotspot based system which uses a mobile network for communication between bus drivers and passengers to better serve passengers, especially those who have special needs (disabled people, people carrying bicycles, *etc.*). Moreover, the bus shelter contains an electronic display board to display local related information about shopping, cultural and sport events, which is the neighborhood life-oriented communication and cooperation hotspot. User needs and human participation cannot

[1]Figure source: `http://www.smartgridcontest.com/idea.php?id=145`, last checked available on Feb 18th 2016

[2]Figure source: `http://newsroom.scania.com/en-group/2009/11/25/buses-the-base-in-sustainable-city-transports/`, last checked available on Feb 18th 2016

[3]Figure source: `http://www.alibaba.com/product-free/107093675/Ekon_Smart_Home_phone/showimage.html`, last checked available on Feb 18th 2016

be ignored, because the smart city is not only the application of new information technologies, but also concerns the participation of citizens in the various activities of the city with human intelligence.

Smart city large scale projects have been launched, in addition to academic research. The Smart City Lyon project([Metropole 2014]), launched by the Grand Lyon Bureau, encourages the development of innovative services for the next generation of cities and is a test bed for promoting experiments. The blue print of Smart City Lyon includes:

- taking into account environmental challenges and energy constraints;
- promoting the networking approach of stakeholders with each other: local authorities, residents and businesses;
- allowing the switch from ownership to use: users' involvement in the design of products and services;
- integrating new technologies (IT and communication, robotics, smart transportation systems, *etc.*).

Though the smart city is gaining increasingly more attention, there are still no successful accurate models to follow. Researchers often have to imagine and design more interesting and promising scenarios of application. However, one of the challenges that the smart cities are facing, is how to bring innovative technologies or concepts to the common appliances or daily life which citizens are already familiar with. This is because it takes time for the latter to accept such changes and user habits cannot be changed either easily or quickly. In this process, while technologies or concepts become less visible to users, users are gradually being immersed in an ubiquitous environment. According to the descriptions of Mark Weiser ([Weiser 1999]), *The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.* But the "disappearing devices" could be defined as all devices in an ubiquitous environment connected with each other seamlessly, so that users can focus on the task they want to do, rather than focus on which devices they should use.

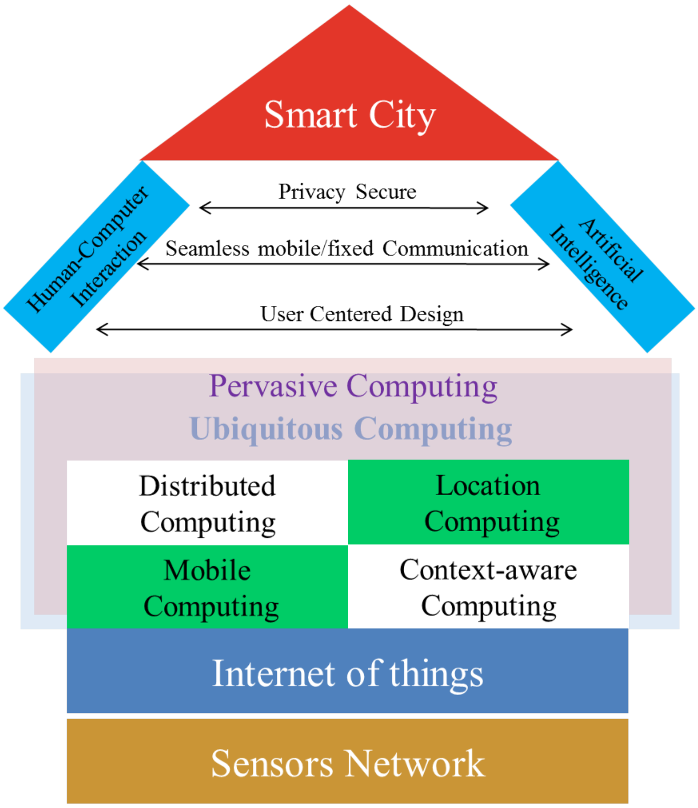


Figure 1.2: Relationship among different concepts in Smart City paradigm

In other words, ubiquitous computing is in fact a user-centric and context-aware interactive environment, based on seamless communication of various in-environment devices, to assist users in completing specific tasks more efficiently or to offer users more intelligent services.

Ubiquitous computing is also known as pervasive computing or ambient intelligence, which aims at building a context-aware interactive environment based on multi-devices and multi-sensors, although their focal points differ in some aspects. Ubiquitous computing tends to describe working environments like the collaborative environment designed by Mark Weiser, which integrated smart boards, pads and tabs to construct a distributed communication and collaborative system in the laboratory. Pervasive computing was first used and supported by IBM in 1998, with the main idea that computing could be conducted everywhere and anywhere by networked digital devices. Ambient intelligence first emerged in 1999 ([Ronzani 2009]), and is actually based upon the theories of ubiquitous/pervasive computing. It combines research with Human-Computer Interaction (HCI), context-awareness (context-aware computing [Schilit & Theimer 1994]), *etc.*, to create an environment that is sensitive and responsive to the presence of people. Figure 1.2 describes the relationship among these different concepts in our view: we believe that Sensor Networks and the Internet of Things (IoT) are the basis. How to capture, organize, process and analyze the data generated from these connected sensors and devices is one of the major issues at stake. Z. Xiong ([Xiong 2012]) thinks that one of the smart city principles is based on the concept of "Data Vitalization". The idea behind data vitalization is to let data have life, and to combine separate data for better usage.

1.2 Location Based Services

1.2.1 Historical View

The form of Location Based Services (LBS) is not new to us in the 21st century, since we either actively use them or passively receive location based push notifications in our daily life. To find a nearby restaurant using one's smart phone, to choose an itinerary towards a parcel pick-up point, to receive welcome messages when traveling across different countries, to receive local weather forecast notifications: all these are just some of the examples of LBS that we are familiar with.

However, location services could date back to the 1970s, when the U.S. Department of Defense was operating the Global Positioning System (GPS), a space-based satellite navigation system serving the positioning of people and objects. Consequently, GPS was initially intended for military purposes, and it was not until the 1980s that the U.S. government allowed positioning data to be freely accessed by other industries worldwide. Both [Bellavista *et al.* 2008] and [Schiller & Voisard 2004] believe that the mobile network played an important role in the emergence of LBS in the late 1990s. Actually, the U.S. government passed a mandate called Enhanced 911 (E911) in 1996, which asked mobile network operators to be able to locate emergency callers with prescribed accuracy, so that operators could deliver a caller's location to Public Safety Answering Points. Cellular technol-

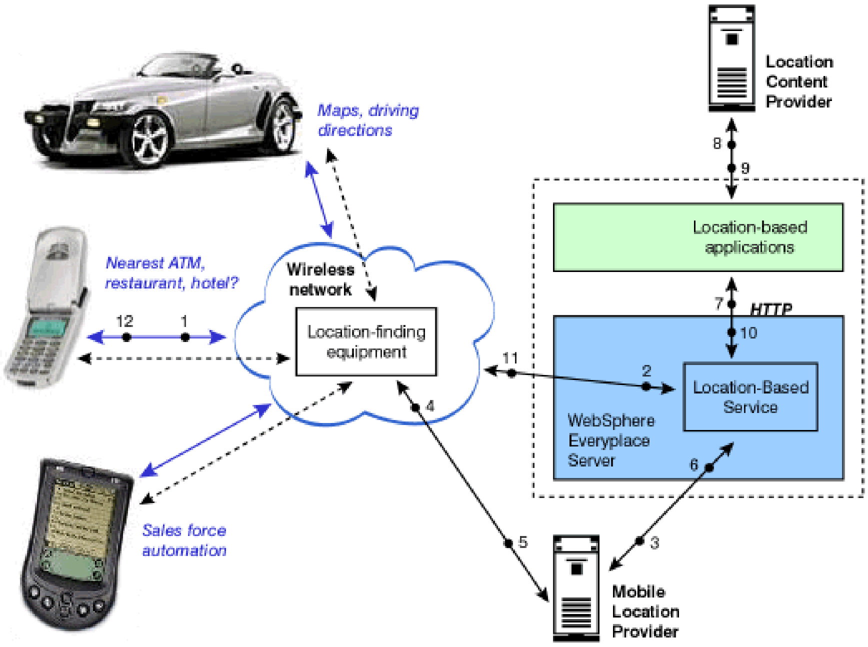


Figure 1.3: LBS applications[1]

ogy could not fulfil these accuracy demands, so operators started making enormous efforts to introduce advanced positioning methods. To gain returns on the E911 investments, operators launched a series of commercial LBS ([Bellavista *et al.* 2008]). These LBS offerings, in most cases, consisted of finder services that, on request, delivered to users a list of nearby points of interest, such as restaurants or gas stations.

In Europe, the European Union (EU) passed Article 26 of the "Directive of universal service and users' rights relating to electronic communications networks and services (2002/22/EC of 7th March 2002)", which asks member states to develop national regulations for mobile operators, enforcing the automatic positioning of emergency calls: "Member states shall ensure that undertakings which operator public telephone networks make a caller location information available to authorities handling emergencies, to the extent technically feasible, for all calls to the single European emergency call number 112." Since more than 80% of European op-

[1]Figure source:`http://www.watblog.com/2008/04/16/mobile-positioning-and-location-based-services-an-insight-into-the-future-of-mobile/` last checked available on June 19th, 2013

erators have implemented Cell-ID technology ([Concise-Insight-Ltd. 2003]), which could offer only a very low accuracy level, network operators tried to look for a commercial purpose for positioning information. This proved that even with Cell-ID, location information can successfully be integrated by operators into many existing and new applications that enhance current value propositions and usability ([Schiller & Voisard 2004]).

In 2005, several significant developments and favorable conditions came together to make LBS re-emerge (Figure 1.3). The emergence of GPS-aware mobile devices, the arrival of the Web 2.0 paradigm, and the introduction of 3G broadband telecommunication technologies all contributed to enabling developments.

1.2.2 What Are LBS

The definition of LBS varies with different perspectives. Some regard LBS as "any service or application that extends spatial information processing, or Geographical Information Systems (GIS) capabilities, to end users via the Internet and/or wireless network" ([Koeppel 2000]), while others view LBS as "geographically-oriented data and information services to users across mobile telecommunication networks" ([Batty *et al.* 2004]). Those are from the point of view of GIS, though the former puts forward the GIS abilities accessible in networked environments and the latter concentrates on geographic data and information-providing services via mobile-networked environments. While traditional internet GIS applications, like online map services, were important LBS applications, current LBS are supported more and more by lightweight mobile devices such as smart phones, personal digital assistants (PDA) and wearable computing devices (smart watches, wristbands, *etc.*) that can deliver more personalized services. Also, we are reminded that location is one type of context, so that the LBS are also related to location-aware computing and context-aware computing.

Location Based Services can be defined as services that integrate a mobile device's location or position with other information so as to provide added value to a user ([Schiller & Voisard 2004]).

So a true LBS application aims to provide personalised services to mobile users whose locations are in exchange ([Jiang & Yao 2006]). According to [Karimi 2004], who uses the word "telegeoinformatics" to refer to LBS, there are some distinct characteristics of LBS that separate them from traditional GIS applications. Actually, these characteristics are components such as models, software, hardware, data and people. Firstly, as we mentioned earlier, there have already been plenty of mobile devices that are location-aware, and each of them may have different hardware configurations and software platforms which also involve various telecommunication technologies. Secondly, the source data of LBS can come from remote sensing, traffic and transportation surveillance, topographic maps, which would be managed dynamically and simultaneously. In consequence, integrating and processing of data in real time seems more challenging. Models for methodology generalization, approach of data visualization and geo-processing in general would also call for further research challenges because the locations of users are changing all the time. Lastly, no LBS can ignore the end users who use the services: how to design the interfaces, how to visualize the results, how to adapt to different user needs, are all points that should be taken into consideration.

1.2.3 Classification of LBS

While researchers and analysts use different approaches to classify LBS applications, one main distinction could be "whether the service is person-oriented or device oriented".

- Person-oriented LBS refer to applications in which a service is user-based. Therefore, positioning of a person or using the position of a person to fulfil a service is the focus of the application. Usually, the person located can control the service, *e.g.* a friend finder application ([Schiller & Voisard 2004]).

- Device-oriented LBS often put users in a relatively passive position, which means that the person or object for positioning usually does not control the service (*e.g.* "Find My Device" for lost phone tracking). A device-oriented application may choose to locate a person, but is a free choice, not a necessary one. For instance, to locate a fleet where there is a group of people, the

Table 1.1: Categories and examples of LBS applications

	Push Services	Pull Services
Person-oriented		
Communication	Ex. 1: You get an alert from a friend zone application that a friend has just entered your area. Ex. 2: A message is pushed to you asking whether you allow a friend to locate you.	Ex. 1: You request from a friend finder application who is
Information	Ex. 3: You get an alert that a terror alarm has been issued by the city you are in.	Ex. 2: You look for the nearest cinema in your area and navigation instructions to get there.
Entertainment	Ex. 4: You have opted to participate in a location-based "shoot 'em up" game and are being attacked.	Ex. 3: You play a location-based game and look for another opt-in in your area to attack.
M-Commerce and Advertising	Ex. 5: A discount voucher is being sent to you from a restaurant in the area you are in.	Ex. 4: You look for cool events happening in the area you are in.
Device-oriented		
Tracking	Ex. 6: An alert is sent to you from an asset-tracking application that one of your shipments has just deviated from its foreseen route. Ex. 7: You get an alert that your child has left the playground.	Ex. 5: You request information on where your truck fleet currently is located in the country.

application can choose to locate only one or two people aboard.

Another classification method, which is rather traditional, is to distinguish between push and pull services.

- Push services mean that information is pushed to users without users having the intention to actively request it. This action should usually be performed with users' prior consent or registered preference (*e.g.* Google Now cards for nearby point of interest recommendation), and users should always be able to modify their subscriptions. But there are times when we cannot reject the push messages, *e.g.* welcome messages upon entering a new country.

- Pull services, on the other hand, imply that users intentionally use an application and actively send requests to "pull" information down from the network. Pull services are the most typical form of LBS.

[Spiekermann 2004] gives an overview of the LBS service dimensions with some application examples, as shown in Table 1.1.

1.3 Internet of Things

1.3.1 What Is IoT

The Internet of things (IoT) allows static and dynamic environmental objects to communicate and update real situations. The basic idea behind the Internet of Things (IoT) concept is the pervasive presence around us of a variety of things or objects - such as Radio-Frequency IDentification (RFID) tags, sensors, actuators, mobile phones, etc., which, through unique addressing schemes, are able to interact with each other and cooperate with their neighbors to reach common goals ([Giusto *et al.* 2010]).

At first glance, the concept of IoT recalls the idea of Ambient Intelligence (AmI) and Ubiquitous Computing (UC). To recall their definitions, the former refers to electronic environments that are sensitive and responsive to the presence of people. In an AmI world, devices work together to support people in carrying out their everyday life activities, tasks and rituals in an easy and natural way using information and intelligence that is hidden in the network connecting these devices; when the devices are smaller and more integrated into our environment, only the user interface remains perceivable by users ([Ambient Intelligence Lab]). The UC focuses on the omnipresence of processing devices, which are small, inexpensive, robust, networked, and distributed at all scales ([Weiser & Brown 2014]). The concept of IoT is closely linked to AmI and UC, but its central issues are to *make a full interoperability of interconnected devices possible, providing them with an ever higher degree of smartness by enabling their adaptation and autonomous behavior, while guaranteeing trust, privacy and security* ([Atzori *et al.* 2010]).

So the concept of IoT is not as simple as just connecting things to the internet, and it is true that, among the research community, there have been vivid discussions about its manifold definitions. As is rather natural, if we decompose the IoT term, we obtain “Internet” and “Things”. The former would throw back a network-oriented view of IoT, while the latter moves towards objects that would be integrated into an existing framework. Some even push further to propose a concept of so-called “Web of Things” ([Guinard & Trifa 2009], [Duquennoy *et al.* 2009]), focusing on reuse of existing Web standards to integrate everyday-life objects that contain an embedded device or computer ([Fielding & Taylor 2002], [Gellersen *et al.* 2009], [Kindberg *et al.* 2002], [Prehofer *et al.* 2007]). To bring Web-based smart things into peoples’ everyday life, and to enable humans to better interact with smart thing environments, more intuitive and explicit interaction mechanisms are required ([Mayer *et al.* 2014]).

1.3.2 The Emerging IoT

Although IoT has not been fully deployed, industrial, standardization and research bodies believe that it could have a huge impact on the behavior and social life of potential users. Possible application areas could be domestic and working fields such as domotics and healthcare, and could also include industrial manufacturing, intelligent transportation and business management. The US National Intelligence Council has placed the IoT in the list of six ”Disruptive Civil Technologies” that will have potential impacts on US national power ([National Intelligence Council 2008]). According to the NIC, ”by 2005, Internet nodes may reside in everyday things - food packages, furniture, paper documents, and more”, ”popular demand combined with technology advances could drive wide-spread diffusion of an internet of things that could, like the present Internet, contribute invaluably to economic development”.

As stated by the authors of ([Atzori *et al.* 2010]), advanced cars, trains, buses as well as bicycles along with roads and/or rails are becoming more instrumented with sensors, actuators, and processing power; roads themselves and transported goods are also equipped with tags and sensors that send important information to traffic control sites and transportation vehicles to better route traffic, and help manage depots. Several works have been published to describe IoT applications in the transportation and logistics field. In ([Gerla *et al.* 2014]), the authors talk about the evolution from intelligent grid to autonomous cars and vehicular clouds as internet of vehicles. This will be a distributed transportation fabric capable to make its own decisions about driving customers to their destinations and will have communication, storage, intelligence, and learning capabilities to anticipate customers' intentions. In the smart on-board transportation management system proposed by the authors of ([Tarapiah *et al.* 2014]), the implementation of the end-to-end system takes the form of a web application, taking advantage of RESTful architecture to meet functional requirements. A "Barcelona city smart vehicle use case" has been designed to demonstrate the value of COMPOSE, an open-source Platform as a Service (PaaS) for developing Smart City solutions based on connected objects ([Doukas & Antonelli 2014]). In this use case, parking sensor information and traffic data from the streets of Barcelona city are processed by the API web services to provide output about the available parking places in the designated area, including traffic conditions and time of arrival estimation for each parking spot.

1.3.3 Integrating Everything

From a technologies' point of view, to integrate "anything" into the Internet, the technologies of identification, sensing and communication are key elements (Figure 1.4) ([Luckenbach *et al.* 2005]). One of the most widely used is the RFID systems, consisting of readers and tags. The tags are attached to each object with its unique identifier and related information stored inside it. When receiving appropriate signals from the reader, a tag nearby can respond to the query to announce its presence and transmit the information contained. Thus, RFID systems could be used to supervise the status of objects in real-time, where objects do not necessarily have to be within sight. Due to the cheap cost and low energy consumption of RFID systems, their use has been widespread in industry, especially in the trading network. Auto-ID Labs ([Auto-Id Labs]), a leading global network of academic research laboratories in the field of networked RFIDs, has been making great efforts

[1]Figure source:`https://en.wikipedia.org/wiki/Internet_of_Things`, last checked available on March 3rd, 2016

Figure 1.4: Connecting everyday-life objects[1]

towards standardization with its successor EPCglobal ([The EPCglobal]).

Sensor networks are also crucial parts in the architecture of IoT ([Priyantha *et al.* 2008]). According to the author of ([Atzori *et al.* 2010]), sensor networks could cooperate with RFID systems to better track the status of objects, i.e., their location, temperature, movements, etc. Sensor networks are made up of a large number of nodes distributed in a certain way, which will report the sensing results to a small number of nodes (sink). A huge amount of research work has been carried out on sensor networks in recent decades, to tackle problems at all layers of the protocol stack ([Akyildiz *et al.* 2002]).

Besides ordinary objects, devices in which a certain degree of intelligence is embedded also need to be included in the network. Varying according to implementation, communication technologies in smart devices could be Zigbee, Bluetooth, low power WIFI and 6LoWPN ([Hui & Culler 2008a], [Hui & Culler 2008b], [IPSO alliance]), enabling their connections to the network.

With more and more devices and sensors linked to the network, we can imagine the quantity of data that they would generate – a huge amount! However, this depends on the scale of the implemented IoT application: for example, there is no need to have a fully-fledged backend platform to store and process data when we just intend to implement a small smart home IoT application supported by less than ten sensors, whilst cloud-based software-as-a-service infrastructure would be necessary to enable millions of daily sensor readings for years of a large-scale IoT product. Furthermore, analytics should be conducted on the data to extract their value from raw bits and bytes, since big data analytics tools have become generally available, as well as basic statistical tools and more advanced machine learning approaches (deep learning): statistical tools finding you the known knowns in the data; machine learning finding the known unknowns; whilst deep learning is able to find the unknown unknowns ([College 2016]).

1.4 Research Area

This work is in the context of our international France-China research project on Smart City in the laboratory ([David & Chalon 2010], [David *et al.* 2011], [David *et al.* 2012], [David *et al.* 2013]): Smart City is a real augmented environment allowing ubiquitous computing, with up-to-date web 2.0, which is collaborative, mobile and contextual. Human actors as well as different things (*i.e.* LBS and IoT) are now an integral part of internet. We proposed an approach called MOCOCO (Mobility, Contextualization, Collaboration) to conduct research work with multiple applications in working, learning and social situations; professional and home working situations, professional and teenager contextual mobile learning situations as well as Smart City applications are taken into account – transportation, goods distribution and local sporting and cultural activities.

This work focuses on LBS in Smart City. While previous work was conducted to study a physical hot spot based ([David & Chalon 2010]) communication / collaboration style LBS, we still have a strong temptation to carry out further critical study on what LBS would be like in a Smart City, and provide new proposals for overall solution design using IoT and implementation for another case study situation, especially in the transportation area.

One aspect that we have not yet talked about in regard to LBS, is privacy preservation, which is extremely important. Apparently, a location or locations the user has visited in the past are sensitive data and risk being misused – criminal intent or behavior analysis for commercial and advertisement purposes. [Bellavista *et al.* 2008] proposed dynamic trust management and user-controlled privacy policies to enforce privacy protection in the LBS. However, privacy issues are not concerned in the research area of this thesis and we shall not discuss them any further at present. The same goes for IoT.

1.4.1 Research Question and Research Challenges

The research question of this dissertation is how to adapt LBS to the requirements of Smart City, using IoT implementation to integrate a certain degree of ambient intelligence in dynamic situations and assuring the reliability, extensibility and reusability of the applications. More specifically, we would like to build an environment that allows location-based interaction, automatic location-based information propagation, and location-based behavior regulation, which are important aspects in our vision of user-centric LBS.

As we have already mentioned previously in the part on key characteristics of LBS, heterogeneity of devices, data management and analysis, and HCI aspects are classical main challenges for LBS.

The evolution of LBS also brings some new challenges, a short version of which could be, LBS become more complicated, involving a) from reactive to proactive, b) from self-referencing to cross-referencing, and c) from single-target to multi-target, d) from content-oriented to application-oriented ([Bellavista *et al.* 2008]). Reactive LBS are explicitly invoked by the user, while proactive LBS are automatically initiated when a predefined event occurs; user and target coincide in self-referencing LBS, while cross-referencing LBS make use of one target location for service-provisioning of another user; the major focus is on tracking one target's position in single-target LBS, while in multi-target LBS, the focus is rather on interrelating the positions of several targets; content-oriented means relevant content is delivered depending on users' locations (a list of points of interest, maps, or information about nearby touring sites), but application-oriented LBS can dynamically deliver impromptu applications tailored to users to provide a more powerful and richer interaction model. Consequently, to design a solution for the LBS problem, much more would need to be taken into consideration.

1.4.2 Research Method and Research Contributions

The research method that we used in this dissertation belongs to the "Design Science Research" methodology category ([Kuechler & Vaishnavi 2008]), which means the design of novel or innovative artifacts and the analysis of the use and/or performance of such artifacts to improve and understand the behavior of aspects of Information Systems. It was mainly involved in the case study stage, as we built an artifact combined with use study as formal modeling of the artifact, in the solution design of an intelligent transportation system, which is also a type of information system.

The main contribution of the work in this thesis is the proposition and validation of using IoT to enhance LBS in a Smart City. More specifically, the contributions are as follows:

- Examine at component level the possibility and feasibility of using IoT to better support LBS in the context of Smart City. This is carried out by conducting bibliographic studies on the three main concepts as well as primary significant achievements and projecting our thoughts onto them.

- Propose a solution design to a transportation problem – dynamic management of road lanes, including architecture, simulation environment and user interfaces ([Wang *et al.* 2014a], [Wang *et al.* 2014b]). This is a case study of our approach with regard to combining LBS with IoT. What should be mentioned here is that the simulation environment is critical: it not only serves as a tool to validate the overall solution design, but also provides a strategy to design the auxiliary tools when dealing with simulation-based projects.

- Make a proof of concept in addition to the simulation environment to validate the technological aspects of the dynamic lane management system, and to provide an example of IoT enhancing LBS. Although the proof of concept is simple, collection of data, integration of objects, transmission of information, and dynamic allocation are all involved and well supported ([Wang *et al.* 2015]).

- Conduct user tests using an evaluation tool based on visualization of the simulator results ([Wang *et al.* 2016]). This dynamic evaluation tool takes into consideration the speed of the vehicle as well as the surrounding circumstances, and allows for real-time user interaction during the tests. This is an important step for the HCI aspects and is also valuable for visualization of data for both end users and developers.

1.5 Organization of the Dissertation

The remainder of this dissertation is structured into the following sections:

- In chapter 2, we survey related works and major references relevant to our topic of research. We also present our reviews and put forward our approach for combining IoT and LBS.

- In chapter 3, we describe the ICT-based solution design to the dynamic lane management problem and we also give an overview of the simulation environment.

- In chapter 4, we introduce the central part of the simulation environment – simulator of core system, including a detailed discussion of model design. We then illustrate all the algorithms related to implementation of the simulator. The proof of concept as a technology validation is also introduced.

- In chapter 5, we first show different forms of visualization and then use a visualization tool to conduct user tests, and explain the analysis results.

- In chapter 6, we conclude the work of the dissertation and provide our perspectives with regard to future work in Smart City.

Chapter 2

State of the Art

In this chapter, we survey related works and primary significant achievements relevant to our topic of research. We begin with a LBS application analysis, then we present how the IoT architecture could enhance LBS, and we also talk about traffic simulation and issues of dynamic management of road lanes.

2.1 Analysis of a LBS – Friend Finder App

We first take a look at an example of LBS, which is a mobile friend finder application developed by AT&T Wireless ([Spiekermann 2004]). The goal was to provide enhanced LBS solutions for people to keep in touch with their family and friends, to be able to find one another, and to get directions to local shops and restaurants. At that time, Kivera just came out with a new location-aware GPRS (General Packet Radio Service) service called mMode (essentially a WAP (Wireless Application Protocol) variant), so the two companies collaborated to create the first-ever Friend Finder application built upon Kivera's Location Engine. In 2001, when most LBS were straight forward simple GIS applications, AT&T's Friend Finder application was indeed a forward-thinking and visionary step in making the first wireless LBS application accessible to the general public.

2.1.1 Features and Usage

The friend finder application provides the following features:

- Deliver relevant user information about the location of a friend or family member's mobile cell phone position.

- Calculate driving directions from a mobile cell phone position to an address or a point of interest

- Provide for selection of a business point of interest meeting place between two mobile cell phone positions.

- Provide for selection of a business point of interest in the proximity of a mobile phone position.

The user is offered a browser-like interface to access all the functionalities, the sequence could be as follows:

- First, a user, *i.e.* Tom, adds selected friends and family members to his friend list. This is done simply by typing the friend's (*i.e.* Alice) AT&T cell phone number from an appropriate screen window.

- Alice, Tom's friend, then receives a text notification on her mobile, asking whether Tom can add her to his friend finder list, which implies that later on Tom is authorized to track her location.

- If Alice replies affirmatively to Tom's friend adding request, she is added to Tom's friend list.

- Tom can now choose Alice from his friend list and select "find friend" in a pop-up menu to locate her. The location of Alice's phone (street address or closest intersection) is transmitted to Tom's phone. At the same time, Alice is also notified by a message that her newly added friend Tom has queried her location with this application. Alice can always choose to be invisible to Tom if she does not want to reveal her location to him, or feels like blocking this contact.

- Once Tom has received Alice's location, he can get directions to her location and find restaurants and cinemas (or other points of interest) near her, or some place in between, so that they can meet.

2.1.2 Data Capture and Collection

No LBS can live without a data source and nor can the friend finder application. Actually, it mainly uses information from the road network database, business and landmark information, and dynamic data such as traffic state and weather reports.

The road network database usually contains digital formats of geographic content and map data. These data can be captured in various ways, such as satellite remote sensing imagery, scanned maps, *etc.*. Some vendors even physically drive each road segment in GPS-equipped cars, registering every change of direction and photographing road signs to keep track of specific road conditions such as speed and height / weight restrictions. Map data are stored in a vector format made up of line segments, which represent the roads and link points representing intersections or other road features. Each link has start and end points as well as possible shape points for curved roads. Attributes are also added to data, such as forbidden turns, tunnels, bridges *etc.*.

So these road network data support the friend finder application's core LBS functionality: geo-coding, phone location display and driving directions. The latitude / longitude coordinates of the user's cell phone are resolved in the application to obtain the address. The coordinates may come either from the mobile operator who looks up the cell tower location, or directly from the embedded GPS chip in the cell phone, or even both.

The point of interest information forms another database, which links the business and landmark information to the road network database, so that the digital representation of the road network becomes more detailed and rich. Map data vendors can enrich their database by attaching point of interest information, while business vendors can also contribute their geo-coded piece of information to the LBS provider's data set, *i.e.* retail stores integrating their digitized location to allow customers to search for the nearest stores.

So point of interest data together with map data can provide users with a selection of meeting places between two mobile phone positions. A user can of course select a business point of interest in his/her own proximity to his/her phone's location.

Last but not least, we have dynamic data. Users who travel daily from home to work by car are really interested in the optimal route during morning and evening peak hours. Traffic situations and weather conditions can largely affect a road's supported traffic speed. However, these data cannot be coded into a map data set a priori. So, in most cases, LBS providers look for reliable dynamic data sources or use directly other dynamic data based services to carry out a service composition in their LBS platforms.

In the case of the friend finder application, it seems that dynamic data are not as important as others, since they are only useful when suggesting driving directions while a well designed navigating system can do a better job.

2.1.3 The LBS Platform and Middleware

Kivera's LBS platform, on which the friend finder application is based, is shown in Figure 2.1. The map data set, point of interest database, geo-coding component, and routing software form the packages that the application developer can use to build custom LBS applications. Some LBS vendors choose to market LBS directly to consumers and to provide stand alone packages, such as store finders, to businesses. These applications are favored by companies that do not need highly customized solutions.

Actually, the Kivera's LBS platform could already be considered a middleware – the part between "Location Infrastructure" and "Application Platform". If we look at the relationship from Kivera's point of view, the application layer is in fact the "client" group in the LBS industry, and the friend finder application is not the only member. Since position determination equipments are set deeply in the network operators, and the number of LBS applications is increasing, a middleware is implemented that comprises all of those services requesting location data to integrate it into the offering. Later on, if a new type of individual data service is added, it should not require complex and lengthy hookup. In other words, the middleware layer can significantly reduce the complexity of service and application integration (Figure 2.2). From the point of view of the developers of the friend finder appli-

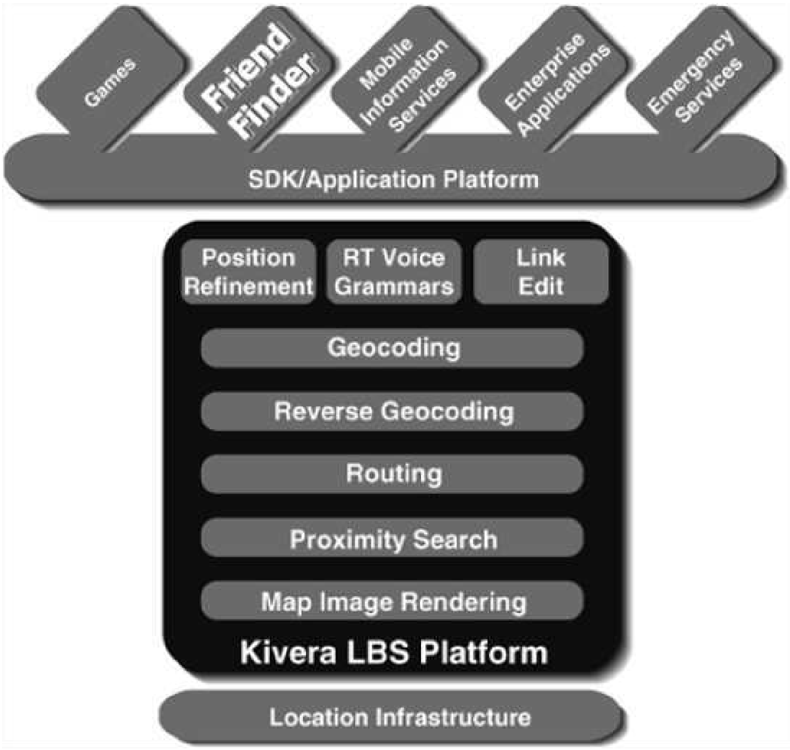


Figure 2.1: Kivera's LBS platform ([Spiekermann 2004])

cation, if one day they find that Kivera's LBS platform can no longer meet their needs as they plan to add more sophisticated features, they can set up their own LBS middleware to better support the application.

According to [Jacobsen 2004], a middleware is defined as a "set of services that facilitate the development and deployment of distributed applications in heterogeneous environments", and a middleware system should abstract the details of the underlying operating system, network substrate and protocols, mask possible failures, and even mask the distribution of interacting systems; in the context of LBS, it refers to the services, abstractions, and models that implement mobile user coordination, information correlation, and information dissemination. More precisely, some of the requirements of LBS middleware ([Cugola *et al.* 2002], [Jacobsen 2001]) are as follows:

- Management of the mobility-awareness inherent to all LBS applications, to support disconnected operations.

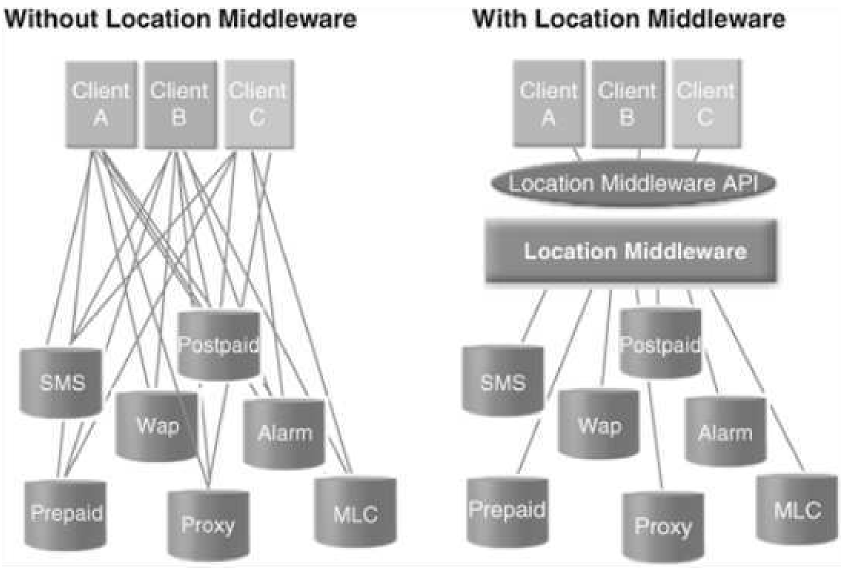


Figure 2.2: Application Integration with / without middleware ([Spiekermann 2004])

- Management of changes in the underlying network topology that may occur, such as ad-hoc location-based services.
- Management of large numbers of data sources, information providers.
- Management of large amounts of content sent to the system for filtering, matching and correlating, propagation of information to thousands of consumers simultaneously.
- Management of variation of users' interests (*e.g.* update, insertion, removal).
- Processing of various content formats such as attribute-value pairs (HTML), XML marked-up data, easy configuration for support of other evolving data formats.
- Support for different notification channels and protocols such as WAP, UDP, TCP, SMTP, SOAP, *etc.*
- Support for approximate subscriptions and approximate events to enhance system flexibility by increasing the expressiveness of the filtering language and the publication language.

- Support for high availability despite node failures (*e.g.* guarantee notification delivery success rate).

2.1.4 Our View

The AT&T's friend finder application as well as its functionalities are no longer cutting edge stuff in our days thanks to the rapid development of information and communication technologies. However, the reason why we cite it here, is that it is really a typical example which shows the limitations in the evolution of LBS, and thus provides us with reference for improvement to adapt to the requirements of Smart Cities. As we mentioned earlier, to integrate a certain degree of intelligence in LBS, and to provide more personalized or customized LBS, we have to make better use of data. Nevertheless, the road network data together with point of interest data have already been leveraged by developers for several years now. Consequently, we need to divert our attention to dynamic data (section 2.1.2).

One possible direction is "Location as a Context ([da Rocha 2004])". According to [Abowd *et al.* 1999], context is defined as "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves". So location is exactly part of context, but context contains far more things than location. Context constitutes an important part of LBS, as both the user and location belong to context ([Jiang & Yao 2006]). In LBS applications, context modeling could capture physical environment elements (absolute or relative location, infrastructure, conditions, *etc.*) and human factors (user, task, social environment). Mobile computing middleware makes use of context information to support adaptability, such as providing proximity-based selection (context-aware middleware for ambient intelligence [Xu 2013]), and to fire actions when there is a context change (proxemic interaction with large public screen [Jin 2014]).

The work flow of the user-centric context-aware middleware in ambient intelligence proposed by [Xu 2013] is described in Figure 2.3: the application informs the versatile context interpreter which context model to use and asks to receive contextual evolution (1&2); the enterprise service bus collects the data from different sensors and propagates them to the context aggregator, before transferring the data to the context knowledge base (3&4); the query engine is then notified of arrived data and it also calls on the context inference engine to apply context inference (5&6); after the inference operation, data are re-collected by the query engine (7&8), before being sent to the application (9); the application can also decide to update actuator states (10).

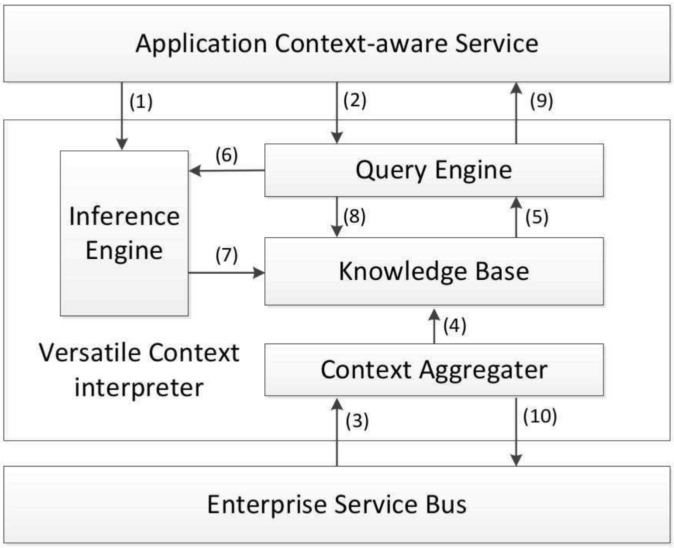


Figure 2.3: Context-aware middleware behavior workflow [Xu 2013]

As for the location-based proxemic interaction with a public display ([Jin 2014]), as shown in Figure 2.4, the space in front of a public display can be classified into several zones according to the user's location, or the distance from the display. In Figure 2.4(a), there are two discrete areas, the information displayed and the interaction type could be different in each area; besides, users in closer areas cannot be disturbed by users in outer areas. In Figure 2.4(b), there is only one continuous interactive area. Although there are no border lines, the saturation of colors indicates the interaction types: from far to close, users can get increasingly more information and gradually engage in increasingly sophisticated interactions or take increasing control of the display screen.

Thus context information is a rich resource for dynamic data, especially when

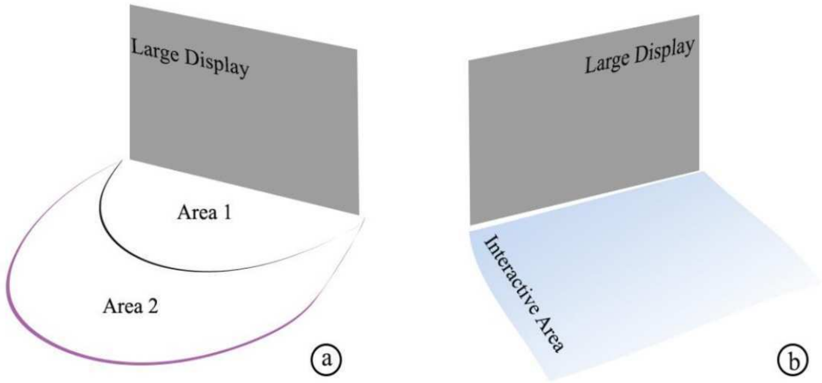


Figure 2.4: Proxemic interactive zones for ambient intelligence [Jin 2014]

there are no appropriate data providers. LBS, in this respect, can be seen as a kind of context-aware service, so that all the context-aware middleware can contribute to LBS design or LBS middleware deployment.

Another direction is counting on IoT. As a matter of fact, the devices that generate location data, or the infrastructure itself, could all be part of the IoT in a Smart City. So why not use the whole IoT implementation to enhance LBS, from data source to middleware, from service integration to data visualization. Another fact is that most of first-hand context information is captured via sensor technology ([Schmidt *et al.* 1999]), and sensor networks are crucial parts in IoT architecture (section 1.3.3). Potentially, the IoT could also enhance context-aware computing.

2.2 IoT Architecture

Instead of going through all the details of IoT architecture, we here concentrate on some important elements that are related or helpful to LBS.

2.2.1 Device Integration – REST Architecture

It is easy to say that objects are connected via wireless communication technologies and then they are ready to serve our applications, but one of the problems that concern us is "how" – how to assure a consistent access to the connected objects and their functionalities. [Guinard 2011] proposed in his work a complete Web of Things (one vision of IoT that concentrates on reusing existing Web standards) architecture, in which a device accessibility layer is placed and REST (REpresentational State Transfer) architecture is used to solve the problem. The REST tries to support "scalability of component interactions, generality of interfaces, the independent deployment of components as well as intermediary components to reduce interaction latency, enforce security, and encapsulate legacy systems" ([Fielding 2000]).

The core idea of REST regards the notion of resource as "any component of an application that needs to be used or addressed" ([Guinard 2011]). Resources could be real physical objects, abstract concepts (collections of objects), as well as transient and dynamic concepts (server-side state of transactions). The principles of REST are as follows ([Fielding 2000]):

C1 *Resource Identification:* the Web relies on Uniform Resource Identifiers (URI) to identify resources, thus links to resources (C4) can be established using a well-known identification scheme.

C2 *Uniform Interface:* resources should be available through a uniform interface with well-defined interaction semantics, like the Hypertext Transfer Protocol (HTTP). HTTP has a very small set of methods with different semantics (safe, idempotent, and others), which allows interactions to be effectively optimized. It also allows for a clean decoupling of the interface (RESTful interface) and the actual service implementation. The uniform interface has 4 main methods: GET, PUT, DELETE and POST, with good coverage of CRUD (Create Read Update Delete) types of applications. So they are also supported to explicit every action a client can execute on a resource, whatever the type of application.

C3 *Self-Describing Messages:* agreed-upon resource representation formats make it much easier for a decentralized system of clients and servers to interact without the need for individual negotiations. On the Web, media type support in HTTP and the Hypertext Markup Language (HTML) allow peers to cooperate without individual agreements. For machine-oriented services, media types such as the Extensible Markup Language (XML) and JavaScript Object Notation (JSON) have gained widespread support across services and client platforms. JSON is a lightweight alternative to XML that is widely used in Web 2.0 applications and directly parsable to JavaScript objects.

C4 *Hypermedia Driving Application State (Connectedness):* clients of RESTful services are supposed to follow links they find in resources to interact with services. This allows clients to "explore" a service without the need for dedicated discovery formats, as well as allowing clients to use standardized identifiers (C1) and a well-defined media type discovery process (C3) for their exploration of services. This constraint must be backed by resource representations (C3) with well-defined ways in which they expose links that can be followed.

C5 *Stateless Interactions:* this requires requests from clients to be self-contained, in the sense that all information to serve the request must be part of the request. HTTP implements this constraint because it has no concept beyond the request / response interaction pattern; there is no concept of HTTP sessions or transactions. It seems important to point out that there might very well be states involved in an interaction, either in the form of state information embedded in the request (HTTP cookies), or in the form of a server-side state that is linked from within the request content (C3). Even though these two patterns introduce the state into the service, the interaction itself is completely self-contained (in other words, does not depend on the context for interpretation) and thus is stateless.

Bearing in mind the REST architecture, we can integrate all the location-aware devices and objects that can provide useful data and services in the framework, by following the steps proposed by [Guinard 2011]:

1. Resource Design: identify the functionality or services of an object, organize the hierarchy of these services and link them together, fulfilling constraints C1 and C4.

2. Representation Design: decide which representations will be served for each service, fulfilling constraint C3.

3. Interface Design: decide on the actions allowed for each service, fulfilling constraints C2 and C5.

4. Implementation Strategy: choose a strategy to integrate the objects in the Internet and the Web, either directly or through a Smart Gateway.

2.2.2 Middleware Consideration

Now that we are quite clear with the requirements of LBS middleware (2.1.3), it is also important to check how well IoT middleware can work with, or at least not interfere with, LBS middleware.

Due to the fact that objects in the IoT are often abstracted to resources or services, IoT middleware usually follows the Resource Oriented Architecture (ROA) or Service Oriented Architecture (SOA) style. The concept of ROA was proposed by [Richardson & Ruby 2008] and its design goal is to assure that, in a decentralized and large-scale service architecture, millions to billions of available resources and loosely coupled clients, with potentially millions of concurrent interactions with one service provider, are all well aligned ([Guinard 2011]). On the other hand, the principles of SOA, adopted by [De Deugd *et al.* 2006] and [Pasley 2005], allow for decomposing complex and monolithic systems into applications consisting of an ecosystem of simpler and well-defined components, and the development of business processes enabled by the SOA is the result of the process of designing work flows of coordinated services ([Atzori *et al.* 2010]).

In the LBS context, we really could not tell which one is better than the other, between the SOA and the ROA, as both could be integrated into the LBS middleware. However, in our point of view, what really counts is the layer just above the objects – the Object Abstraction layer for the SOA and the Accessibility layer for the ROA (Figure 2.5). Then, we could either perform the mash-up or composition of resources to form some useful APIs for the LBS middleware, or carry out the service management directly in the IoT architecture and provide a service package for LBS middleware.

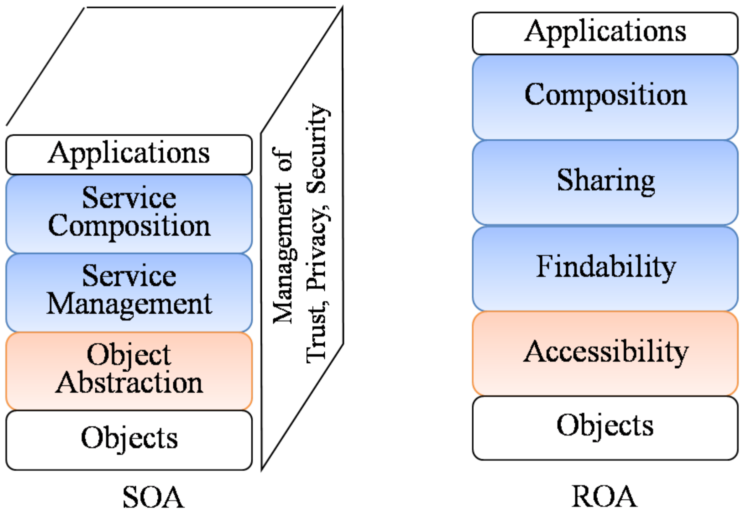


Figure 2.5: SOA and ROA for IoT middleware

2.2.3 Other Work on IoT Infrastructure

COMPOSE and Fastprk

COMPOSE is an FP7 EU supported project targeting the achievement of an open-source infrastructure and a set of tools and methods for developing smart applications that can communicate with smart objects such as smartphones, sensors, actuators, *etc.*, and external information resources ([Doukas & Antonelli 2014]). Its technical approach is to “integrate the IoT and the IoC with the IoS through an open marketplace, in which data from Internet-connected objects can be easily published, shared, and integrated into services and applications” ([Compose-Project 2012]). In the mean time, the marketplace would provide a coherent and robust framework covering both delivery and management aspects of objects, services, and their integration. The key features are as follows ([Compose-Project 2012]):

- Object virtualization: enabling the creation of standardized service objects
- Interaction virtualization: abstract heterogeneity while offering several interaction paradigms
- Knowledge aggregation: creating information from data
- Discovery and advertisement: of semantically-enriched objects and services

- Data Management: handle massive amounts and diversity of data / metadata
- Ad hoc creation, composition, and maintenance: of service objects and services
- Security, heterogeneity, scalability, and resiliency: incorporated throughout the layers

Although we have not examined the technical details of the platform, we are sure that it handles most of the primary elements of IoT architecture, and of course its goal is far more than to provide LBS. However, we discovered that the Smart City test bed used in [Doukas & Antonelli 2014] for developing the Smart Vehicle Sharing use case of the COMPOSE platform, is also used to implement a Smart Parking System – Fastprk ([Fastprk 2012]). In the Smart Vehicle Sharing use case, real-time traffic from the streets of Barcelona city and parking sensors as well as the smartphones of users are virtualized as collections of services, and developers can query COMPOSE for traffic, parking, or user-location and have direct access to each individual service. On the other hand, the Fastprk system, with embedded sensors in each parking bay in the area and delivering instantly occupancy information not only via illuminated panels but also via mobile applications, can provide personalized guidance for drivers, helping them find the nearest vacant parking spaces. Moreover, the system can also be correlated to payment information. So the COMPOSE platform proved to be capable of implementing a type of LBS.

Avatar Architecture

[Mrissa *et al.* 2015] from the LIRIS laboratory proposed an Avatar Architecture for the Web of Things as shown in Figure 2.6. They focused on not only integrating objects into the Web, but also on bestowing a clever behavior on them so that they become smart objects.

An avatar is an extensible runtime environment endowed with an autonomous behavior, so it has a life cycle and main tasks ([Mrissa *et al.* 2015]). After being instantiated from the avatar builder, an avatar will be attached to the object that it connects to and, thanks to the reasoning engine, will decide to deploy a set of components according to the object's sensors and actuators as functionalities (Web

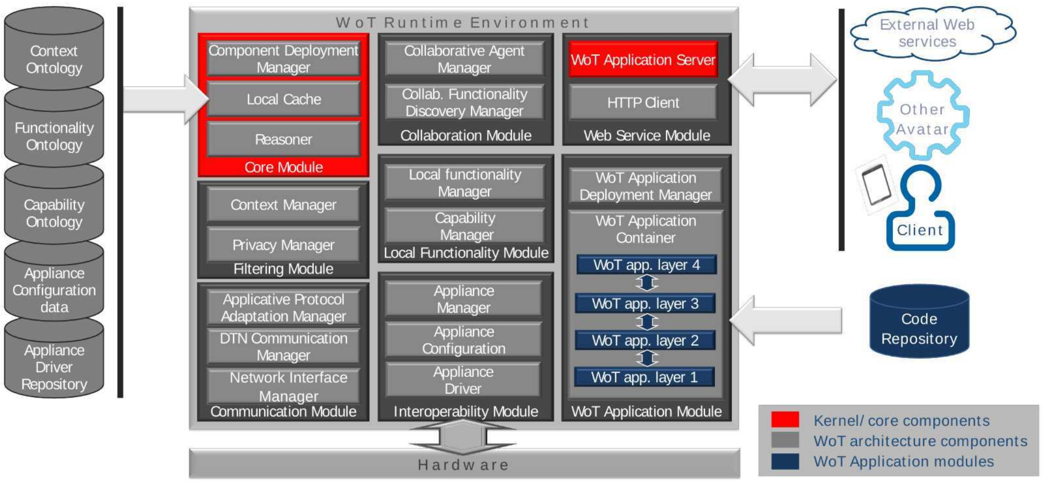


Figure 2.6: Web architecture of avatar agents([Mrissa *et al.* 2015])

services) exposed to the infrastructure or other avatars. Throughout its life cycle, the avatar will work autonomously, providing services and exchanging data with other avatars or end users. When it is time for the avatar's life to end, perhaps due to the disconnection of the object, the avatar will be destroyed and free the memory occupied, while some information will be saved to a local cache to speed up infrastructure performance if a similar avatar returns.

2.3 Transportation Issues and Traffic Simulation

2.3.1 IoT in the Transportation and Logistics Area

As mentioned in section 1.3.2, much work has been conducted to apply the IoT to the transportation and logistics area.

Assisted driving

With increasingly more sensors and actuators implemented on roads and rails, and with in-vehicle devices and interfaces becoming more advanced, a car's driver or passengers of a bus could be provided with more personalized information and services. Collision avoidance systems and monitoring of transportation of hazardous materials are two typical example functions ([Atzori *et al.* 2010]). Governmental regulators can take advice from more accurate real-time road traffic pattern information for planning and adjustment purposes. Also, personal vehicle drivers could be assisted by better routing suggestions from more intelligent navigation systems.

Mobile ticketing

A bus station could be equipped with NFC tags, visual markers or other numeric identifiers to allow passengers to get information about a certain bus line (stations, costs, timetables, available seats, *etc.*), or directly buy a ticket, via the mobile application of smart-phones ([Broll *et al.* 2009]). In the meantime, public transportation companies, with access to passengers' destinations and special requests, can organize dynamically rapid or snail transportation ([David & Chalon 2010]).

Logistics

Thanks to RFID and NFC tags, each link in the supply chain could be monitored such as raw material purchasing, storage, distribution, transportation conditions, return and after-sales service. So for companies, processing time needs from customer requirements to product supply could be considerably shortened. In addition, real-time access to the ERP (Enterprise Resource Planning) program helps shop assistants to better inform customers about availability of products ([Atzori *et al.* 2010], [Karpischek *et al.* 2009]).

Connected vehicles

The authors from Lyon University proposed in their research work a way to improve traffic flow by introducing cooperation between vehicles through their embedded communication and perception capabilities ([Guériau *et al.* 2015]). Road Side Units (RSU) are placed along the road to collect information for the infrastructure. Therefore, vehicles can not only communicate with each other (V2V communication) but also exchange information with the infrastructure part of the Cooperative Systems (V2I & I2V), as shown in Figure 2.7. Cooperation between vehicles takes place by allowing a vehicle to sense its surrounding environment and also broadcast all its sensor data to other vehicles. Then, the self-decision layer of the vehicle will decide whether and, if affirmative, how to cooperate with other vehicles ([Monteil *et al.* 2014]). Their work is based on a multi-agent approach.

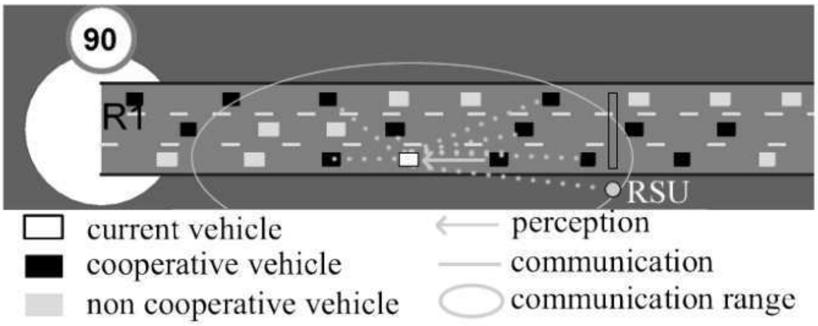


Figure 2.7: Cooperative traffic view ([Guériau *et al.* 2015])

2.3.2 Traffic Simulation

Intelligent Transportation Systems (ITS) are essential to the urban functioning and prosperity of modern, industrialized societies, as well as to Smart Cities, to assure citizens' qualitative mobility. The motor vehicle is the primary and widely used means to meet our traffic demands. However, road capacity is limited, with more and more motor vehicles traveling on the road, and traffic congestion is a severe problem in many cities. In addition, imploring traffic authorities to widen lanes or to build more roads in the central area of a city is neither a thoughtful solution nor an always feasible choice, not to mention the related environmental impact and energy consumption. That is why we need ITS to make utilization of road networks more efficient and to improve operations.

Modeling and simulation are often employed as a means of trying to analyze, understand and control complex systems, including transportation systems. Indeed, application of modeling and simulation in the field of transportation dates back to the 1950s, and simulation in this sense has been used very successfully as a tool for planning and building transportation infrastructure ([Aydt *et al.* 2012]). Now that computing power has increased incredibly in recent decades, realistic traffic simulation has proved its value not only in generating surrounding traffic in a virtual reality driving simulator but also in large-scale road network simulations for prediction of traffic conditions and travel times ([Kesting *et al.* 2008]).

The mathematical description of traffic flow dynamics plays an important role in traffic simulations. According to [Kesting *et al.* 2008], there are, in general, two major approaches for describing the spatio-temporal propagation of traffic flows. Macroscopic traffic flow models utilize the snapshots of traffic flow as a physical flow of a fluid. Traffic dynamics are characterized in terms of aggregated macroscopic quantities such as traffic density, traffic flow, average velocity, *etc.*as a function of space and time corresponding to partial differential equations ([Kesting *et al.* 2008]). The underlying assumption of all macroscopic models is the conservation of vehicles (expressed by he continuity condition), which was initially proposed by the authors of [Lighthill & Whitham 1955, Richards 1956]. Claimed to be more advanced, "second order" models additionally use macroscopic velocity as a dynamic variable in order to also consider the finite acceleration capability of vehicles ([Kerner & Konhäuser 1994, Treiber *et al.* 1999]).

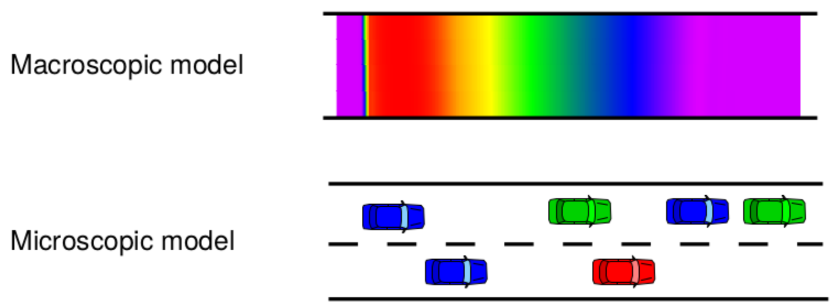


Figure 2.8: Different traffic modeling approaches: macroscopic vs microscopic ([Kesting *et al.* 2008])

On the other hand, microscopic traffic models apply to every individual vehicle and describe its motion (Figure 2.8). Actions such as acceleration, deceleration and lane changes of each driver are modeled as a response to the surrounding traffic. Microscopic traffic models are especially suited to the study of heterogeneous traffic streams consisting of different and individual types of driver-vehicle units ([Kesting *et al.* 2008]). This has largely powered agent-based vehicular traffic simulation ([Chen & Cheng 2010]) because actually the smartness of ITS stems from the fact that there are agents in the control system that accept input from sensors, make sense of the information they collect, and then have some forms of actuation on the environment. Also, a multi-agent vehicular traffic system functions as follows: agents operate in a shared environment provided by the road infrastructure and react to the neighboring vehicles – the job that microscopic traffic models can handle. Subclasses of microscopic traffic models could be time-continuous, cellular automata, or the hybrid form of the two former ones ([Kesting *et al.* 2008]). Some available microscopic traffic simulation software are SUMO ([Simulation of Urban MObility 1998]), FreeSim ([Miller & Horowitz 2007]), and Treiber's web-based simulator ([M. Treiber 2007]).

The authors of [Aydt *et al.* 2012] view things from another point of view. Although traffic simulation has been successfully used as a tool to evaluate hypothetical scenarios for their impact on traffic, existing tools and methodologies based on it are not good at capturing real-time data and thus are not suitable for providing real-time decision-making. Even if many vehicles are now equipped with smart on-board navigation systems, which are able to incorporate traffic data providing real-time information about congestions and sophisticated routing services to users, they are not smart in the sense that they can forecast how the traffic situation will develop or anticipate how widespread use of navigation systems can cause problems of its own. For instance, navigation systems can incorporate real-time traffic information and divert drivers in case of jams and slow traffic; however, if there are too many of these systems in use, this may cause a significant number of vehicles to be diverted through the same alternative route and thus effectively divert the congestion itself ([Aydt *et al.* 2012]). A real smart routing system will perform global optimization, which keeps goals regarding overall throughput and congestion minimization in mind while finding an individual's best route from one location to another. So a form of symbiotic simulation, Living Simulation, was proposed by them ([Aydt *et al.* 2012]) to merge ideas from collaborative, social computation with advanced individual-based modeling techniques to provide on-the-fly information support systems to individuals.

2.3.3 Dynamic Management of Road Lanes

Our study concerns dynamic management of road traffic, which is regularly increasing both in towns and outside built-up urban areas. The first approach designed to allow increase in traffic leads to solutions such as increasing the number of lanes, while the second approach aims at segmenting traffic according to categories (private vehicles, heavy vehicles, public transportation, priority vehicles) by proposing specific development and traffic rules, with, in particular, the creation of specialized lanes (bus, tram, trolley). This second choice can lead to satisfactory solutions provided that there is sufficient space.

When space is lacking and the frequency of this type of specialized traffic is not sufficient, there is a sense of waste and poor management. A third solution then emerges, namely dynamic allocation of lanes to different types of transportation. A significant study consisting of data gathering, analysis and classification was carried by J. Nouvier from the CERTU (Center of studies on networks, transportation, urbanism and public construction) [Nouvier 2007], as shown in Figure 2.9.

J. Nouvier [Nouvier 2007] collected and presented a large number of varied solutions, from the more physical (ad hoc movement of low walls with trucks, as shown in Figure 2.9 (a)&(b)) to the more informational (signposts with variable displays, as shown in Figure 2.9 (d)&(e)), enabling less or greater speed of dynamicity.

For dynamic bus lane allocation, it is important to mention the Lisbon experiment [Viegas & Lu 2004, Viegas *et al.* 2007]. The authors believe that road space use is inefficient (bus lanes) when bus frequency is low (less than 20/h), but that a few buses suffering in congested road sections is not acceptable either. So they put forward the idea that one lane can be reserved for the bus just enough time to allow it to move separately from general traffic. That is the goal of the Intermittent Bus Lane (IBL) system: a lane that can change its status from a regular lane (accessible for all vehicles) to a bus lane, for the time strictly necessary for a bus to pass; the status of the IBL is propagated to drivers using a system of both vertical and horizontal variable signaling: vertical variable message signs, and a linear set of LEDs installed on the road infrastructure. During the six-month demonstration which began in September 2005, they studied drivers' level of acceptance and compliance (claimed as good). The impact of the IBL on bus movement and general traffic flow conditions was quantitatively evaluated: 20 to 25% increase in bus average speed in all routes that use this road link, with peak hour gains in the order of 50%; no significant impact in general traffic main attributes (flow, vehicle speed),

(a) Heavy solutions

(b) Golden bridge SF manual dynamicity

(c) Easier solutions

(d) ICT based solutions with horizontal and vertical signs

(e) Emergency / Stop lane use and driving orientation reversibility

Figure 2.9: Set of existing solutions [Nouvier 2007]

as the controlling factor was a traffic signal downstream. We were also inspired by the work [Carey *et al.* 2009, Guler & Menendez 2013].

2.4 Summary

In this chapter, we examined at component level the possibility and feasibility of using the IoT to better support LBS in the context of Smart City. We highlighted the REST architecture as an important method for integrating devices, and we also surveyed other work on IoT infrastructure. After having discussed the applications of IoT in the transportation and logistics area and the traffic simulation topic, as well as the dynamic management of road lanes issue, we are ready to present our solution design in the following chapters, demonstrating our approach for combining LBS and IoT.

Chapter 3

ICT-based Dynamic Lane Management System

In order to better demonstrate our approach, we choose a particular problem related to traffic management in a confined circulation infrastructure, as a case study. In this chapter, we will introduce the system design and architecture of our solution, and the user interfaces as well as the simulation environment ([Wang *et al.* 2014a], [Wang *et al.* 2014b]).

3.1 Our ICT-based Dynamic Road Lane Management System

3.1.1 A System-Perspective Description

It is an established fact today that telematics or embedded and/or mobile ICTs can provide solutions leading to very high dynamicity (clearing a lane for a bus or an ambulance in real-time), provided that users are sufficiently informed and that regulations are complied with in terms of transportation (or suggestions to modify it), in particular, of user safety. Hereafter, we give a brief description of the ICT vision, in a system-perspective.

"Dynamic circulation lane allocation" aims at providing a system designed to share circulation lanes dynamically between public transportation (buses) or rescue services (fire-fighters and ambulances) and personal vehicle transportation, in order to share traffic lanes appropriately in the context of low frequency of specialized traffic or lack of space (impossibility or inadequacy of static allocation of circulation

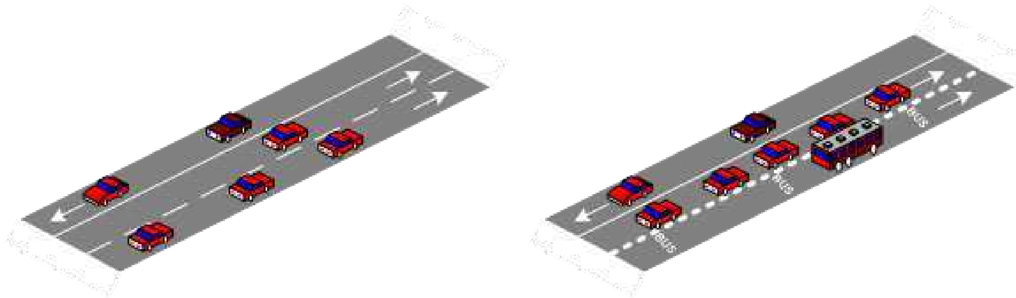


Figure 3.1: Dynamic circulation lane allocation example

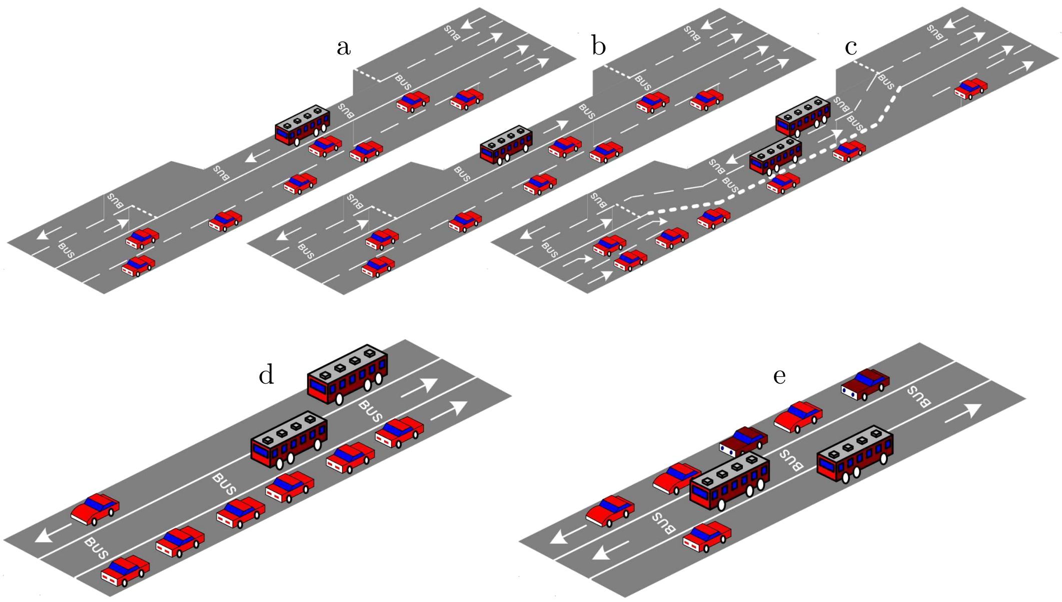


Figure 3.2: More sophisticated situations are also possible

lanes). When there are no buses, all lanes are allocated to general traffic. When a bus approaches and on the bus driver's request, the right-hand lane is reserved for it. Once the bus has passed, the reserved lane is returned to general traffic (Figure 3.1). However, if all lanes are already jammed, the system switches to "static mode", *i.e.* the right-hand lane is allocated to buses permanently. More sophisticated situations are also possible (Figure 3.2). In (3.2a & 3.2b) a one-way sharing lane in the narrow part is alternatively used by opposite running buses. When two opposite running buses are present at the same time, two lanes are allocated to them and only one lane is devoted to personal vehicles (3.2c). In (3.2d & 3.2e) the change in orientation of the central lane is proposed. Dynamicity is less considered in this case because, for security reasons, long periods of time are needed to be sure that no buses are still engaged in this lane at the time of change.

3.1.2 Overview of the System

The global view of our approach is shown in Figure 3.3. The infrastructure is based on a multi-lane road with sensors allocated under and near it, vertical and horizontal signs with associated actuators, and a management system connecting them. On the road, sensors, collecting traffic situation information and specific demands from authorized drivers (buses, emergency vehicles, *etc.*), are able to communicate observed situations to the management system. This system is able to decide on appropriate reactions complying with the management policy elaborated by traffic authorities, and propagates appropriate commands to vertical and horizontal road signs as well as the new system status to all users by appropriate media (radio, GSM, Wi-Fi, *etc.*) to inform them of expected behavior (use of lanes reserved for a bus by non-priority drivers if no bus is expected, leaving a lane which is now reserved for priority drivers when a bus is approaching, *etc.*).

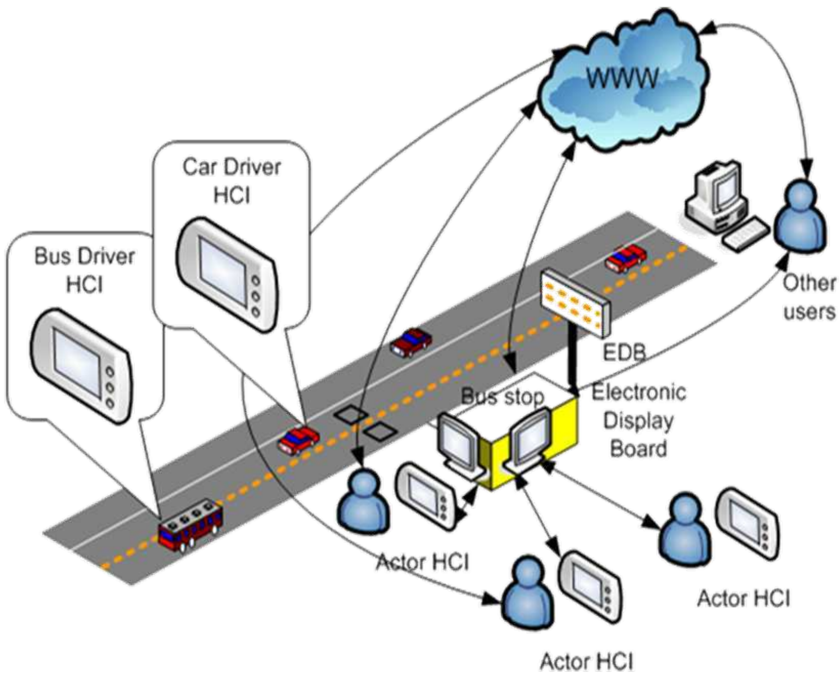


Figure 3.3: Global view of the dynamic lane allocation environment

3.1.3 System Architecture

The system architecture shows the main components and information exchanges (Figure 3.4). In and near the road lanes, information sensors are installed to collect the state of the traffic and priority requests. In and near the lanes, vertical and horizontal signalings are also implemented, able to receive positioning commands from the management system. Main users of the system are active users such as bus drivers, ambulances, firemen, in some cases trucks with their vehicles that require behavioral changes on lane allocation. Passive users, mainly personal vehicles, are receivers of the imposed modifications either by observation of the display changes, or by more direct means – information sent to in-vehicle User Interfaces. Other circulation infrastructure users are pedestrians, cyclists, *etc.*, also concerned by infrastructure changes. The last category of users consists of the traffic and information managers, who are either decision-makers of regulation strategies or concerned by traffic information broadcast to external information systems such as specialized media, TV, *etc.*

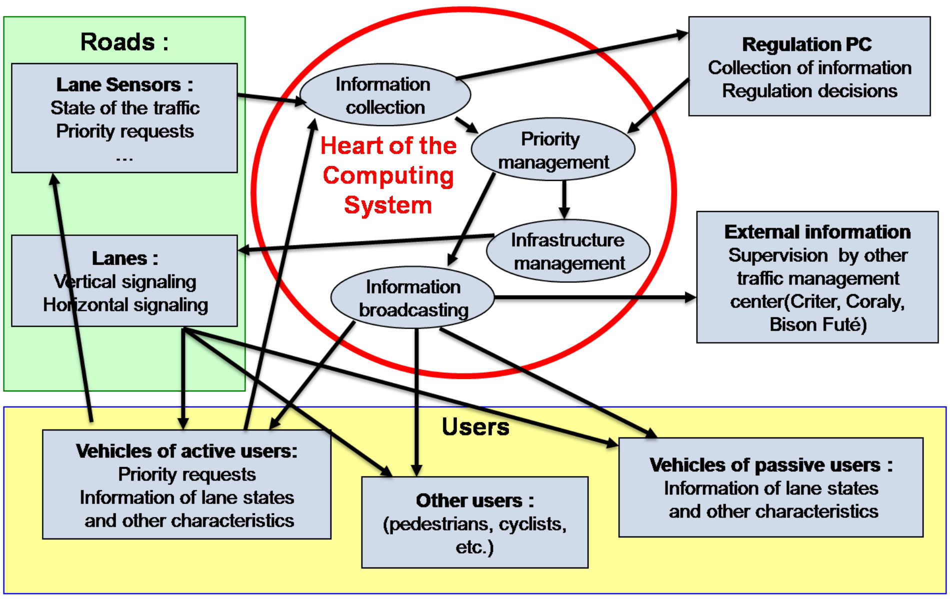


Figure 3.4: System architecture diagram[1]

[1]In "External information", Criter: Centralized traffic management of Lyon Metropolis, in charge of real time traffic on the 2400 km of roads (green waves, public transportation priority, etc.); Coraly: Coordination and regulation of traffic on high ways in the Lyon area; Bison-Futé: Real time traffic information provider

All the elements are included in the management system, *i.e.* collecting, aggregating, processing and broadcasting the appropriate information to all users. The system architecture (Figure 3.4) involves these main elements, namely:

- sensors in the lane responsible for collecting traffic situation information and priority requests;
- vehicles of users who request priority and receive information regarding the state of the lanes (allocation of lanes, authorized or unauthorized dynamic priority requests) for on-board display;
- vehicles of passive users who cannot take action, but who receive information regarding the state of the lanes (whether or not they are dynamically allocated to different user categories: prohibited lanes, lanes reserved for priority vehicles, unmarked lanes);
- the regulation PC. This is a vital component for coordination, which chooses the lane management mode: static allocation when traffic is heavy with a large number of buses or prioritized vehicles justifying static allocation of prioritized lanes, or dynamic allocation if there is less bus traffic and there are more personal vehicles. Action taken by the regulation PC is not systematic, and occurs only when the management mode changes (from static to dynamic and vice versa).

The core of the management (computing) system is structured in four modules:

- the information collection module, responsible for receiving the information sent by in-environment sensors and propagating it to the priority managers and to the Regulation PC;
- the priority manager is responsible for applying the road lane allocation strategy as decided by the Regulation PC: static mode if traffic is heavy, or dynamic mode if it seems better to blend the traffic and make the allocation on request;
- the infrastructure management propagates priority manager choices to vertical (variable message signs) and horizontal (on the road) signaling;

- the information broadcasting module is charged with propagating the current situation to all circulation users and media.

In this dynamic system, real-time data exchange between vehicles and infrastructure must take place as automatically as possible and in compliance with environmental constraints. Human participation in these activities must be limited to essential situations, as human attention is the most critical and must be primarily devoted to driving activities. In this way, communication must be established between things (in the IoT approach), allowing actuators (in vehicles), sensors (on the road) and other system elements to communicate mainly without human participation.

3.1.4 Location-Based Aspects

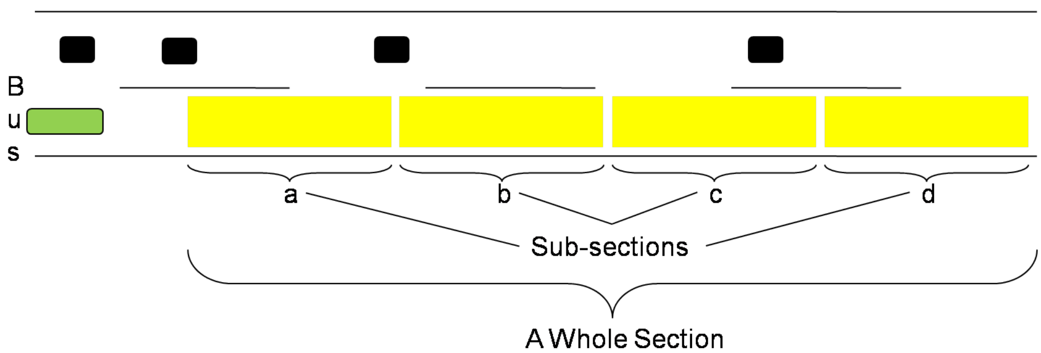


Figure 3.5: A whole section divided into several sub-sections

After having identified the main elements of the system, we find that both active and passive users are apparently mobile ones because they are in the vehicles running on the route, while the section potentially able to be reserved for priority vehicles is fixed a priori since sensors (detectors) and actuators (signaling) must be implemented in advance. However, the potential section could be divided into several sub-sections (Figure 3.5), the number of which varies with actual situations (*i.e.* the whole section is long or short), in order to make allocation more flexible. In any case, for a priority vehicle, the allocation and reversion decisions depend largely on its location (relative position regarding the potential section), and its velocity:

- If the priority vehicle is too far from the potential section, at least it cannot be detected by the nearest sensor, and even if the driver sends a priority request via the on-board interface, that would not be taken into account: thus no allocation would be made.

- If the priority vehicle is detected by the nearest sensor, but travels at a fairly low speed, the whole section would not be allocated, but only the first one or two sub-sections. Otherwise, the section could be clear (passive users' vehicles all leave) while the priority vehicle has not arrived. On the contrary, if the priority vehicle is relatively fast, then it sounds better to allocate the sub-sections behind, in order to make sure that normal vehicles have enough time to leave the section.

- Signaling on a sub-section could also be different during allocation, in the sense that those nearer to the priority vehicle could be "obliged" and those a little far away could be "still available" but with an action of "giving way" suggested – gradual change.

3.1.5 User Interfaces

The dynamic allocation system cannot work without every user's participation and compliance with the rules. To achieve that, one of the major prerequisites is that information is delivered to all concerned users and is well perceived by them. Examples of embedded in-vehicle interfaces are shown in Figure 3.6: the GPS, LANE INFO, and the REPORT ACCIDENT entries are accessible to all users, while the PRIORITY page is reserved for priority vehicle users. For passive users, we also propose a simplified notification interface, as shown in Figure 3.7.

3.2 The Simulation Environment

After designing the system architecture and identifying the main elements of the system, we would like to know whether the whole system works, whether the technology meets our requirements, and how the users of Smart City find it. So we create

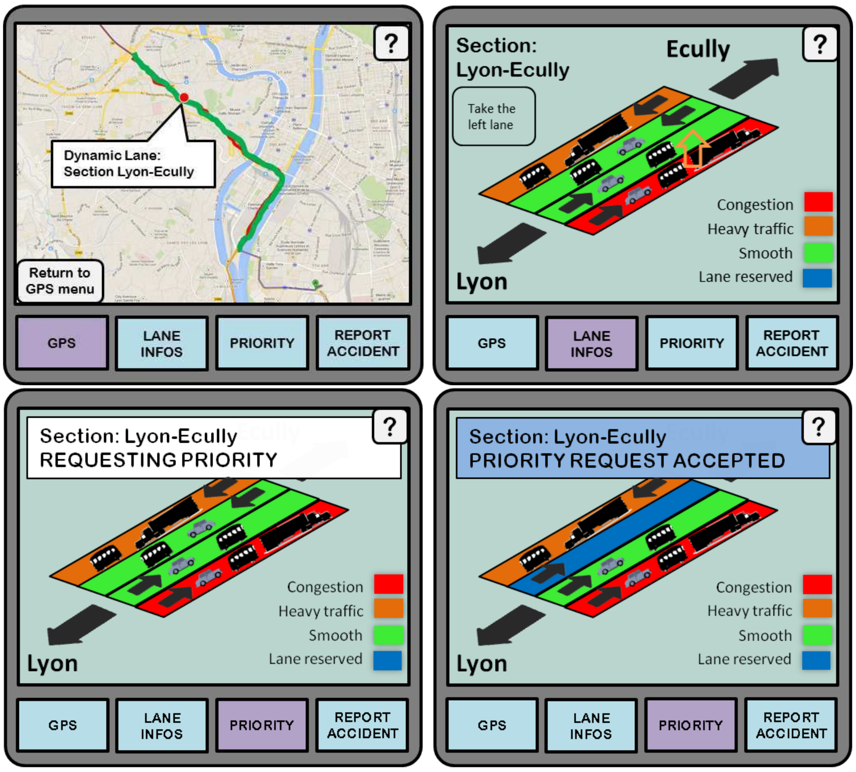


Figure 3.6: Examples of in-vehicle interfaces

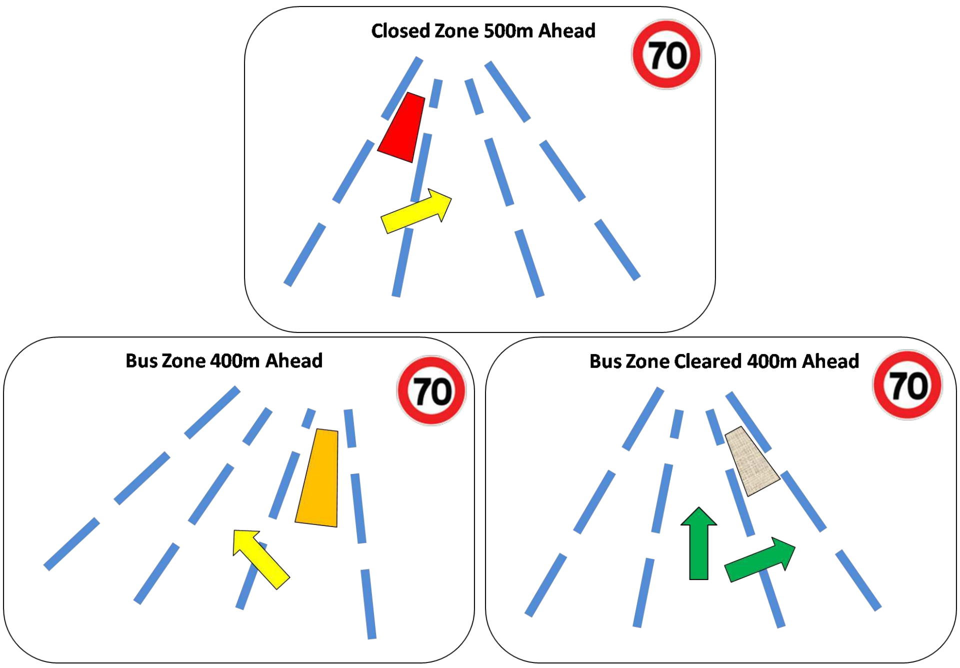


Figure 3.7: Simplified notification interfaces for passive users

a simulation environment, the functionalities of which are shown in Figure 3.8.

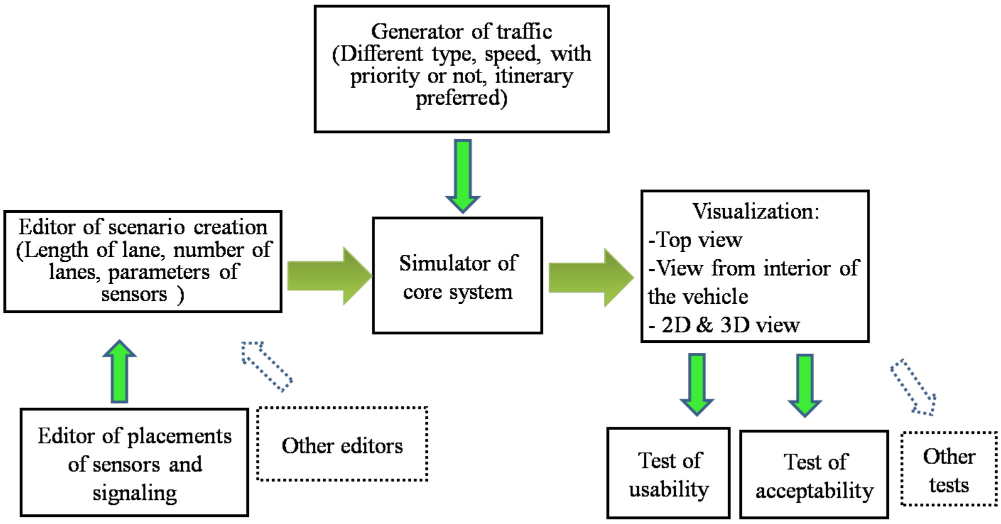


Figure 3.8: Simulation environment functionalities

The central unit is the core system simulator, where we simulate information flow and exchange among the elements of the system, and, especially, the behavior of these elements in different situations. Pre-defined rules can be added or appended to the simulator in a modularized way. The core system simulator is discussed in more detail in a later chapter.

Besides the core system simulator, other parts of the environment are also modularly designed, which assures the flexibility and extensibility of the whole environment. Modules can be easily added or removed without disturbing the function of other parts.

As in other simulation systems, the environment has its inputs (traffic and road scene) and outputs (results which record the behavior of all vehicles and the state of the scene). The scenario creation editor is used to specify simulation parameters and lane state, such as lane length, number of lanes, duration of simulation, number of sensors and initial lane status, as well as the geometrical characteristics of the road. In our case of dynamic management of road lanes, we design a graphic scenario editing tool as shown in Figure 3.9.

A sensor editor toolkit is also included, able to set sensors at appropriate locations and to adjust their effective range or other parameters. All parameters

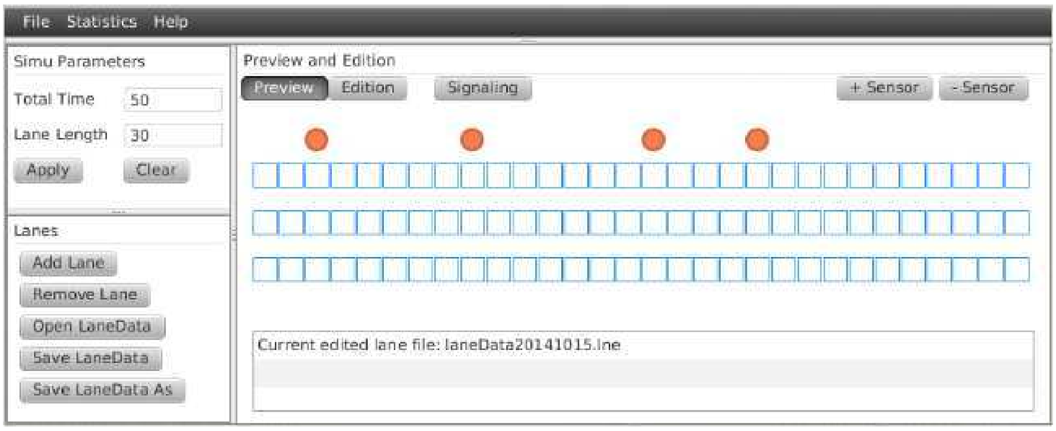


Figure 3.9: Scenario editing tool of the simulation environment

are saved into a "lane data file" via the scenario editing tool as an input to the simulator. We also provide a command line interface to generate the lane file for advanced users, especially when there are a large number of sensors or signaling to set. Doing this manually via a graphic tool does not seem to be a good idea. Moreover, this command line script is highly configurable, so that it can meet future needs when the simulator expects more input items. Another input is vehicles, which are produced by the traffic generator. The implementation of this module in our simulation environment is shown in Figure 3.10.

Figure 3.10: Traffic generator tool of the simulation environment

We have a list of all the vehicles on the left-hand side, and by selecting one, we can modify its parameters such as type, speed, initial lane, launch time, *etc.* Vehicles can be easily created with default values or removed from the list. We also provide a command line script to help generate large amounts of vehicles where we only have to specify two arguments: total number of vehicles and percentage of those with priority. The list of vehicles is stored in a "vehicle data file", which is later used by the simulator.

Another part of the simulation environment is the visualization tool, in which we can observe the behavior of vehicles via the 2D top view. The results in different scenarios with 3D viewing from the interior of the vehicle can be used for further tests such as usability ([Barnum 2010]), acceptability, *etc.*

3.3 Summary

In this chapter, we presented a solution design for a traffic lane management problem, including the architecture of the system, the user interfaces, and the simulation environment, as a case study of our approach of combining LBS and IoT. While we gave here an overview of the simulation environment, we introduce more details of the core system simulator in the next chapter.

Chapter 4

Core System Simulator and Technology Validation

In this chapter, we will provide a detailed presentation of the core system simulator in the simulation environment, including model design, implementation and observation related to theoretical aspects, before introducing a technology validation based on a proof of concept ([Wang *et al.* 2015]).

4.1 Simulator Model Design

The basic idea is to create a road scenario, with all input data, let the vehicles run from left to right respecting certain rules, and take down the state of the scenario and behavior of the vehicles throughout the simulation as output data. Meanwhile, information collection, priority management, infrastructure management and information broadcasting are all taken charge of, assured by communication and collaboration between the elements.

4.1.1 A Two-Layer Model

We propose a two-layer general model design for the simulator: traffic layer and communication layer, as shown in Figure 4.1. The strategy, rules and logic of the vehicles' behavior are placed in the traffic layer. In the communication layer, there are corresponding methods that help complete the task when collaboration with other elements is needed, *i.e.* a lane-changing behavior supported by the following vehicle in the target lane. The "macroscopic" decoupling of these two layers and the "microscopic" coupling of the elements in each layer can provide a high degree of flexibility.

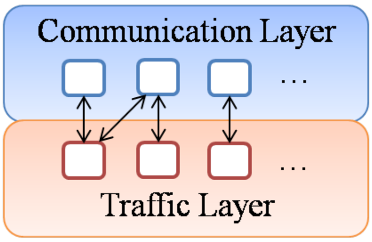


Figure 4.1: The two-layer model of the simulator

4.1.2 Simple Traffic Modeling

We firstly propose a simple traffic model, the form of which is similar to a cellular automaton. The lane is divided into a number of cells of equal size, where each one can not be occupied simultaneously by more than one vehicle. For each time step, a vehicle tries to advance one cell by checking the availability of the next cell:

$$P_v(t+1) = \begin{cases} P_v(t)+1 & \text{if next\_cell.empty()==true} \\ P_v(t) & \text{if next\_cell.empty()==false} \end{cases} \tag{4.1}$$

where $P_v(t)$ stands for the position of a vehicle at time t. The act of availability checking requires information on the current traffic state, to simplify the situation in the simulation, which the lane will directly provide to the vehicle. In order to maintain the autonomy of each element as far as possible, we propose to use a Multi-Agent approach ([Doniec *et al.* 2008, Ksontini *et al.* 2014]) implemented with multi-thread technology, so that the concurrency issue is handled by the machine itself on which the simulation is carried out. The order that who gets the answer first when several vehicles are checking with a lane, is consequently not hard-coded.

Moreover, we also introduce an abstract public clock as a timing to insure that a vehicle tries at least once to move, but does not actually move more than once at time t. The task diagram is shown in Figure 4.2. Tasks for the clock are executed sequentially, giving a flag value to each auto. The "Check" tasks for each auto are carried out in parallel all with "lane", as is the case of a task manager of a multi-core processor. An auto gets the answer, "A" for "Advance", or "N" for "None".

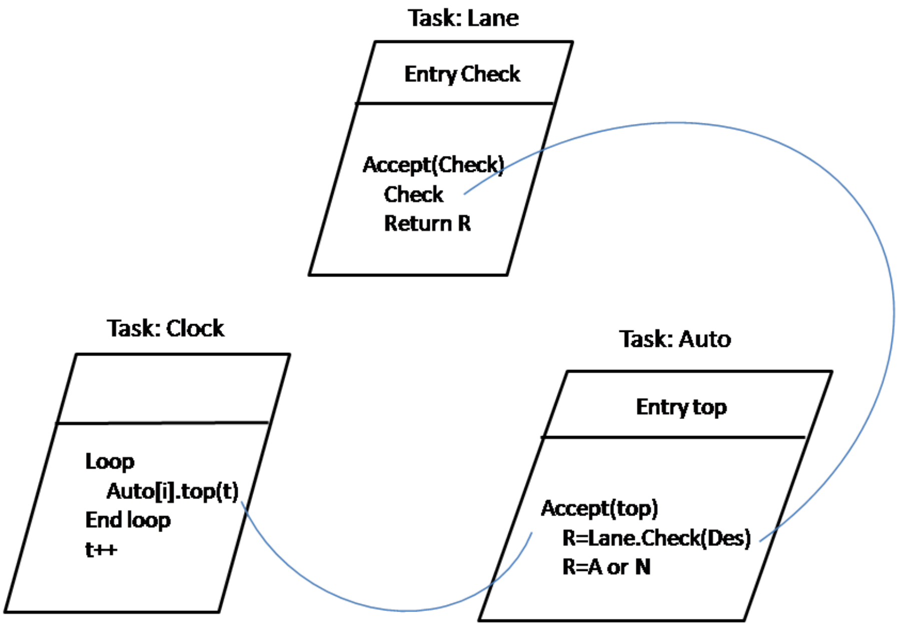


Figure 4.2: A task diagram among instances of auto, lane and clock

Some complexity arises since a road contains not only one lane, and in our lane management system simulator, lane-changing behavior has to be taken into consideration. We propose the strategy that a vehicle communicates with the road and checks first with the current lane of the road: if the answer is Advance, it advances just as before and if the answer is None, it checks with other lanes. If it gets an Advance from certain other lanes, it takes a lane-change into that lane and advances by one cell. If checks are iterated on all other lanes and the answer is still None, then the vehicle stays for this round. The definition of (4.1) is still valid under the condition that we introduce a second dimension that measures the variation of lane number $L_v(t)$, so that the position of a vehicle at time t is defined as:

$$S_v(t) := (P_v(t), L_v(t)),$$

and we have

$$S_v(t+1) = (P_v(t+1), L_v(t+1))$$

$$S_v(t+1) = \begin{cases} (P_v(t)+1, L_v(t)) & \text{if current\_next.empty()==true} \\ (P_v(t+1), L_v(t) \pm 1) & \text{if current\_next.empty()==false} \end{cases} \tag{4.2}$$

where $L_v(t) \pm 1$ stands for the left-hand or the right-hand lane, provided that it does not exceed the limit of the road's lane number and that a lane does not receive a second check from the same vehicle during one road check session. The modified task diagram is shown in Figure 4.3. "Road" acts as man-in-the-middle between "Lane" and "Auto". An Auto receives such answers as A(Advance), TL(Turn Left), TR(Turn Right) and N(None).

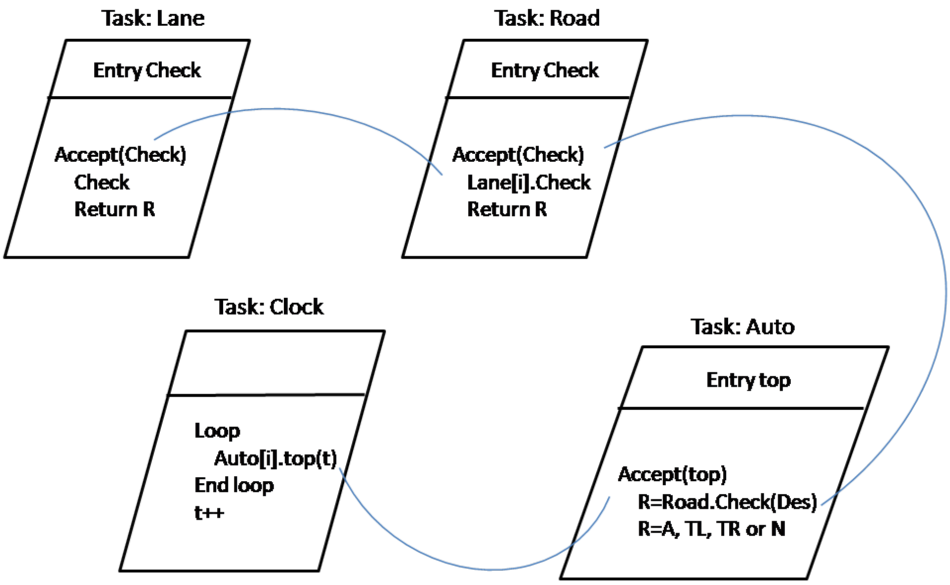


Figure 4.3: A task diagram among instances of auto, road, lane and clock

4.1.3 Modeling of Priority and Dynamicity

With the basic traffic model established, we need to deal with priority and dynamicity which are based on certain rules. So we introduce a labeling method for the lanes, each cell of a lane has three states:

- "close to all", none of the vehicles has the right to occupy;
- "open to all", every vehicle has the right to occupy, if it is free;
- "reserved", only a certain type of vehicle has the right to occupy.

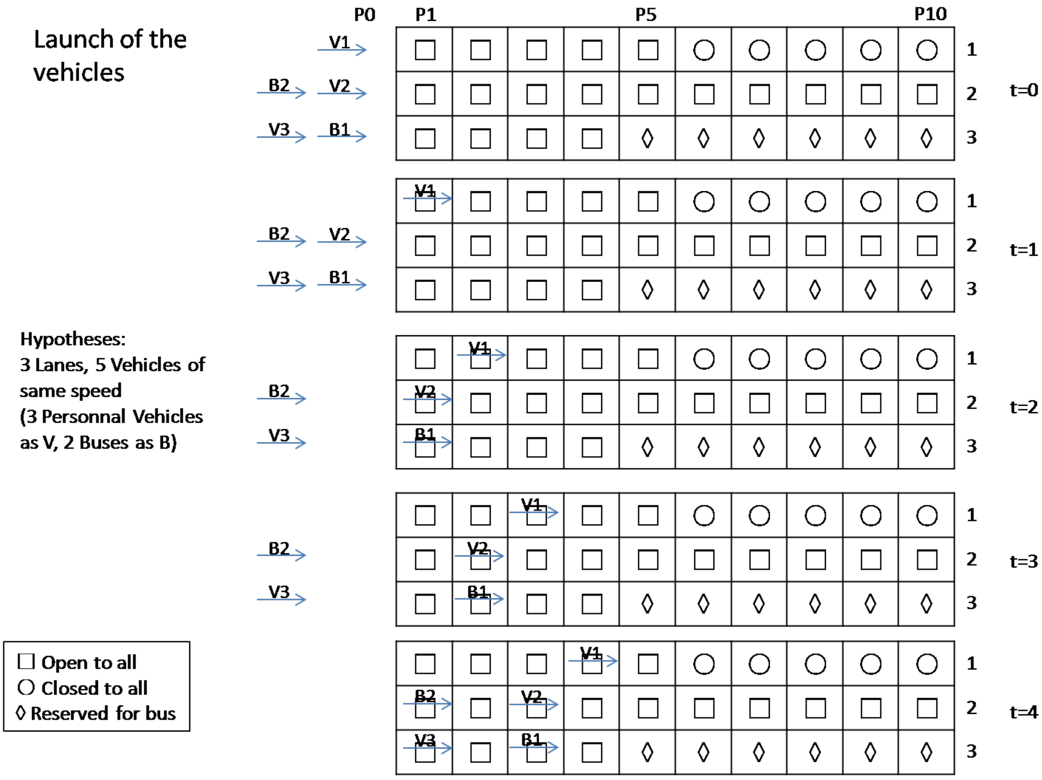


Figure 4.4: A simulation launch diagram

A set of consecutive "close to all" cells could signify a blocked section caused by an accident or construction work. Respectively, we label the vehicles with integer numbers as their priority, *i.e.* personal vehicles with 1, buses with 2, ambulances with 4: the larger the number, the higher the priority. In consequence, the reserved cells can adapt to different degrees of priority. In a scenario where a section is initially reserved for a bus with priority 2, an ambulance also has the right to use it due to higher priority; however, a bus cannot enter the section reserved for an ambulance due to lower priority. Thus the priority check is easily conducted by comparing two integers.

With the basic traffic model and priority management in hand, we could imagine the simulation scenario as in Figure 4.4. A road contains three lanes: part of the first lane is blocked, part of the third lane is reserved for the bus, and the middle lane is open to all vehicles. Five vehicles (3 personal vehicles and 2 buses) are waiting at position P_0 when $t = 0$. Every vehicle has its own start time to avoid congestion at the entrance of the road. All the vehicles have entered the road when $t = 4$.

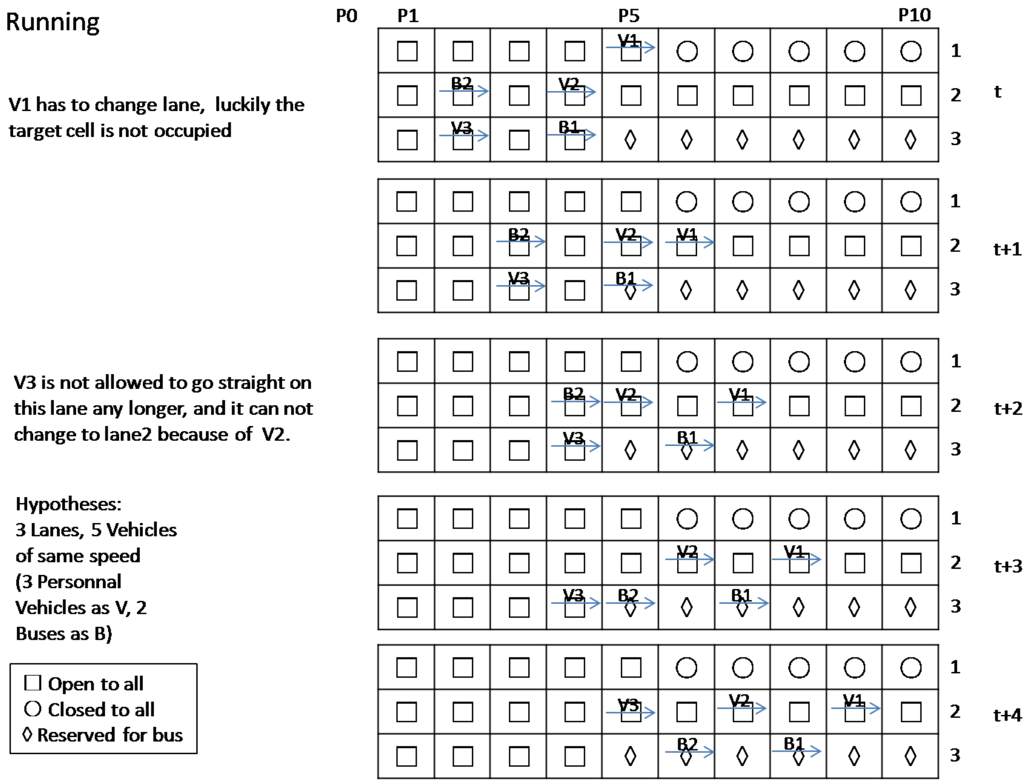


Figure 4.5: A simulation execution diagram

As the time step increments, the scenario could evolve to something as in Figure 4.5. At time t, the vehicle $V1$ comes across the blocked section. Then, when it checks with the road, it gets a lane-changing response and occupies the cell $(P6, 2)$; on the other hand, the bus $B1$ enters the reserved area without hesitation since it just has the priority. From $t+1$ to $t+2$, we observe that all the vehicles except $V2$ advance by one cell, because $V2$ is blocked by $V1$. This is one of the possible results because if the vehicle $V1$ finishes checking first and frees the cell it occupied, then $V2$ could advance normally. From $t+2$ to $t+3$, $V3$ has no right to enter the reserved area and checks with the road for a possible lane-changing, only to find that it is blocked by $V2$, so it stays; the bus $B2$ is also blocked by $V2$, but it suddenly finds that there is a reserved area in the third lane, so it carries out a fortunate lane-changing to $(P5, 3)$.

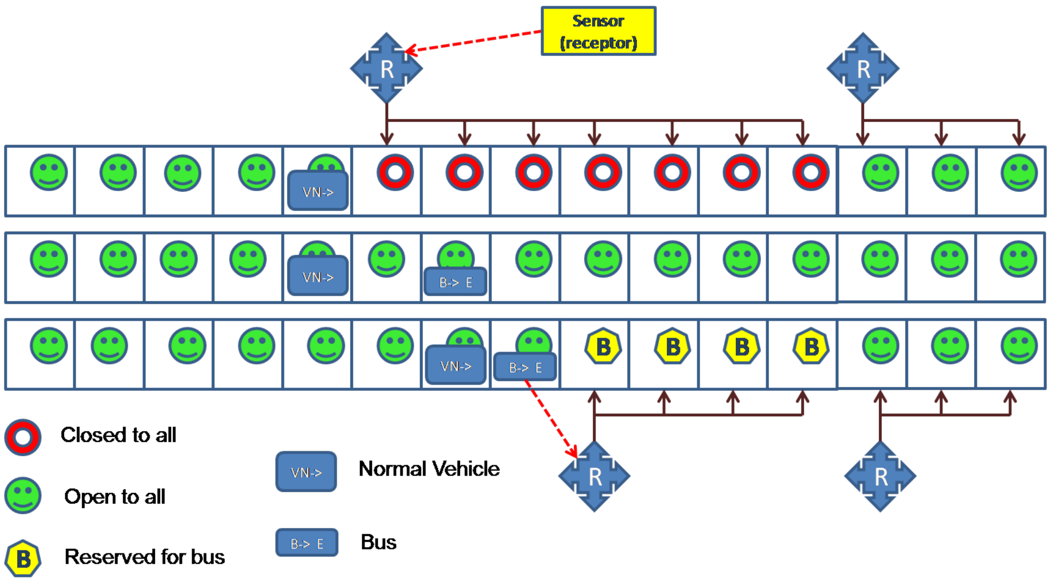
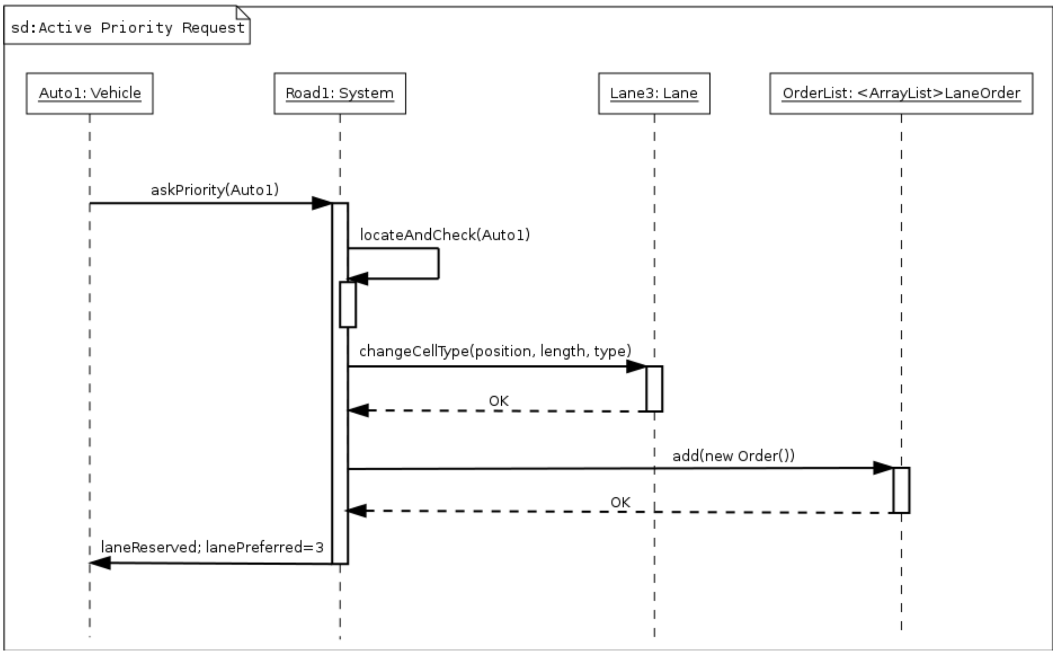


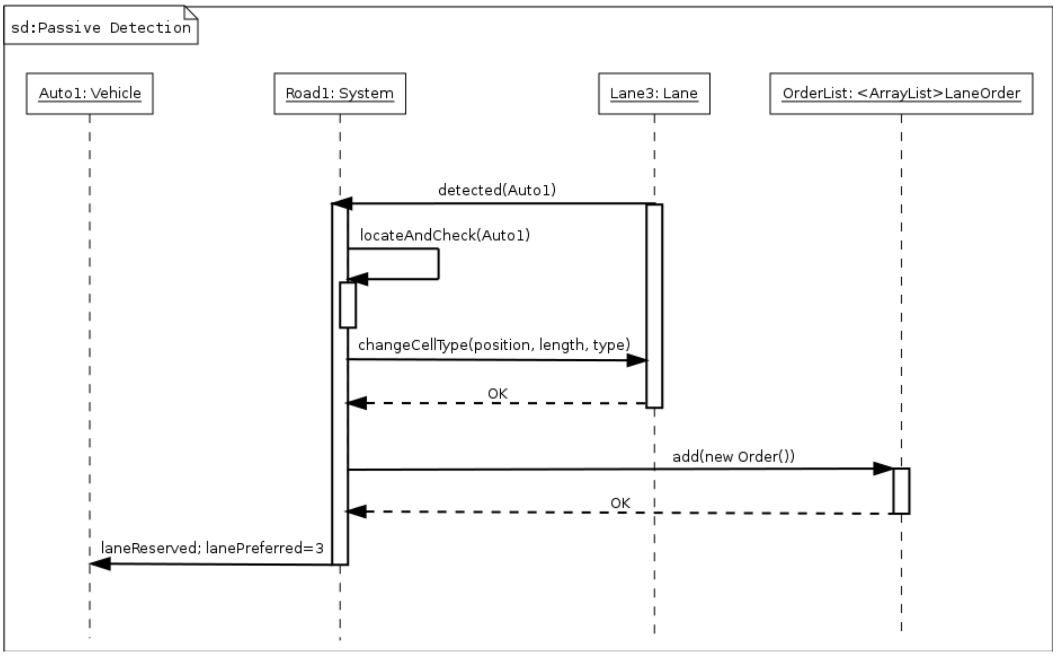
Figure 4.6: A more complete configuration considering dynamicity

The vehicles are running but we have not seen any dynamicity up to now. The reserved areas in Figure 4.4&4.5 are beautiful but how come they exist from the very beginning instead of being location-based? Then, we add sensors to the road (Figure 4.6) to provide at least two types of interaction: active priority request and passive priority zone reservation. The former means that the bus driver sends a request to the management center, the nearest sensor in front takes charge of the request and tries to make a reserved section; the latter means that the bus is detected automatically by a sensor, and then checks the possibility to make a reservation with the management center. So the reservation will be dynamic and depends on the location of the bus. Since the sensors collect the traffic state information, when an accident is detected by a sensor, it can also try to create a closed-to-all section.

We can observe more specifically the allocation process in the sequence diagrams as in Figure 4.7. We do not separate Road and System (Regulation PC) to keep the diagram compact, and from the point of view of a vehicle, it does not see any difference. The reason why we introduce an instance of "OrderList" is that it is necessary for recovering the lane when the reservation is over. The order contains the start point and the length of the allocation area as well as the vehicle ID. When



(a) Active priority request



(b) Passive detection by a sensor

Figure 4.7: Sequence diagrams considering dynamicity

the bus leaves the reserved section, the sensor will notify the road to change the lane to its initial state. The number of orders in the list during a period of time can also help the system to decide whether to pass to static allocation mode.

Active or passive, when the allocation is made, the bus is notified with a specific reserved lane number and a flag value "laneReserved". Thus the bus tries to change to or keep in the reserved lane in advance, otherwise it could miss it due to other possible lane-changing choices. Since the allocation concerns only a section, other vehicles that are far from the bus are not affected and can still use the reserved lane.

4.1.4 More Realistic Traffic Modeling

The simple traffic model of section 4.1.2 is easy to understand and implement. However, if we would like to see more realistic behaviors such as the process of acceleration or deceleration and velocity variation, it requires more realistic traffic modeling.

4.1.4.1 Intelligent-Driver Model

We propose to use the Intelligent-Driver Model (IDM) [Treiber *et al.* 2000] as an alternative to the simple traffic model, which is a time-continuous, longitudinal car-following traffic model. It is also a microscopic model that combines a vehicle and its driver and treats them as an active particle. In this model, a series of vehicles are present in the lane one after the other, and the traffic state is determined by the position and speed of each vehicle.

For a single driver-vehicle entity, it advances by applying a certain value of acceleration, in general either throttle or brake. Then, speed and position are calculated via the classic kinematic equations. The decision-making of the acceleration value is influenced only by:

- its own speed v,
- the bumper to bumper distance s from the leading vehicle,

- the relative speed (speed difference) Δv with the leading vehicle.

We obtain the following equation:

$$\frac{dv}{dt} = a\left[1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{s^*(v, \Delta v)}{s}\right)^2\right] \tag{4.3}$$

where $s^*(v, \Delta v)$ is the desired dynamical distance depending on its own speed and the speed difference

$$s^*(v, \Delta v) = s_0 + \max\left[0, \left(vT + \frac{v\Delta v}{2\sqrt{ab}}\right)\right], \tag{4.4}$$

and the bumper to bumper distance s complies

$$\frac{ds}{dt} = -\Delta v. \tag{4.5}$$

The model parameters are shown in Table 4.1. So the mathematical form of the IDM model equations is that of coupled ordinary differential equations. They are differential equations since, in (4.3), the dynamic quantities v (speed) and its derivative dv/dt (acceleration) appear simultaneously. They are coupled since, besides the speed v, the equations also contain the speed $v_l = v - \Delta v$ of the leading vehicle, with (4.5) coupling the gap s to the speeds of the two vehicles.

The acceleration is divided into a "desired" acceleration $a[1-(\frac{v}{v_0})^{\delta}]$ on a free road ($s \to \infty$), and braking decelerations induced by the front vehicle. The acceleration on a free road decreases from the initial acceleration a to zero when approaching the desired speed v_0.

Table 4.1: Parameters of IDM

Parameter	**Unit**	**Description**
v_0	m/s	Desired speed when driving on a free road
a	m/s^2	Acceleration in everyday traffic
b	m/s^2	Comfortable braking deceleration in everyday traffic
T	s	Desired safety time headway when following other vehicles
s_0	m	Minimum bumper-to-bumper distance to the front vehicle
δ	none	Acceleration exponent

Table 4.2: Parameter examples of IDM [Treiber 2010]

Parameter	**Value Car**	**Value Bus**	**Remarks**
Desired speed v_0	$90km/h$	$72km/h$	For city traffic, one would adapt the desired speed while the other parameters essentially can be left unchanged
Acceleration a	$2.4m/s^2$	$1.9m/s^2$	Very low values to enhance the formation of stop-and-go traffic. Realistic values are $1-2m/s^2$
Deceleration b	$3.6m/s^2$	$1.9m/s^2$	Very high values to enhance the formation of stop-and-go traffic. Realistic values are $1-2m/s^2$
Time headway T	$1.5s$	$1.8s$	Recommendation in Germain driving schools: $1.8s$; realistic values vary between $2s$ and $0.8s$ and even blow
Minimum gap s_0	$2m$	$2m$	Kept at complete standstill, also in queues that are caused by red traffic lights

The braking term is based on a comparison between the desired dynamical distance s^*, and the actual gap s from the front vehicle. If the actual gap is approximatively equal to s^*, then the braking deceleration essentially compensates the free acceleration part, so the resulting acceleration is nearly zero. This means that s^* corresponds to the gap when following other vehicles in steadily flowing traffic. In addition, s^* increases dynamically when approaching slower vehicles and decreases when the leading vehicle is faster. As a consequence, the imposed deceleration increases with:

- decreasing distance from the leading vehicle (the temptation to maintain a certain safety distance);
- increasing own speed (the safety distance increases);
- increasing speed difference with the leading vehicle (to avoid the dangerous situation when approaching the leading vehicle too quickly).

Table 4.2 shows some example values of IDM parameters. Actually, each driver-vehicle entity can have its own set of parameters, *i.e.* for a bus, the values of

v_0, a and b could be relatively low; for a careful driver, he or she could have a longer safety time headway T to provide enough reaction time; aggressive driving behaviors could be characterized by short T combined with high values of v_0, a and b.

In the IDM model, we also have a public clock that updates the whole traffic state with a finite numerical update time interval Δt. Also, it is assumed that during Δt, all the vehicles have a uniformly accelerated motion, which is an approximation of the solution of the coupled differential equations. For each driver-vehicle, the new speed, new position and new gap are calculated by:

$$
\begin{aligned}
&\text{new speed:} && v(t+\Delta t) = v(t) + (\tfrac{dv}{dt})\Delta t;\\
&\text{new position:} && x(t+\Delta t) = x(t) + v(t)\Delta t + \tfrac{1}{2}(\tfrac{dv}{dt})(\Delta t)^2;\\
&\text{new gap:} && s(t+\Delta t) = x_f(t+\Delta t) - x(t+\Delta t) - L_f.
\end{aligned}
$$

Where $\frac{dv}{dt}$ is the IDM acceleration calculated at time t, x is the position of the front bumper and L_f represents the length of the front vehicle. For the first vehicle of the lane that has no front vehicle, we just set the gap to a very large value such as $10000m$.

4.1.4.2 Lane-Changing Model

Just as in simple traffic modeling, lane-changing behavior cannot be ignored. However, in reality, normally, we do not make lane change decisions by simply checking the availability of the space right in front of us in a target lane, as this could either disturb the other vehicles following or accidents could occur. What's more, in the tactical stage, a driver usually prepares and initiates a lane change by advance acceleration or deceleration, and may receive the cooperation of drivers in the target lane [Hidas 2005]. So when we consider a lane change, we make sure it is both safe and desirable [Gipps 1986], that nobody would be in danger and that it would improve our local traffic situation.

We propose to use the model "Minimizing Overall Braking Induced by Lane change" (MOBIL) [Kesting *et al.* 2007] to complete IDM in multi-lane situations. MOBIL is ideal for car-following models because it also operates on accelerations,

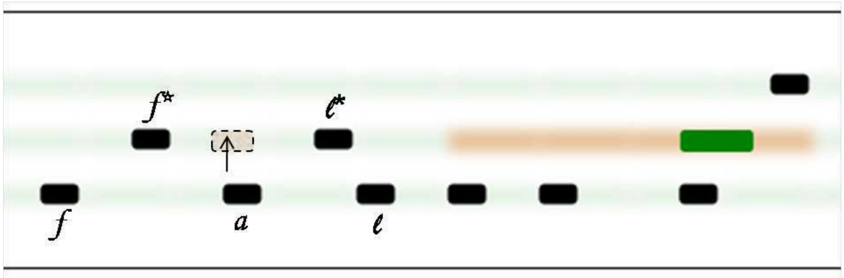


Figure 4.8: Vehicle "a" considering lane change to the center lane

making it possible to reuse the IDM equations and to allow for a compact model formulation with quite a few additional parameters. We can imagine a scenario in which a vehicle tries to move from the median lane to the center lane as shown in Figure 4.8: f and l denote, respectively, the current following vehicle and the leading vehicle. Likewise, f^* and l^* represent the vehicles in the target lane after the lane change. Firstly, the safety criterion is modeled by

$$acc'(f^*) \geqslant -b_{\text{safe}} \quad , \tag{4.6}$$

where acc' stands for the IDM braking deceleration imposed on the following vehicle f^*, which should not exceed a certain given limit b_{safe}. So the safety criterion's dependence on the gap, the follower's speed and the approaching rate are established via the IDM acceleration. If the following vehicle in the target lane is faster than its own speed, larger gaps between the following vehicle and its own position would be required. On the other hand, if the following vehicle in the target lane is slower, smaller gaps are acceptable. Moreover, since the IDM model is a collision-free car-following model, crashes due to lane changes are automatically excluded as long as the limit in (4.6) is well below the maximum possible deceleration b_{max}.

Secondly, the incentive criterion is measured by weighting the driver's own advantage on the target lane (increased acceleration) against the disadvantage imposed on other drivers (increased braking deceleration) via a politeness factor p

whose values are typically less than 1,

$$\underbrace{acc'(a) - acc(a)}_{\text{driver}} > p \left(\underbrace{acc(f^*) - acc'(f^*)}_{\text{new follower}} + \underbrace{acc(f) - acc'(f)}_{\text{old follower}} \right) + \Delta a_{th} \tag{4.7}$$

where acc' stands for the accelerations after a possible lane change and acc is the current IDM accelerations. The two terms on the left denote the driver's own advantage, namely that the driver can go faster in the new lane; the first term on the right describes the combined disadvantage of the two immediately affected following vehicles, weighted with p; finally, the switching threshold Δa_{th} on the right is added to avoid lane changes if the overall advantage is only marginal compared with a "keep running in the current lane" directive. Table 4.3 shows some example values of the MOBIL model's parameters. The value of politeness p can lead to different behaviors:

- $p = 0$, a purely selfish behavior but still respecting the safety criterion;
- $p \in]0, 0.5]$, a realistic behavior: advantages of other drivers have a lower priority, but are not ignored;
- $p > 1$, a very altruistic behavior;
- $p < 0$, a malicious personality who takes pleasure in thwarting other drivers even at the cost of own disadvantages.

The threshold Δa_{th} influences lane-changing behavior globally, while the politeness

Table 4.3: MOBIL parameter examples [Treiber 2002]

Parameter	**Typical Values**	**Remarks**
Politeness factor p	$0 \ldots 0.5$	realistic behavior
Maximum safe deceleration b_{safe}	$4m/s^2$	Must be lower than maximum deceleration of about $9m/s^2$
Threshold Δa_{th}	$0.2m/s^2$	Must be below the lowest acceleration ability (IDM parameter a) of any vehicle type

factor p affects local lane-changing behavior depending on the neighbors concerned. The special case of $p = 1$ and $\Delta a_{th} = 0$ means lane changes take place whenever the sum of the accelerations of all affected drivers increases after the change, or equivalently, the overall decelerations are minimized.

In most European countries, the driving rules for lane usage are explicitly asymmetric: passing in the right-hand lane is forbidden unless traffic flow is congested; the right-hand lane is the default one, the left-hand lane should only be used for overtaking. Implementing these rules requires further parameters and formulas [Treiber 2002].

4.1.5 Comparison and Optimization

At first glance, the difference between simple traffic modeling and the combination of IDM & MOBIL is rather intuitive: the latter is space-continuous and provides decision-making of acceleration values that makes vehicles' behaviors more realistic; the former is both space-discrete and time-discrete, giving the impression that the simulation is made up of a set of traffic state snapshots.

Meanwhile, during the early simple traffic modeling experiment, we discovered

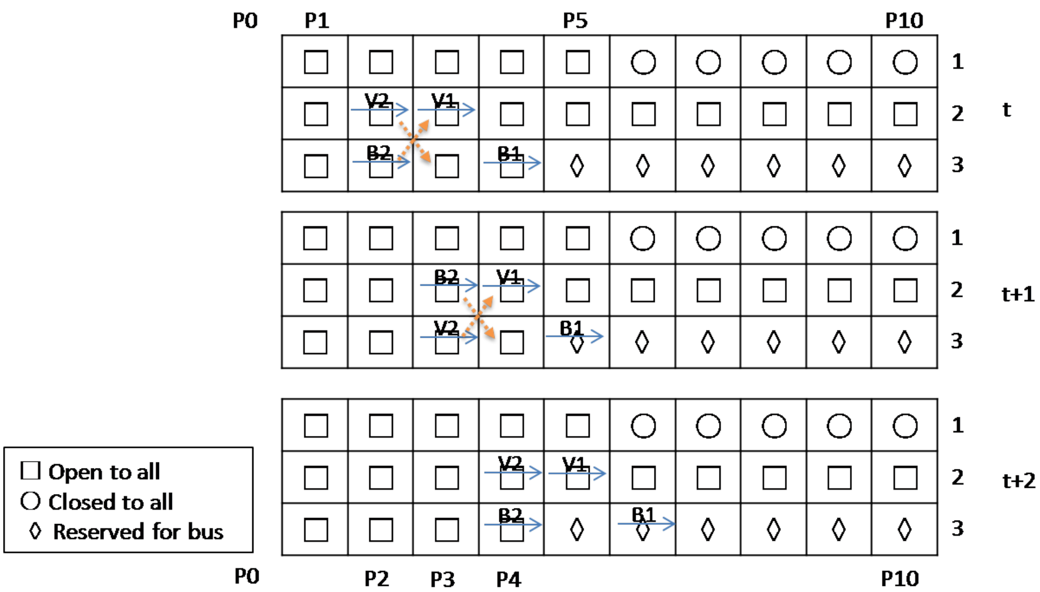


Figure 4.9: A lane hopping observation diagram

some disadvantages, one of which is lane hopping of vehicles. Consider a scenario as shown in Figure 4.9, where two personal vehicles and two buses are running on two lanes. We assume that at time t, $V2$ checks with road first, being blocked by $V1$, it chooses to change lane towards $(P3, 3)$; then $V1$ and $B1$ advance normally; at last, it is $B2$'s turn to check with road, it goes to $(P3, 2)$ because of $V2$. In consequence, we have the state at time $t + 1$, among the four vehicles, only $V2$ and $B2$ have swapped position relatively and the same could happen resulting in state of $t + 2$: $V2$ goes back to lane 2 and $B2$ back to lane 3. Comparing the first with the third state, we could find that the position swapping between $V2$ and $B2$ makes no sense: especially if $V2$ noticed the reserved section in front, it could have kept running in lane 2 even it would cost one turn halt because changing to lane 3 would not do any good, neither to $B2$ nor to itself.

The lane hopping scenario is reproducible but not deterministic, because it depends on the order of each vehicle's road checking. However, probability is high when two vehicles are running side by side with either one preceded closely by a third vehicle. This involves another factor, which is also another disadvantage of simple traffic modeling: there is no speed difference among all vehicles. During a time unit, each vehicle either advances by one cell, including a lane change, or stands still: there is no reaction time, no acceleration or deceleration process. It seems that we should abandon the simple traffic modeling and adopt the more realistic model definitively. However, as a model is never exactly the same as in reality and there will always be a more realistic model, we encourage the demand-driven principle.

In the context of the dynamic allocation of road lanes system, we focus on dynamicity, which depends largely on communication between vehicles and the infrastructure. Simple traffic modeling inherently provides easy access to the communication layer (checking with lanes and the road) so that priority and dynamicity components can be easily added, while for MOBIL it is necessary to introduce new formulas and additional parameters for acceleration decision-making. Besides, simple traffic modeling would be sufficient for off-line visualization, and its inherent randomness also allows various scenarios to be generated for utility and acceptability tests; IDM & MOBIL would be a good choice for real-time operations, *i.e.*

controlling a vehicle in the simulation, or for real-time visualization during the simulation.

We thus decided to use simple traffic modeling. However, this does not prevent us from using the other model in future work. With regard to the disadvantages mentioned earlier, we provide some workarounds. The first is to separate vehicles at launching time and enlarge the inter-vehicle gap by one cell, in order to reduce the occurrence of lane hopping prerequisites. This could be done by modifying input vehicle data: change the launch time and initial lane number for a vehicle in the traffic generator tool (Figure 3.10). The second is to specialize the left-hand lane to a fast lane, which is also in accordance with the lane usage rule in most European countries: the right-hand lane is the default lane, while the left-hand lane should only be used for the purpose of overtaking. The idea is to let a vehicle use the left-hand lane and advance two cells in one time step temporarily, then it must go back to the right-hand lane. The modifications to the model are only adjustment of the order of checking conditions and adding the vehicle's behavior of two-cell-advancing.

4.2 Simulator Implementation

4.2.1 Data Flow

We decided to use Java's multi-thread technology to implement the multi-agent simulator. The data flow global view is shown in Figure 4.10. The core system simulator reads input data from the initialization tool, executes the simulation and, finally, sends output data to the visualization tool. All data are in plain text format.

4.2.2 Main Thread

The main tread is the entry to the simulator, where all the elements are created or instantiated, initialized, and the simulation starts. It contains three parts: initialization, execution and post-simulation. The first part is implemented as shown in Algorithm 1. Since the vehicle class in the initialization tool differs from that in the main thread, we carried out a conversion step, with the help of xstream library

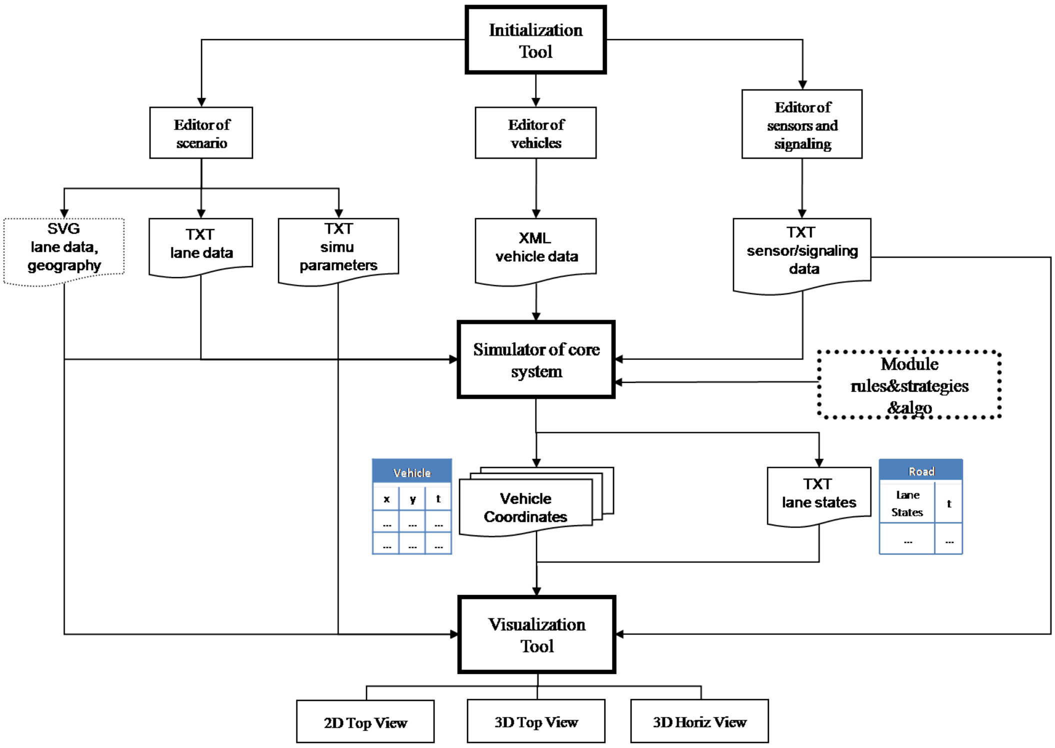


Figure 4.10: Global view of simulator data flow

[Walnes 2013], which converts xml data to Java class instances. In the parameter extraction step, we first read all data lines, and then perform a text string split operation, so that the parameter can be extracted easily by feeding keywords to Java string functions.

Algorithm 1 Initialization part in main thread

```
Load vehicle data from xml file and convert it to vehicle array list
for each vehicle v do
    create Vehicle thread
end for
dataString ← lane data, parameters and sensor data from text file
for p in ParameterList[ ] do
    extractParameter ( p , dataString )
end for
simulation time ← Parameter[0]
total lane number ← Parameter[1]
length of lane ← Parameter[2]
total sensor number ← Parameter[3]
sensor position list ← Parameter[4]
Create Road thread
for i from 0 to total lane number do
    Create Lane with lane length
    Add Lane to lane list[i] of Road
end for
Create Clock thread
```

The execution part is as simple as specifying the main simulation time and calling the start method of each thread instance. Then we put the main thread in a try-catch block to wait (with the sleep method) until the end of simulation time. Finally, the post-simulation part includes data logging and calling the stop method of each thread instance.

4.2.3 Other Key Threads

4.2.3.1 Vehicle Thread

The vehicle thread mainly generates the behavior of an instance of vehicles. The *run*() function is shown in Algorithm 2. Once a vehicle thread starts to run, it always waits for the public clock to change its flag, and it cannot move before its own launching time. Its behavior is controlled by the return value of the "check road" method: advancing by one cell, advancing by two cells, turning left, turning right, or staying. Finally, all the position information is written to output file.

Algorithm 2 *Run*() of vehicle thread

```
Create output file VehicleX
Create header
while not at end of simulation do
    Wait clock
    if time < launch time then
        Continue
    end if
    if flag is false then
        Continue
    end if
    check road
    if get answer A then
        x+1
    end if
    if get answer A2 then
        x+2
    end if
    if get answer TL then
        x+1,y-1
    end if
    if get answer TR then
        x+1,y+1
    else
        Print fail
    end if
    Free current cell
    Occupy new cell
    flag ← false
    Write position to buffer
end while
write buffer to output file
```

The "check road" method is declared as *synchronized*, which means that if a vehicle thread is calling this method, the road thread is locked and cannot accept other calling requests until the current one is finished.

4.2.3.2 Lane Thread

Algorithm 3 Check methods of Lane thread

Require: Vehicle v

function CHECKLANE(Vehicle v)
    if $v.x + 1 >$ lane length **then**
        return "OUT"
    else if cell($v.x+1$) empty && cell($v.x+1$).type $\geqslant 1$ && v.type $\geqslant$ cell($v.x+1$).type **then**
        yield and notify
        return "A"
    else
        yield and notify
        return "NONE"
    end if
end function

function CHECKLANESPEEDX2(Vehicle v)
    if $v.x + 2 >$ lane length **then**
        return "OUT"
    else if cell($v.x+1$) empty && cell($v.x+2$) empty && cell($v.x+2$). type $\geqslant 1$ && v.type $\geqslant$ cell($v.x+2$).type && v.type $=$ cell($v.x+1$).type **then**
        yield and notify
        return "A2"
    else
        yield and notify
        return "NONE"
    end if
end function

Ensure: String answer

A lane thread is rather like a daemon service: it is aware of the state of all the cells in it and waits only for calling of its check methods in the *run*() function. These check methods (Algorithm 3) are lower-level methods since they are not called by a vehicle directly but are called by the road thread.

An instance of vehicle is required as input, and the check is run by verifying the vehicle's position, the availability of the next cell and the vehicle's level of priority. The "speedX2" variant is designed for the fast lane, which requires additional conditions such that the cell after the next one is also empty, the vehicle should not accelerate too soon before entering the zone, *etc.*. Finally, these methods return a string answer to the road thread, which the decides either to return the answer directly to the vehicle or to call another lane thread for checking.

Besides, the lane thread also has a method which can adjust the type of cells in it. All the cells are created as type 1 during initialization, then according to request, for example, one or more cells can be changed to type 2 for the bus, and later back to type 1 for public use.

4.2.3.3 Road Thread

The road thread also acts like a daemon service because in its $run()$ function it does nothing special apart from being ready to react when calling methods arrive. In terms of composition, it is the aggregation of all the lane threads, so that it can call the check methods of each one. Concerning functionalities, the road thread communicates directly with all the vehicles to guide their behaviors, as well as managing reservation dynamicity, with the help of the sensors.

Firstly, we look into the check method of the road thread, which is more complicated than those of the lane thread, as shown in Algorithm 4. The situation is generally divided into two cases: reserved or not. If there has already been a reservation for a vehicle, this vehicle should try to change to that lane if it is not yet there, or keep running in that lane if it is already there, and be ready to enter the reserved zone. This is the reason why we introduced the logical variable “laneReserved” and the integer variable “lanePreferred”, without which a priority vehicle could totally ignore the reservation made for it. Then, to magnify the effect of the reserved zone, a priority vehicle is allowed to use it temporarily as a fast lane, overtaking other normal vehicles, which is also reasonable in reality.

Algorithm 4 Check method of Road thread

```
Require: Vehicle v
function CHECKROAD(Vehicle v)
    answerRoad ← answerLane ← answerLaneSpeedx2 ← emptyString
    if v.x + 1 > lane length then
        return "OUT"
    end if
    if v.laneReserved then
        answerLane ← Lane(v.lanePreferred).checkLane(v);
        if answerLane="A" then
            answerLaneSpeedx2 ← Lane(v.lanePreferred).checkLaneSpeedX2(v)
        end if
        if answerLaneSpeedx2="A2" then
            answerLane ← answerLaneSpeedx2
        end if
        answerRoad ← answerLane
    else
        if v.y ⩾ 0 && v.y + 1 <Max Lane Number && v.x ≠ 0 then
            if Lane(v.y + 1).checkLane(v) = "A" then
                return "TR"
            end if
        end if
        answerLane ← Lane(v.y).checkLane(v)
        if answerLane = "A" then
            if v.type=1 && v.y=0 then
                if Lane(v.y).checkLaneSpeedX2(v) = "A2" then
                    return "A2"
                end if
            end if
            answerRoad ← answerLane
        end if
        if answerLane = "NONE" then
            if v.x = 0 then
                return "NONE"
            end if
            if v.y > 0 && v.y + 1 ⩽ Max Lane Number then
                if Lane(v.y − 1).checkLane(v) = "A" then
                    return "TL"
                end if
            end if
            answerRoad ← answerLane
        end if
    end if
    return answerRoad
end function
Ensure: String answerRoad
```

If there is no reservation, the rules are the same for vehicles with and without priority. The road thread checks the right-hand lane first for a vehicle to comply with the "keep right" principle, then checks the current lane to see the possibility of advancing, and finally the left-hand lane as the last choice. If all the lane checking answers are "NONE", that would also be the road checking answer for this vehicle.

Algorithm 5 DetectAndReserve method of Road thread

```
Require: Vehicle v
function detectAndReserve(Vehicle v)
    if v.type < 2 then return
    end if
    if v.x in listSensorX then
        areaHead ← v.speed∗factorSpeed
        Lane(v.y).changeCellType(sensorX+areaHead, area, v.type)
        if not v.laneReserved then
            v.laneReserved ← True
            v.lanePreferred ← v.y
        else
            v.laneReservedContinue ← True
        end if
        reservOrderList ← new Order(v,Lane(v.y),sensorX+areaHead,area)
    end if
end function
```

Secondly, we discuss the method that detects a priority vehicle and makes a reservation for it (Algorithm 5). The level of priority of a vehicle is equal to its integer *type* attribute, and vehicles with a *type* value smaller than 2 would not be detected. When a bus passes by a sensor, the reserved zone location would be calculated, as well as the length of the area. Then, the zone is created by calling the changeCellType method of the target lane. The "laneReservedContinue" variable applies to the situation in which the vehicle running in the reserved zone is detected by the next sensor so that a second reservation is made. Afterwards, when the vehicle leaves the first zone and heads to the second zone, when its "laneReservedContinue" is *True*, the road thread would not wipe out its lane reservation state and its lane preference (reserved lane number), meaning that the reservation continues. Finally, the reservation is stored in a list under an order form containing all the relevant parameters.

In the modeling section, we presented two kinds of dynamic allocation: passive detection and active priority request. Consequently, there is also another method named requestPriority whose function body is almost the same as the detectAndReserve method, except for the first few lines where the vehicle initiates communication and the nearest sensor in front takes charge of the request.

Algorithm 6 RecoverLane method of Road thread

```
function RECOVERLANE
    mainLit ← reservOrderList.listIterator()
    while mainLit.hasNext() do
        order ← mainLit.next()
        if order.getVehicle().x > order.getAreaEnd() then
            order.getLane().changeCellType(order.getAreaHead(),order.getArea(),1)
            secondLit ← reservOrderList.listIterator(mainLit.nextIndex())
            count ← 1
            while secondLit.hasNext() do
                if secondLit.next().getVehicle() = order.getVehicle() then
                    count ← 0
                    break;
                end if
            end while
            if count = 0 then
                order.getVehicle().laneReservedContinue ← True
            end if
            if count = 1 then
                order.getVehicle().laneReservedContinue ← False
            end if
            if not order.getVehicle().laneReservedContinue then
                order.getVehicle().laneReserved ← False
                order.getVehicle().lanePreferred ← -1
            end if
            mainLit.remove()
        end if
    end while
end function
```

Finally there is the recoverLane method (Algorithm 6), which is crucial in dynamicity management. Its main task is to change the reserved zones back to their normal states when a priority vehicle leaves the zone so that other vehicles can use the lane. Therefore we use a main list iterator, which is a built-in functionality of Java List containers, in order to traverse the order list and verify whether the subject has left the reserved area. Then, the road thread calls the *changeCellType* method to do the job. If the reservation continues, as is mentioned above, we need to make sure that in the remaining elements in the order list, there is at least one reservation for the same vehicle. Due to the fact that a Java List iterator cannot complete the full iteration when the total number of elements in the list changes during the loop (new order added or expired order removed), we introduced a secondary list iterator which is initialized exactly from the next element of the main iterator's current value, to verify whether the reservation continues. If this is not the case, the lane reservation state and the lane preference of the concerned vehicle would be set to default, and the reservation order would be removed from the list.

4.2.3.4 Clock Thread

Algorithm 7 *Run*() of clock thread

```
Require: listVehicle l
Create output file roadState.txt
Create header
road ← l[0].road
for each time step do
    for vehicle in l do
        vehicle.top(current time)
        road.detectAndReserve(vehicle)
    end for
    road.recoverLane()
    roadState.txt ← time, all cell codes
end for
```

The clock thread plays a role of commander in the whole core simulator. It controls the total duration of the simulation, along with the main thread apparently. For each time step, it transmits the current time to the vehicles as a time stamp so that the vehicles are allowed to take actions. In the vehicle list loop, the clock thread also asks the road thread to detect priority vehicles. At the end of each time step, the road thread is asked to clean up the expired reservations, and the information of all cell types encoded is written to the output file buffer.

4.3 Technology Validation via a Mock-up

The digital simulator can provide us with much useful information while designing and developing the system, since it can easily generate different scenarios for comprehensive consideration. However, what we have simulated was all related to software aspects such as dynamicity and allocation management algorithms, virtualization of vehicles' behaviors *etc.*, which will probably work as expected provided that the technologies they are based on are reliable. In other words, we need to conduct a technological or "Hardware" validation to complete the study, which is also a necessary step before in-the-field deployment. Therefore, we produced a preliminary mock-up with a simple scenario to validate the technologies.

4.3.1 Scenario Description

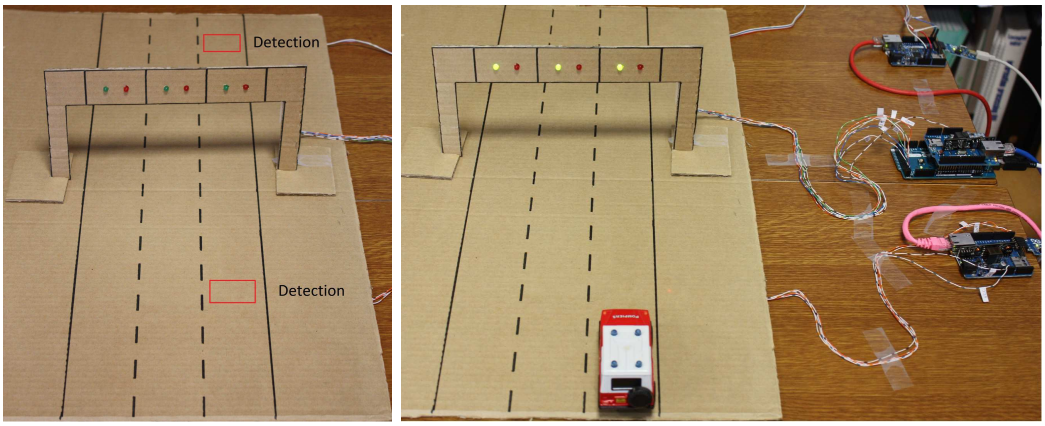


Figure 4.11: Mock-up overview

In the scenario, a priority vehicle is detected, and then a section of lane is reserved and is closed to other vehicles during the presence of this vehicle. As shown in Figure 4.11, three lanes are painted on the cardboard: the potentially reserved section is situated on the right-hand lane, and a display panel is placed over the cardboard as horizontal signaling. Priority vehicles are detected both at the start and at the end of the section: when a priority vehicle is detected at the entrance, the display panel will show that this section of lane is temporarily closed to other vehicles (red LED on), and when the vehicle is detected at the exit, the

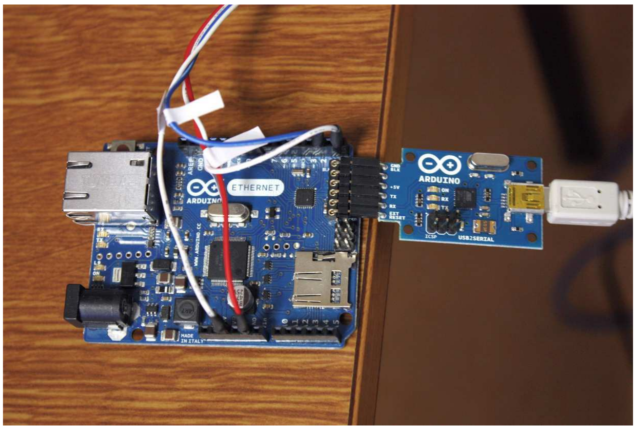


Figure 4.12: Arduino card

display panel will change the signaling, showing that this section of lane is now open to all vehicles (green LED on).

Although this appears to be practical work on digital circuit courses with several electronic components, in this dissertation our aim is rather to attempt to integrate individual objects into a unified framework via network infrastructure with IoT visions. The vehicle, the detection sensors and the display panel are connected and can communicate with each other, thanks to the TCP/IP network, but not via the traditional method that everything connected to a centralized microprocessor.

4.3.2 Hardware Integration and API

4.3.2.1 Hardware

The main integration tool that we used is Arduino, which is a development platform for electronic projects (Figure 4.12), and has been widely used as a connected-devices prototyping tool ([Hodges *et al.* 2013]). There are several Arduino cards models, the size of which is similar to a credit card. Although these cards possess different characteristics, they have some interfaces in common, such as a set of analog/digital input/output pins and several types of data bus (SPI, 12C AND RS232), controlled by a programmable microcontroller. As these interfaces allow a wide range of electronic components to be run, we thus use three Arduino cards to integrate two detection sensors and one LED display panel. The detection sensor is actually simplified as a RFID reader stuck on the backside of the cardboard, and a RFID tag is attached to the vehicle, as shown in Figure 4.13. The reader that we used here has a function area of 10cm in diameter, while a more powerful solution is needed for in-the-field deployment.

Figure 4.13: RFID reader and vehicle with RFID tag

With the addition of an Ethernet Shield, the data processed can be transmitted via a TCP/IP network. To facilitate management of the Arduino cards, we also implement a server PC which plays the same role of a regulation PC in our system architecture. The server can be administrated done remotely via web interfaces.

4.3.2.2 Component Management and Resource Representation

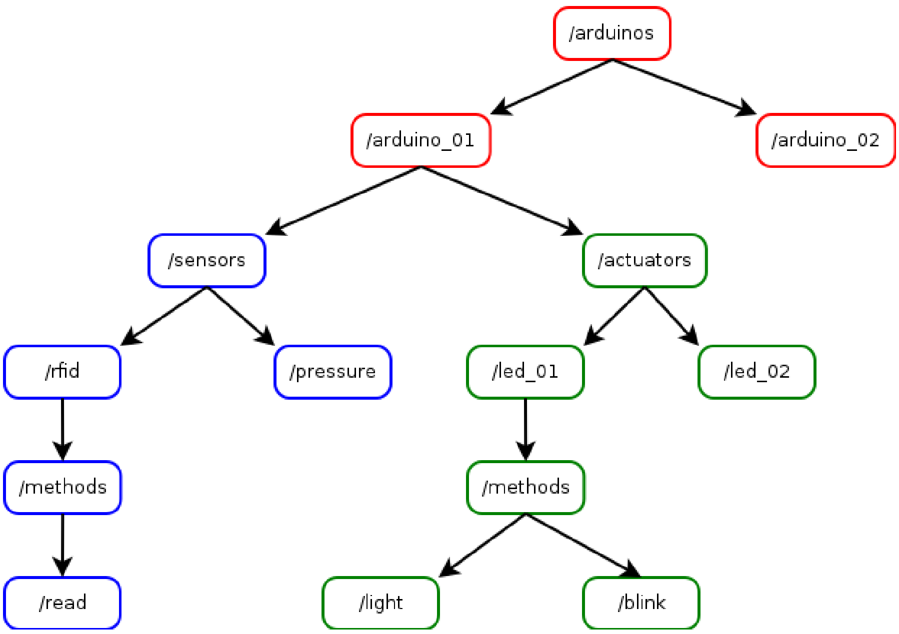


Figure 4.14: Collection of the components

We organised the objects as collections with hierarchy as shown in Figure 4.14. The collection is identified by the URI `/arduinos`, and the server PC can access and manage all the Arduino cards and the objects attached to them. Each Arduino node contains the resources *sensor* and *actuator*. For a sensor or actuator instance, methods are proposed by the object that can be used to interact or communicate with it.

Table 4.4: URI methods of component interfaces

HTTP Method	**URI**	**API**
GET	*/Arduinos*	getArduinos()
GET	*/Arduinos/nameArduino*	getArduino()
GET	*/Arduinos/nameArduino/sensors* */Arduinos/nameArduino/actuators*	getComponents()
GET	*/Arduinos/nameArduino/sensors/nameSensor* */Arduinos/nameArduino/actuators/nameActutator*	getComponent()
GET	*/Arduinos/nameArduino/sensors/nameSensor/methods* */Arduinos/nameArduino/actuators/nameActutator/methods*	getMethods()
GET	*/Arduinos/nameArduino/sensors/nameSensor/methods/nameMethod* */Arduinos/nameArduino/actuators/nameActutator/methods/nameMethod*	getMethod() executeMethod() getData() sendData() compute() sendResult()
PUT	*/Arduinos/nameArduino/sensors/nameSensor/methods/nameMethod* */Arduinos/nameArduino/actuators/nameActutator/methods/nameMethod*	updateMethod() executeMethod() getData() sendData() compute() sendResult()

Table 4.4 describes in detail the URIs and related interfaces for manipulating the objects of an Arduino node. GET methods usually apply to retrieval of object values, while PUT methods are generally used to modify operation properties of an object.

```
Actuator:
Content:
    name = led_01
    subtype = LED
    methods = [blink,light]
    links:
        method = GET
        href = /Arduinos/Arduino_01/led_01/methods
        rel = List actuator method
```

Figure 4.15: Led_01 resource representation

Now that all objects and their methods are considered as resources identifiable via URIs in a REST architecture, we designed the contents of resources as shown in Figure 4.15, 4.16 and 4.17 (LED examples).

In Figure 4.15, we can identify that the object *led_01* belongs to the *actuator* type and is attached to the *arduino_01* node. It executes the request `GET /Arduinos/Arduino_01/actuators/led_01` so that we can retrieve this piece of information.

In Figure 4.16, collection of methods of the object *led_01* is included. Each method can be used in two ways, as well as having different parameters. The link with `GET` would return the status of this method, and the status information contains the on/off value of this LED. The link with `PUT` would modify the status of the method.

Figure 4.17 is the resource representation of the light method. It informs us that the LED is off and shows us how to use the `PUT` method. Actually, the parameters will be added to the `HTTP` request, taking the form “name=value”.

4.3.2.3 Server-Client Communication

Theoretically, the REST architecture can be implemented on the Arduino card. However, the memory of the Arduino Ethernet shield is limited to 2KB, which is not enough to create the variables for sending or receiving the request strings. We also tried to use the memory of the Arduino card itself, which is 32KB, as long as there is some space left apart from the main program. However, the difficulty here

```
Methods:
Content:
    name = light
    links:
        method = GET
        href = /arduinos/arduino_01/led_01/methods/light
        rel = Method Status

        method = PUT
        href = /arduinos/arduino_01/led_01/methods/light
        rel = Switch On/Off LED
        parameters:
            name = action
            value = on|off

    name = blink
    links:
        method = GET
        href = /arduinos/arduino_01/led_01/methods/blink
        rel = Method Status

        method = PUT
        href = /arduinos/arduino_01/led_01/methods/blink
        rel = Blink in seconds LED
        parameters:
            name = interval
            value = integer
```

Figure 4.16: Led_01 method representation

```
Method:
    links:
        method = PUT
        href = /Arduinos/Arduino_01/led_01/methods/light
        rel = Switch On/Off LED
        parameters:
            name = action
            value = on|off
Content:
    name = light
    status = off
```

Figure 4.17: Light method representation

is that we can only access the variables statically and, moreover, the text strings are read "character by character", which makes processing very slow.

Consequently, we use a server PC as a back-end, equipped with `Node.JS`, which is a server-oriented platform for web applications, using the popular JavaScript language. Thus the Arduino cards would work as HTTP clients while communicating with the server.

Figure 4.18 shows an example of Asynchronous JavasScript and XML (AJAX) function for interacting with the LED from the server side. The LED named *greenLed1* would be switched on with its *light* method. Then, we could create two functions: *reserveLane3()* and *freeLane3()*. The former contains two AJAX functions that switch off the green LED and then switch on the red LED of lane 3, while the latter contains the same functions but reversing the on/off order.

```
/*Ajax function using parameters:
  - url: URI of the resource
  - type: HTTP verb
  - data: parameter of the request body
  - async: asynchronous or not */
$.ajax({
    url: "http://" + ip_server +
        "/api/adruinos/arduinoled/actuators/greenLed1/methods/light",
    data: {action: 'on'},
    async: false,
    type: 'PUT',
}).fail(function(error){
    alert(error.message);
});
```

Figure 4.18: Example of an AJAX function for switching on a LED

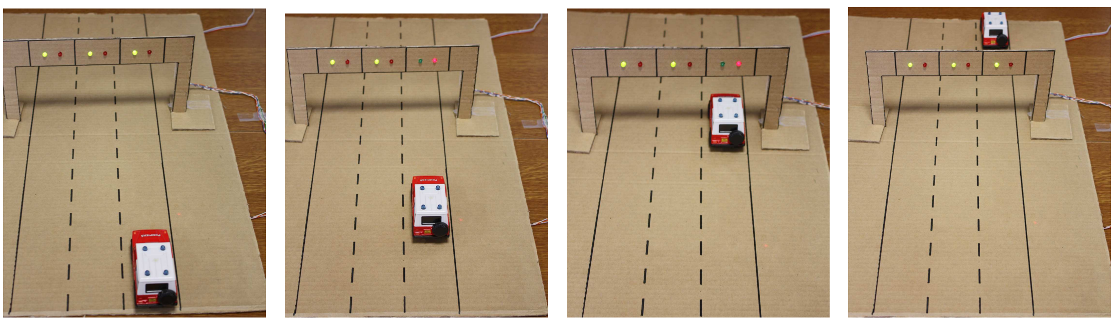
As mentioned above, a priority vehicle is detected by using RFID scanning technology: the vehicle tag is registered, and we need to retrieve the tag information when the vehicle runs over the reader in real-time and to verify the tag. Figure 4.19 shows part of the codes that set up a *arduinoPooling* event listener, which triggers message processing when receiving read information from a tag.

```
// Connection with the server
var socket = io.connect('ip_server');
// Listening of arduinoPooling event
socket.on('arduinoPooling', function(data){
    if (isScenarioStarted){
        // Received from arduino rfid 1
        if(data.return.value == '0CE985E' && data.arduino == 'arduinorfid1'){
            reserveLane3();
        }
        // Received from arduino rfid 2
        if(data.return.value == '0CE985E' && data.arduino == 'arduinorfid2'){
            freeLane3();
        }
    }
});
```

Figure 4.19: RFID event listener

4.3.3 Scenario Execution

We can observe the results from some snapshots in Figure 4.20. On startup, the socket connection between the server PC and all the Arduino cards is established, all the LEDs are initialized (green on, red off), and the priority vehicle is ready to advance. Once the vehicle is detected at the entrance, the red LED of lane 3 is switched on, meaning that this section is reserved and closed to other vehicles. When it reaches the end of the section, the red LED is switched off and the green LED is switched on.



(a) Initial state (b) Detection at entrance (c) Middle of the section (d) Detection at end

Figure 4.20: Scenario execution snapshots

4.4 Summary

In this chapter, we described in detail the core system simulator by looking into the modeling of traffic, modeling of priority and dynamicity, and by making comparisons with more realistic models and thus some optimization. We then presented the implementation of the simulator by specifying algorithms for each thread. Finally, we introduced a Mock-up with a simple scenario as a technology validation, in order to demonstrate the feasibility of using the REST architecture to integrate objects and assure functionality.

Chapter 5

Visualization and HCI Analysis and Tests

In this chapter, we present another important part of the simulation environment – visualization of simulation results. We also introduce a dynamic evaluation tool based on the Unity 3D engine, which takes into consideration the dynamic situation and surrounding circumstances, and allows for real-time user interaction during the tests ([Wang *et al.* 2016]). When transportation users become users of interactive systems, special attention should be paid with regard to principles, knowledge, methods and models. We suggest further reading of [Kolski 2011] for more general and complete contents of "HCI in Transport".

5.1 Visualization Tool

Besides the core system simulator, which focuses on quantitative models and communication among elements, visualization of the simulation results is also an important part of the simulation environment. We present here several developed forms.

Command-line symbolic print

As shown in Figure 5.1, during the early development phase, we pipe the state of the traffic (position of vehicles and lane status) to the standard output, with a time stamp. We use different symbols to mark the lanes: "><" means a closed cell, "[ ]" represents an available cell and "<>" a reserved cell. For those cells occupied by vehicles, we replace the markers with their names (V1,V2,V3,B6, *etc.*). Taking a rather crude form, this view does not seem to be user friendly. However, it helped us a lot in debugging and refining the simulator codes, because it is simple and straightforward. Just imagine that the simulator is running on a cloud-based platform where we have no access to graphical interfaces: operation and maintenance could only be performed via a command-line shell. The symbolic print form would suit this kind of situation.

```
t=14
 >< >< >< >< [] [] [] [] [] V5 [] [] [] [] [] [] [] >< >< >< >< >< >< >< >< >< >< >< [] []
 [] [] [] [] V9 [] <> B7 <> <> <> <> <> <> [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] []
 [] V11 V10 [] V8 [] [] V6 V4 V3 [] V2 [] V1 [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [
t=15
 >< >< >< >< [] [] [] [] [] [] [] V5 [] [] [] [] [] >< >< >< >< >< >< >< >< >< >< >< [] []
 [] [] V11 [] [] V9 <> <> <> B7 <> <> <> <> [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] []
 [] [] [] V10 [] V8 [] V6 V4 [] V3 [] V2 [] V1 [] [] [] [] [] [] [] [] [] [] [] [] [] []
t=16
 >< >< >< >< [] [] V9 [] [] [] [] [] [] V5 [] [] [] >< >< >< >< >< >< >< >< >< >< >< [] []
 [] [] [] [] [] [] <> <> <> <> <> B7 <> <> [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] []
 [] [] [] V11 V10 [] V8 V6 [] V4 [] V3 [] V2 [] V1 [] [] [] [] [] [] [] [] [] [] [] [] [] [
t=17
 >< >< >< >< [] [] [] [] V9 [] [] [] [] [] [] [] [] >< >< >< >< >< >< >< >< >< >< >< [] []
 [] [] [] [] [] [] <> <> <> <> <> <> <> B7 V5 [] [] [] [] [] [] [] [] [] [] [] [] [] [] []
 [] [] [] [] V11 V10 [] V8 V6 [] V4 [] V3 [] V2 [] V1 [] [] [] [] [] [] [] [] [] [] [] [] [
t=18
 >< >< >< >< [] [] [] [] [] [] V9 [] [] [] [] [] [] >< >< >< >< >< >< >< >< >< >< >< [] []
 [] [] [] [] [] [] <> <> <> <> <> <> <> B7 [] V5 [] [] [] [] [] [] [] [] [] [] [] [] [] []
 [] [] [] [] [] V11 V10 [] V8 V6 [] V4 [] V3 [] V2 [] V1 [] [] [] [] [] [] [] [] [] [] [] [
t=19
 >< >< >< >< [] [] [] [] [] [] [] [] V9 [] [] [] [] >< >< >< >< >< >< >< >< >< >< >< [] []
 [] [] [] [] [] [] <> <> <> <> <> <> <> <> B7 [] V2 [] [] [] [] [] [] [] [] [] [] [] [] []
 [] [] [] [] [] V11 [] V10 [] V8 V6 [] V4 [] V3 [] V5 [] V1 [] [] [] [] [] [] [] [] [] [] [
```

Figure 5.1: Command-line symbolic print as a form of visualization

Graphical symbolic snapshot

In order to make the visualization form more graphical, we upgraded the command-line symbolic print: colors are used to mark different cells of a lane, and thumbnails of normal vehicles and buses are drawn in the scene, of course no longer in a command-line shell (Figure 5.2). The scene is actually based on a panel component of Java Swing Library, and we add a scene-update function which redraws all the elements (vehicles with new positions, lanes with new states) in a main loop, so that every 0.2 seconds the whole scene changes. Therefore, what we can observe is the "movements" of the vehicles. The main advantage compared to its predecessor is that we do not have to look up and down to compare two time stamp states, trying to figure out whether a vehicle has advanced a cell.

We call it a "snapshot" because the movements of vehicles are not continuous. So the perception would be something like listening to a piece of music with regular pauses in the middle all the time. Another shortcoming is that, when vehicles are

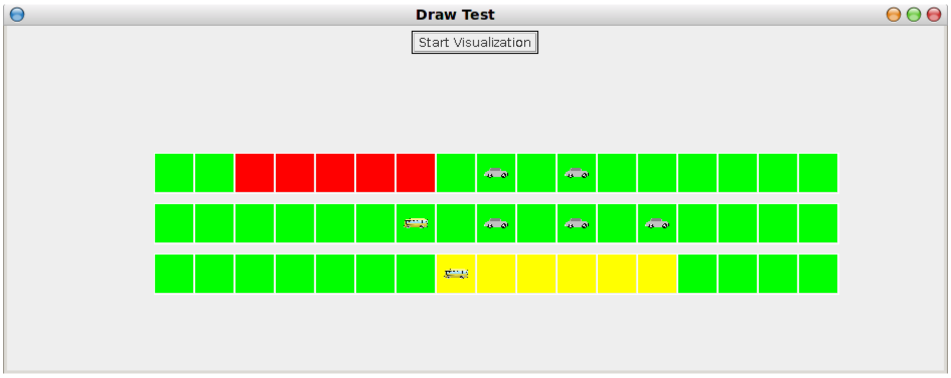


Figure 5.2: Graphical symbolic snapshot as a form of visualization

close to each other, it is difficult to identify each one's path, because all the vehicles "jump" at the same time. Therefore, we cannot tell exactly which vehicle goes to which cell.

Graphical symbolic animation

What we really want is animated visualization. Animation, by definition, is the process of creating the "illusion of motion and change" by means of the rapid display of a sequence of static images that minimally differ from each other (*i.e.* traditional animated cartoons). So the real deficiency of the graphical symbolic snapshot that prevents it from giving the sense of animation, is that two neighboring snapshots, or frames if speaking of animation, differ slightly more. We thus turned to another solution based on JavaFX Animation Library. The main idea is to use the built-in timeline tool to manage animation play, while we just add key frames to the timeline. Interpolation will be carried out to generate all the frames in between. The software is developed using the MVC (Model-View-Controller) approach.

As shown in Figure 5.3, normal vehicles are represented by black rectangles, and buses by green rectangles, while the brown section is an allocated bus zone. All the components of JavaFX Library look more attractive. We also implemented some control buttons such as Start, Pause, Step By Step, Play From Start, to simplify observation and analysis. Thus, we could finally visualize both the individual paths and the collective dynamics of the vehicles in the simulation.

3D in-vehicle viewer

We also developed a 3D visualization tool that is more immersive, using the

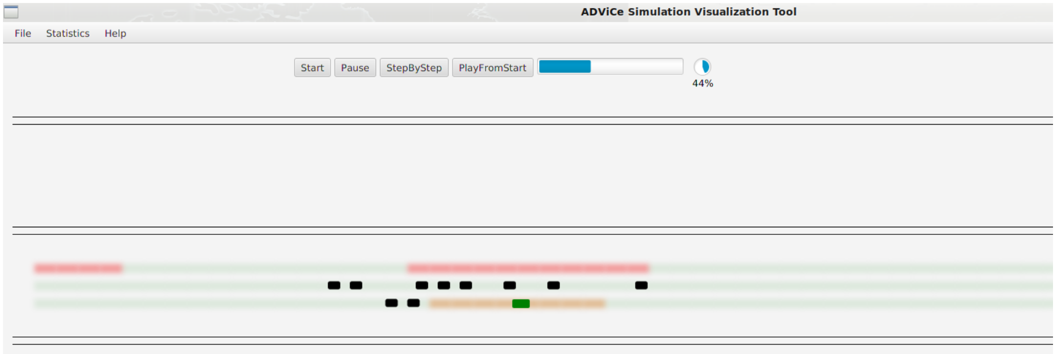


Figure 5.3: Graphical symbolic animation as a form of visualization

Unity 3D engine. Figure 5.4 shows the view from the interior of the vehicle – a driver's first person perspective. We can see other vehicles, the signaling, as well as the surrounding landscape just as through the windscreen of a real motor vehicle. What's more, the rearview mirror function is also taken into consideration. The viewer provides the possibility for users to interact with the simulation, in the place of a driver, which would be more meaningful than just watching the 2D graphical symbolic animation.

Figure 5.4: 3D in-vehicle view

5.2 User Tests

It is necessary to conduct ergonomic studies on the acceptability of this kind of approach to dynamic lane management. This is because there is a possibility that people think it is too complicated: we can then identify the problems and try to solve them. According to our experience, while it is generally not hard for people to understand the basic idea of this approach, difficulties may arise in the specific scenario. Some new road signs have been proposed as variable message signs (some of them may contain symbols already existing in the highway code) that go along with the allocation process, mainly to deliver information. We are concerned with the understanding and acceptability by potential users (drivers) of this kind of signaling, especially in complex situations.

5.2.1 Motivation

Several levels of tests can be produced, using different view forms. Figure 5.5 shows one type: a static photo of a driving scene is shown to drivers requesting them to answer questions on the observed situation. This type of test has been seriously conducted on 187 subjects by the authors of [Hugot *et al.* 2015] from LESCOT/IFSTTAR using on-line questionnaires, and some conclusions were drawn from analysis of results. However, we argue that it does not seem sufficient to use only static-photo tests because users can take 2 seconds or 5 minutes to put forward an answer while, in a real driving situation, we may not have much time to think and react. So we have considered "dynamic" tests in the sense that the scenario takes into consideration the speed of the vehicle, surrounding circumstances through rearview mirrors, signaling through the windscreen, *etc.*

The goal is to go beyond passive viewing by offering a way to test the acceptability of the dynamic approach by the vehicle driver, allowing driver involvement. So we added some interactivity to the 3D in-vehicle viewer to develop an evaluation tool, enabling users to express their perception of traffic either passively (comment on the validity or not of observed driver behavior) or actively (controlling the vehicle in which the driver is located). Theoretically, these possibilities of COMMENT

Do you have the right to go straight?	Yes	No
Do you have to change to the right lane?	Yes	No
Do you have the right to keep the same speed?	Yes	No
Do you have to reduce the speed?	Yes	No

Figure 5.5: An example of a static photo-based test

or COMMAND of situation progress could allow us to access to understanding of the signaling proposed in connection with the required response times. From this point of view, our work is not to repeat the full test as in the study conducted by [Hugot *et al.* 2015] with the new tool, but aims at validating the evaluation tool, to see whether we can obtain more information related to tests from the interaction between users and the tool itself. The tests conducted were preliminary.

5.2.2 Experimentation Configuration

Several test cases can be planned with different traffic speeds ($30km/h$, $60km/h$, $90km/h$) and different visibilities (day, night, rain, snow, fog, *etc.*). The driver observes a driving scenario either as an observer of a particular vehicle or as a driver of a vehicle. In the first case, the tester tries to evaluate the situation and can comment on perceived behavior as appropriate or inappropriate, while in the second case the tester chooses (command) actions applied to the vehicle.

With in mind the goal of validating the evaluation tool itself, we have not carried out a strict sample selection as in formal tests. Instead, for convenience, we have chosen students and researchers in the university as testers. Their ages vary from 19 to 58 and are not uniformly distributed. Also, not all testers have driving experience or a driving license.

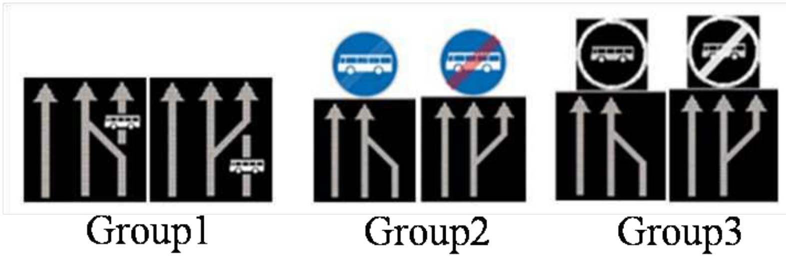


Figure 5.6: Three groups of evaluated road signs (pair of signs: start / end of lane reserved for priority vehicles, mainly buses in this case)

The first series of experiments took place in the context of the Ecole Centrale campus, using three groups of road signs (Figure 5.6), where each group had two different types of signaling (road signs only, or road signs plus marking). Therefore, there were six possible test types and each subject was expected to complete only one out of the six. The sample group was composed of 20 subjects, 15/20 men, 5/20 women; 7/20 of them are French; 8/20 of them have no driving experience or a driving license; 18/20 are PhD students at the Ecole Centrale de Lyon.

The second series of experiments used only the Group 1 signs, and the type of signaling was “road signs plus marking”. However, two different speeds were applied: $30km/h$ and $60km/h$. We invited another 20 subjects (different from the testers in the first series of experiments), with 14/20 men, 6/20 women; 18/20 of them with experience or a driving license; 8/20 of them are PhD students at the Ecole Centrale de Lyon.

5.2.3 Experimentation Protocol

The experimentation protocol was defined as follows:

Step 1 A demonstration phase of the operation during which the subject (potential driver) observes the unfolding of a scenario. The tester looks at the 3D driving animation and tries to interpret the sequence of situations in order to understand overall behavior of the system. The tester then comes to a first impression as to how to deal with these new situations.

Step 2 The tester comments on the behavior of the vehicle in which s/he is. In this case, s/he is observing the actor of a driving session, and comments on perceived behaviors: “I think the operation is allowed”, “I think the move is inappropriate”. To evaluate his/her implication, s/he is credited by an initial account *-2*. If s/he finds the only dissimilated error, s/he gets *+2*; if s/he makes a mistake (a comment incompatible with the right behavior), s/he gets *-1* per fault. So, the ideal behavior in this case is to get 0 points.

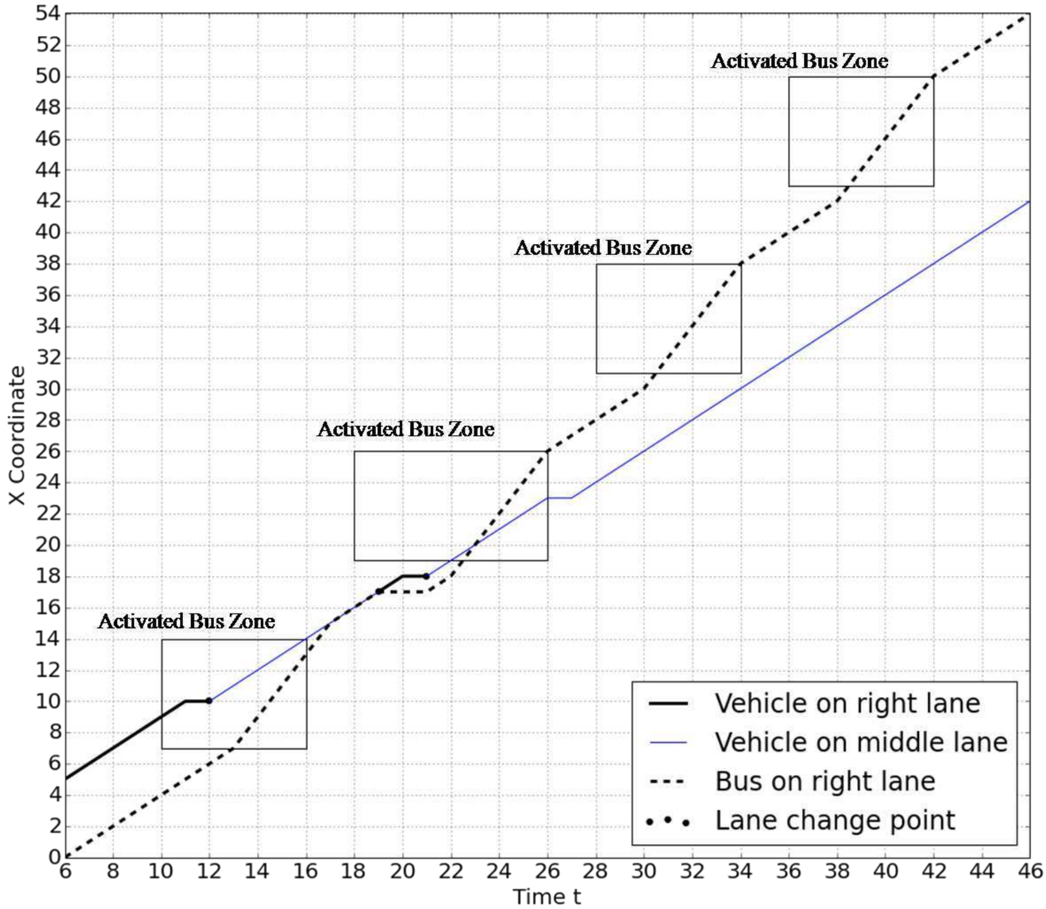
Step 3 The tester commands (controls) the vehicle in which s/he is located. In this step s/he has to choose out of four actions: Forward (F), Slow Down (S), Go Left (L), Go Right (R). Since vehicle behaviors are determined by the simulation scenario, only the COMMANDs are recorded, but the vehicle's behavior does not change. Here we count the number of good actions.

Step 4 The tester fills in a questionnaire (Appendix A) and carries out a free debriefing. In it the user specifies whether s/he has a driving license and for how long, and whether s/he has any driving experience. The tester also comments on his/her understanding of the observed signs and notes them on a Likert 5-point scale. In the debriefing, s/he expresses his/her feelings in relation to this test.

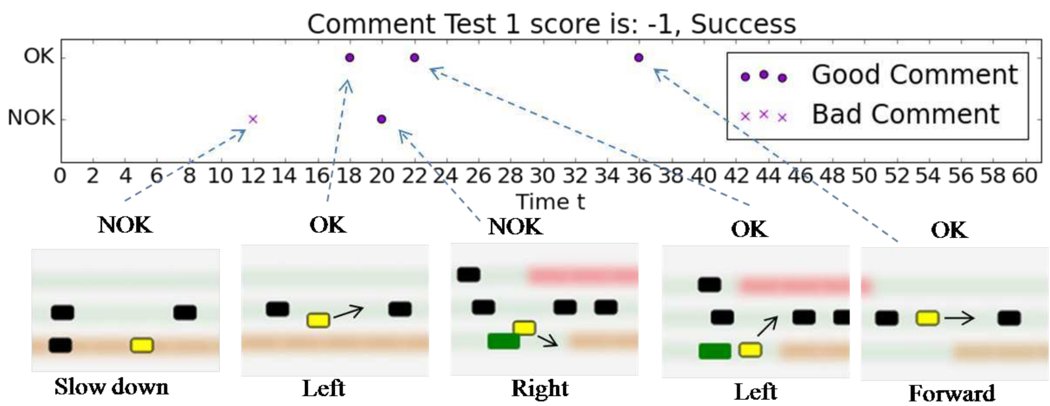
5.2.4 Analysis of Results

Due to the small sample group and only a few representative populations, we treat the current results with caution (part of the original raw data is listed in Appendix B). This means that we do not draw formal strong conclusions in regard to the understanding and acceptability of those road signs, but try to validate the performance of the evaluation tool. We firstly examine several representations, which aim to show the individual behavior of each subject (potential driver). Then, we show some examples of possible conclusions if formal tests are carried out with enough carefully selected subjects in the future.

One of the advantages of the dynamic test is the time dimension. Figure 5.7 (a) represents the Time-Space diagram of the bus and the vehicle in which the subject is situated (other vehicles are not shown for increased clarity). The lane position of



(a) Time - Space diagram



(b) Comment results coupled with simulation senarios

Figure 5.7: Analysis of individual results for COMMENT

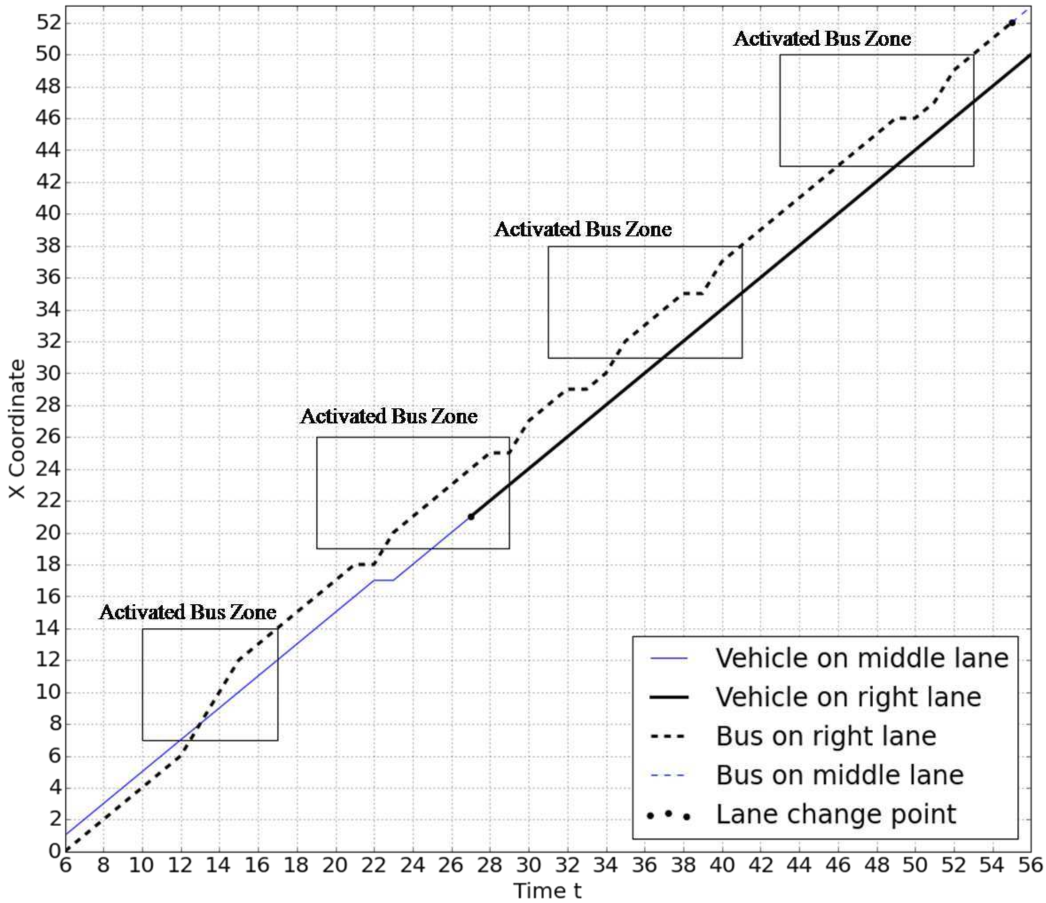
the bus is represented by a dotted line, while the position of the subject vehicle is represented by a continuous line; if the vehicles are in the right-hand lane, the line will be thicker and black, while they will be thinner and blue when in the middle lane; the length and activation time of the bus lane is represented by a rectangle. We can see the utility of bus lane activation when the bus is overtaking the subject vehicle at time 23.

During this COMMENT session, misbehavior of the subject vehicle was introduced at time 19-20: being aware of the reserved section in front, the subject vehicle still chose to change to the right-hand lane and blocked the bus, which is a really disturbing behavior. We then plot the tester's answers in Figure 5.7 (b) on the y-axis (appropriate - OK, inappropriate - NOK) and if the subject was right (bullet) or wrong (cross). For each answer, we also associate the detailed views (coming from the graphical symbol animation bird eye view, the subject vehicle is represented by a yellow rectangle, while the bus is represented by a green rectangle) of the context on the road, which allow us to better understand each answer. Here we can see from the figure that the tester put a "NOK" comment at time 20, which we considered as a good comment, as the tester has found the dissimilated error. What's more, the tester provided the comment just in time, without hesitation. It is possible that a tester could give a good but late comment.

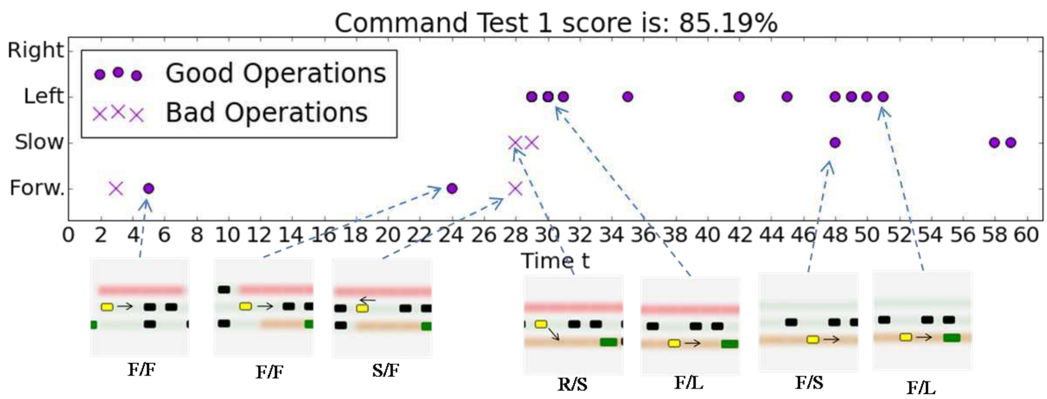
In the case of the COMMAND session, the representations are similar (Figure 5.8). The subject can give four possible answers (Forward, Slow Down, Left, Right), which are plotted and analyzed (correct - bullet, wrong - cross). Some detailed views (but not all) of the context on the road at the same instants are also shown and completed by a pair of letters like A/B, where A is the expected action from the simulation and B the subject's answer. In some cases, the subject's answer can be right even if it differs from the expected action because there are several valid actions.

To summarize (Table 5.1), the main global results are:

- Understanding Column: the note of understanding of the sign is derived from the questionnaire based on a Likert scale of 1 to 5.



(a) Time - Space diagram



(b) Command results coupled with simulation senarios

Figure 5.8: Analysis of individual results for COMMAND

- Column Comment Success Rate: indicates the percentage of testers able to find the dissimilated error.

- Column Comments Point: gives the average mark commenting on the driving course. The closer to 0, the better it is.

- Column Good Command: percentage of good actions for the command test.

One example conclusion could be that road signs plus marking configuration give a higher commenting success rate for most groups of signs. Arguably, road marking improves outcomes, which is also true for the COMMAND test.

Although the notes to the sign groups 2 and 3 are better appreciated in the tests than the sample group 1, according to the survey (questionnaire), the signs of group 1 (Bus Inlaid) are more comprehensible and clearer (Figure 5.9). So here we can clearly see the limitation of an overly small non-representative sample population in preliminary tests, which leads to conflicting assertions: there are only 3∼4 subjects for each combination of road signs and signaling type. With more and better selected subjects, combined with the questionnaire results, we can try to figure out whether age, having or not a driving experience, education level, profession, nationality, *etc.* have an effect on the understanding and acceptability of the road signs, and what is the correlation of these elements.

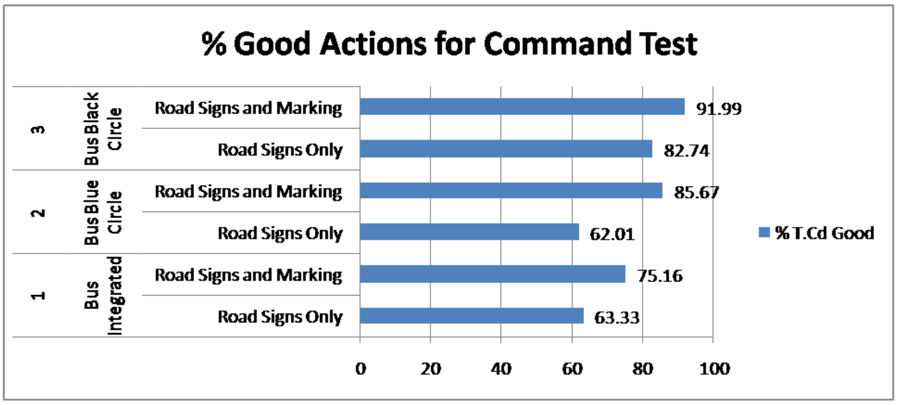


Figure 5.9: Results of the three sample groups tested in the COMMAND test

Table 5.2 shows similar information but for two different speed settings. We find that subjects carrying out tests more slowly had a better understanding of the

Table 5.1: Synthesis of global results

Group	Type	Type of Signaling	Under-standing	Comment Success Rate	Comments Point	Good Commands
1	Bus Inlaid	Road Signs Only	4.5/5	3/3	−0.6	63.33%
		Road Signs and Marking	4/5	2/3	−1.3	75.16%
2	Bus Blue Circle	Road Signs Only	3.8/5	2/4	−1.5	62.01%
		Road Signs and Marking	1.6/5	3/4	−1.75	85.67%
3	Bus Black Circle	Road Signs Only	4.2/5	3/3	0	82.74%
		Road Signs and Marking	4.3/5	3/3	−1	91.99%

Table 5.2: Synthesis of global results for tests with different speeds

Speed	Under-standing	Comment Success Rate	Comments Point	Good Commands
$v = 30km/h$	4.5/5	8/8	−0.88	76.70%
$v = 60km/h$	4.2/5	6/12	−3.25	55.82%

road signs and better comment and command results. It is reasonable that, at a lower speed, we have more time to observe and react, while if we drive fast it will be easier to make mistakes.

It is important to mention that, with the first preliminary results and the feedback from the testers, we have already improved the evaluation tool: for example, visibility of road signs is related to simulation speed, optimization of transformation from an agent-based vehicle to a steerable vehicle in the COMMAND test. We also managed to distinguish situations such that a tester who fully understood the meaning of the road signs but failed the test due to late answers or actions, and that a tester who passed the entire test but actually totally misunderstood the meanings of the road signs.

5.3 Summary

In this chapter, we showed different forms of visualization of simulation results that we developed, from the command-line symbolic print to the graphical symbolic animation and 3D in-vehicle viewer. We presented our dynamic evaluation tool, which takes into account the situation and surrounding circumstances as well as vehicle speed, and we also described the user tests in which users can not only express their perception but also interact with the tool in real-time. Preliminary analyses of results were presented.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This dissertation, prepared in the context of our international France-China research project on Smart City in the laboratory, mainly focuses on Location-Based Services and Internet of Things. After conducting a literature review in the areas related to our topic of research, we believe that the Internet of Things as well as the underlying sensor network are the basis of the sense of smartness that we find in the Smart City. So the goal of the dissertation is to validate the approach of combining IoT and LBS, and to make the latter smarter to meet the requirements of Smart Cities.

For the theoretical aspects, we identified the key characteristics of LBS, especially during the evolution, the challenges need to be handled: heterogeneity of devices, service management, data management and analysis, HCI aspects. The power that the Internet of Things can provide to better support LBS, lies not only in the fact that we can get more data from more devices, but also in that the IoT architecture would act like a King's Hand. Integrating various objects would give LBS better management of location-aware devices, sensors enriching the data source, the IoT middleware, good at object abstraction and service composition, all contribute to deploying more intelligent and customized service components to LBS.

For the practical aspects, we chose a dynamic lane management system as a Smart City application to conduct a case study, as intelligent transportation systems are essential to the urban functioning and prosperity of Smart Cities. We used the approach of IoT enhanced LBS when designing the solution: dynamic allocation depends largely on the position of the priority vehicles and the speed, and allocation strategies could be very complicated. The solution is based on a whole framework of networked sensors and actuators including detectors, signaling, vehicles, embedded interfaces, regulation PC, smart phones, *etc.*, the communication and collaboration between elements were identified, and user interface design was also taken into account.

Since modeling and simulation are often employed as a means of trying to analyze, understand and control complex systems, including transportation systems, we developed a simulation environment to validate the solution design. It contains a core simulator, which simulates the system functioning and vehicle behavior, a scenario editor and a traffic generator as initialization tools. Different forms of visualization of the simulation results were also developed, from simpler ones to richer and more complicated ones, demonstrating our footprints on the beach of research work.

A proof of concept of the dynamic allocation scenario, which involves collection of data, integration of objects, transmission of information, was produced in addition to the simulation environment to validate the technological aspects. It demonstrated how to actually use the IoT architecture to integrate objects and define the APIs to use them in the service components.

Visualization of the simulation results was used to conduct user acceptability studies, the goal of which is to validate acceptability of new driving situations from the driver's point of view, in the context of dynamic lane allocation. The specialty was that, instead of using static scene photos, we used 3D simulated driving situation animation to conduct the tests and allowed users to interact with the tool. The goal is not to repeat the full tests but to validate the performance of the interactive evaluation tool, which will allow other researchers to conduct formal tests in the future. Based on the rich-context representations and diagrams of the results, we would be able to better understand users' answers. Also, we showed some examples of conclusions that can be drawn if the tests are conducted formally with sufficient and well selected sample groups.

6.2 Future Perspectives

The work of this dissertation is interdisciplinary, and thus leaves us several possibilities for extending it in different directions.

A more powerful simulator

Although we are not specialists in traffic modeling and mathematical theories of traffic flow, nothing could stop us from trying to combine more realistic traffic models with our approach of dynamic lane allocation: part of the work has already been carried out by us based on an open-source project (MovSim: Multi-model Open-source Vehicular traffic SIMulator [Treiber & Kesting 2013]). Real-time visualization of vehicles using property binding was tested with success, but still requires more effort for implementing the communication layer. A more realistic simulator would help considerably in the future before real in-the-field deployment of the system (*i.e.* parameter determination, sensor placement, *etc.*).

With regard to the evaluation tool, which is based on 3D Visualization, it is a pity that we have not leveraged the Unity 3D engine to provide game style interaction during the simulated session for tests, so that, when users COMMAND the vehicle, they do not need to pretend to do so but the vehicle will really obey the users' orders. At the same time, other simulated vehicles would also be influenced and react to the incidents, which sounds more realistic. It is true that we were in the logic of "off-line visualization", which means the results are determined by the simulator. The visualization tool does nothing but show it to the users without modifying the data. However, real-time visualization and interaction would surely offer different user experiences. We also need more representative sample groups to conduct tests in order to draw strong conclusions.

A more complete simulation or/and proof of concept

The proof of concept mainly demonstrated the technological aspect of the IoT. However, in the simulation environment, we did not actually call the APIs of the real objects: those APIs were called in the proof of concept's own software implementation. In other words, the toy vehicle and the infrastructure in the proof of concept were not equipped with all the methods coded in the simulation software, but rather with some typical minimal functionalities that allowed them to work. So, they were separated, and the solution is to make either one more complete.

Long term perspective

It seems important to point out the evolution today of the concept of priority vehicles, initially limited to buses, emergency vehicles, police cars, and fire engines. It now also incorporates cabs and, as in other countries, vehicles with several passengers (HOV: High Occupied Vehicles). Also, cleaner, electric or hybrid energy-economic vehicles can benefit from the allocation state of priority vehicles. This may be an incentive to the generalization of the dynamic lane allocation approach, for the future of Smart Cities.

We can also imagine integrating the approach proposed by this dissertation work into the research on self-driving vehicles [Google Self-Driving Car Project 2014]. Apparently, a self-driving vehicle has to use sensors and embedded systems to sense objects like pedestrians, cyclists, and vehicles based on their size, shape and movement pattern, in order to drive safely. And it will also predict what all the objects might do next. From a user's point of view, a self-driving car is designed for riding instead of driving. It is another example of the IoT-based ubiquitous computing scenario. Indeed, how a self-driving vehicle will behave in dynamic lane allocation situations is an interesting issue: for example, will it receive the notification directly from the information system just like the simulated vehicles in our simulator or use computer vision based methods to "read" from signaling, being aware of the situation like human drivers? Can we also develop a simulator or an artifact for such situations?

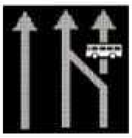
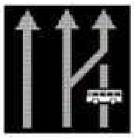
The LBS addressed in this dissertation mainly fall into the outdoor category, especially the dynamic allocation of road lanes case study. However, indoor LBS are also crucial for life in Smart Cities. Perhaps we do not notice that we spend much time indoors, working, shopping, eating, at the office, on campus, at the mall. Indoor positions can be used for navigation ([Mulloni *et al.* 2012]), location sharing, shopping list routing, *etc.*. What we should pay attention to is the seamless handover between outdoor and indoor LBS. While a user could benefit from GPS and mapping services outdoors, when s/he goes indoors, other technologies and services (wearable devices or smart phones) will continue to serve him/her. Consequently, the automatic transition and intelligent sensing managed by a comprehensive platform or framework needs to be developed.

Appendix A : Questionnaire

Post-test Questionnaire

For ADViCe 3D Test 2015

Personal Information					
Name		Gender	☐ Male ☐ Female	Age	
Education Level		Profession			
Do you have driving experience?	☐ Yes How long	☐ No			
Do you have a driver license?	☐ Yes How long	☐ No			

Questions About Understanding The Signs						
No.	Questions	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1	**I understand the meaning of** 					
The meaning is:						
2	**I understand the meaning of** 					
The meaning is:						

Questions About Sensations After Test						
No.	Questions	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1	**I think I did well**					
2	**I am satisfied with what I did**					
3*	**The Signaling band on road (horizontal) helps understand the situation**					

Comments and Remarks:

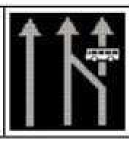
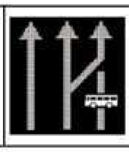
Post-test Questionnaire

Pour ADViCe 3D Test 2015

Informations Personelles							
Nom		Prénom		Sexe	☐ H ☐ F	Âge	
Niveau d'éducation		Profession					

Avez-vous une expérience de conduite?	☐ Oui Combien de temps	☐ Non
Avez-vous un permis de conduire?	☐ Oui Combien de temps	☐ Non

Questions sur la Compréhension des Panneaux

No.	Questions		Fortement en Désaccord	Désaccord	Neutre	D'accord	Tout à fait d'accord
1	**Je comprends ce que veut dire**						
Ce panneau veut dire :							
2	**Je comprends ce que veut dire**						
Ce panneau veut dire :							

Questions sur des Sensations Après le Test

No.	Questions	Fortement en Désaccord	Désaccord	Neutre	D'accord	Tout à fait d'accord
1	**Je pense que j ai bien fait**					
2	**Je suis satisfait(e) de ce que je fesais**					
3*	**Le marquage au sol aide à la compréhension de la situation**					

Remarques:

Appendix B : Raw Test Data

GroupSign	Vertical or Mixte	ctSucc	ctScore	% cdScore	DrivExp / years	DrivLic / years	Names	Understand1	Understand2	Explications	I think I did well	I am satisfied with what I did	I think "marquage au sol" helps
1	M	0	-4	54.05	0	0		3	3	0.5	2	3	3
1	M	1	0	71.43	26	26		5	5	1	5	5	5
1	M	1	0	100	0	0		4	4	1	4	4	5
2	M	1	-1	100	0	0		3	2		3	3	2
2	M	1	-1	85.19	30	30		2	2	1	3	3	5
2	M	0	-3	70	11	9		1	1		3	2	5
2	M	1	-2	87.5	3	0		1	1		3	3	5
3	M	1	-1	87.5	10	10		5	5	1	5	5	5
3	M	1	-1	88.46	22	22		4	4	1	4	4	5
3	M	1	-1	100	7	7		4	4	1	4	4	5
1	V	1	-1	53.33	0	0		3	4	0	4	4	N/A
1	V	1	-1	66.67	1	7		5	5	1	3	5	N/A
1	V	1	0	70	0	0		5	5	1	4	5	N/A
2	V	1	0	66.67	0	0		4	4	1	4	4	N/A
2	V	0	-3	40	0	0		2	2		1	1	N/A
2	V	1	0	75	0	0		3	5	0	3	4	N/A
2	V	0	-3	66.67	3	3		5	5	0.5	5	5	N/A
3	V	1	0	85.71	4	10		5	5	0.5	4	4	N/A
3	V	1	0	62.5	9	8		4	4	1	3	3	N/A
3	V	1	0	100	5	5		4	3	0.5	4	4	N/A

GroupSign	Vertical or Mixte	ctSucc	ctScore	% cdScore	DrivExp / years	DrivLic / years	Names	Speed	Understand1	Understand2	Explications	I think I did well	I am satisfied with what I did	Male/Female
1	M	1	0	76.92	0	1		30	5	4	1	3	3	F
1	M	1	0	86.36	10	10		30	5	5	1	4	4	M
1	M	1	0	72.73	0	0		30	4	4	1	3	3	M
1	M	1	0	84.62	40	40		30	5	4	1	3	3	F
1	M	1	-3	60	1	2		30	3	5	1	5	5	M
1	M	1	-1	73.33	0	0		30	4	4	1	4	4	M
1	M	1	-1	81.82	3	2		30	5	5	1	4	4	M
1	M	1	-2	77.78	3	2		30	5	5	1	4	3	F
1	M	1	-1	57.14	5	5		60	4	4	1	3	4	M
1	M	0	-3	83.33	1	8		60	4	4	1	3	3	F
1	M	1	-2	40	8	6		60	4	4	?	4	3	M
1	M	0	-3	55.56	1	2		60	5	5	1	3	4	M
1	M	0	-5	57.14	2	1		60	4	4	1	3	3	F
1	M	1	-1	58.82	2	2		60	0	5	1	3	3	F
1	M	0	-3	39.13	?	?		60						
1	M	1	-2	61.54	37	37		60	5	5	1	5	5	M
1	M	0	-3	58.33	?	?		60						
1	M	0	-5	55.56	10	20		60	5	5	1	4	4	M
1	M	1	-4	47.06	4	1		60	4	4	1	5	5	M
1	M	1	-7	56.25	1	1		60	5	5	1	5	5	M

Publication

Conference: Chen Wang, Bertin Nayo, Bertrand T. David and René Chalon. *Allocation dynamique des voies de circulation : tests d'interprétation de la signalisation*. In Les Rencontres de la Mobilité Intelligente, Paris, France, Jan 2016

Conference: Chen Wang, Bertrand David and René Chalon. *A Smart City Case Study: Dynamic Management of Road Lanes*. In Panos Markopoulos Norbert Streitz, (Ed.), Second International Conference, DAPI 2014, Held as Part of HCI Interational 2014, Lecture Notes in Computer Science, pages 629–640. Springer, jul 2014. doi: `http://dx.doi.org/10.1007/978-3-319-07788-8_58`

Conference: Chen Wang, Bertrand David and René Chalon. *Dynamic road lane management: A smart city application*. In 3rd International Conference on Advanced Logistics and Transport (ICALT'2014), pages 72–77. IEEE, May 2014. doi: `http://dx.doi.org/10.1109/ICAdLT.2014.6864085`

Conference: Bertrand David, Tao Xu, Huiliang Jin, Yun Zhou, René Chalon, Bingxue Zhang, Chuantao Yin and Chen Wang. *User-oriented System for Smart City approaches*. In Analysis, Design, and Evaluation of Human-Machine Systems, volume 12, pages 333–340. IFAC / Elsevier, 2013. doi: `http://dx.doi.org/10.3182/20130811-5-US-2037.00072`

Journal: Chen Wang, Bertrand David, René Chalon and Chuantao Yin. *Dynamic road lane management study: A Smart City application*. Transportation Research Part E: Logistics and Transportation Review, vol. 89, pages 272–287, 2015. doi: `http://dx.doi.org/10.1016/j.tre.2015.06.003`

Journal: Di Huang, Huaxiong Ding, Chen Wang, Yunhong Wang, Guangpeng Zhang and Liming Chen. *Local circular patterns for multi-modal facial gender and*

ethnicity classification. Image and Vision Computing, vol. 32, no. 12, pages 1181 – 1193, 2014

Bibliography

[Abowd *et al.* 1999] Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith and Pete Steggles. *Towards a better understanding of context and context-awareness.* In Handheld and ubiquitous computing, pages 304–307. Springer, 1999. 25

[Akyildiz *et al.* 2002] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. *Wireless sensor neworks: a survey.* Computer Networks, vol. 38, no. 4, pages 393–422, 2002. 13

[Ambient Intelligence Lab] Ambient Intelligence Lab. `http://www.ami-lab.org`. Consulted on 11/04/2013. 10

[Atzori *et al.* 2010] Luigi Atzori, Antonio Iera and Giacomo Morabito. *The Internet of Things: A survey.* Computer Networks, vol. 54, pages 2787–2805, 2010. 10, 11, 13, 30, 33, 34

[Auto-Id Labs] Auto-Id Labs. `http://www.autoidlabs.org`. Consulted on 11/04/2013. 12

[Aydt *et al.* 2012] Heiko Aydt, Michael Lees and Alois Knoll. *Symbiotic simulation for future electro-mobility transportation systems.* In Proceedings of the Winter Simulation Conference, page 149. Winter Simulation Conference, 2012. 35, 37

[Barnum 2010] Carol M Barnum. *Usability Testing Essentials: Ready, Set, Test!* Elsevier, 2010. 51

[Batty *et al.* 2004] M Batty, N Shiode, C Li, P Longley and D Maguire. *The Impact and Penetration of Location-Based Services.* In H A Karimi and A Hammad, (Editors), Telegeoinformatics: location-based computing and services, pages 349–366. CRC Press, 2004. 7

[Bellavista *et al.* 2008] Paolo Bellavista, Axel Kupper and Sumi Helal. *Location-based services: Back to the future*. Pervasive Computing, IEEE, vol. 7, no. 2, pages 85–89, 2008. 5, 6, 15

[Broll *et al.* 2009] Gregor Broll, Enrico Rukzio, Massimo Paolucci, Matthias Wagner, Albrecht Schmidt and Heinrich Hussmann. *Perci: Pervasive service interaction with the internet of things*. Internet Computing, IEEE, vol. 13, no. 6, pages 74–81, 2009. 34

[Caragliu *et al.* 2011] Andrea Caragliu, Chiara Del Bo and Peter Nijkamp. *Smart Cities in Europe*. Journal of Urban Technology, vol. 18, no. 2, pages 65–82, 2011. 1

[Carey *et al.* 2009] Graham Carey, Thomas Bauer and Karen Giese. *Bus Lane with Intermittent Priority (BLIMP) Concept Simulation Analysis*. Rapport technique, US National Bus Rapid Transit Institute, 2009. 40

[Chen & Cheng 2010] Bo Chen and Harry H. Cheng. *A Review of the Applications of Agent Technology in Traffic and Transportation Systems*. Trans. Intell. Transport. Sys., vol. 11, no. 2, pages 485–497, Jun 2010. 36

[College 2016] King's College. *What are the components of an IoT product*. `http://www.futurelearn.com/courses/internet-of-things`, 2016. Consulted on 02/21/2016. 14

[Compose-Project 2012] Compose-Project. `http://www.compose-project.eu`, 2012. Consulted on 02/21/2016. 31

[Concise-Insight-Ltd. 2003] Concise-Insight-Ltd. *European Location-Based Services: Operator Sttus and Market Drivers*, 2003. London. 7

[Cugola *et al.* 2002] Gianpaolo Cugola, H Jacobsen*et al.* *Using publish/subscribe middleware for mobile systems*. ACM SIGMOBILE Mobile Computing and Communications Review, vol. 6, no. 4, pages 25–33, 2002. 23

[da Rocha 2004] Ricardo Couto Antunes da Rocha. *Middleware for Location-based Services*. Laboratory for Advanced Collaboration, Pontificia Universidade Catolica do Rio de Janeiro, 2004. 25

[David & Chalon 2010] Bertrand David and René Chalon. *Hotspot Based Mobile Web Communication and Cooperation: ABRI+ Bus Shelter as a Hotspot for Mobile Contextual Transportation and Social Collaboration*. In Proceedings of the 2010 International Conference on Semantic Web & Web Services, SWWS 2010, July 12-15, 2010, Las Vegas, Nevada, USA, pages 184–189, 2010. 2, 14, 34

[David *et al.* 2011] Bertrand David, René Chalon and Bernard Favre. Human-Computer Interactions in Transport, chapter *ICT and new human-machine interactions for trucks and buses in the future: e-Truck and e-Bus perspectives*, pages 157–201. ISTE Ltd and John Wiley & Sons, Inc., 2011. 14

[David *et al.* 2012] Bertrand David, Chuantao Yin, Yun Zhou, Tao Xu, Bingxue Zhang, Huiliang Jin and René Chalon. *SMART-CITY: problematics, techniques and case studies*. In Computing Technology and Information Management (ICCM), 2012 8th International Conference on, volume 1, pages 168–174, 2012. 1, 14

[David *et al.* 2013] Bertrand David, Tao Xu, Huiliang Jin, Yun Zhou, René Chalon, Bingxue Zhang, Chuantao Yin and Chen Wang. *User-oriented System for Smart City approaches*. In Analysis, Design, and Evaluation of Human-Machine Systems, volume 12, pages 333–340. IFAC / Elsevier, 2013. doi: http://dx.doi.org/10.3182/20130811-5-US-2037.00072. 14

[De Deugd *et al.* 2006] Scott De Deugd, Randy Carroll, Kevin Kelly, Bill Millett and Jeffrey Ricker. *SODA: service oriented device architecture*. IEEE Pervasive Computing, no. 3, pages 94–96, 2006. 30

[Doniec *et al.* 2008] Arnaud Doniec, René Mandiau, Sylvain Piechowiak and Stéphane Espié. *A behavioral multi-agent model for road traffic simulation*.

Engineering Applications of Artificial Intelligence, vol. 21, no. 8, pages 1443 – 1454, 2008. 54

[Doukas & Antonelli 2014] Charalampos Doukas and Fabio Antonelli. *A full end-to-end platform as a service for smart city applications*. In 2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pages 181–186. IEEE, Oct 2014. 12, 31, 32

[Duquennoy *et al.* 2009] S. Duquennoy, G. Grimaud and J. J. Vandewalle. *The Web of Things: Interconnecting Devices with High Usability and Performance*. In International Conference on Embedded Software and Systems, 2009. ICESS '09., pages 323–330, May 2009. 11

[Fastprk 2012] Fastprk. `http://www.fastprk.com`, 2012. Consulted on 02/21/2016. 32

[Fielding & Taylor 2002] Roy T. Fielding and Richard N. Taylor. *Principled Design of the Modern Web Architecture*. ACM Trans. Internet Technol., vol. 2, no. 2, pages 115–150, May 2002. 11

[Fielding 2000] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software*. Ph.D. thesis, University of California, Irvine, 2000. 28

[Gellersen *et al.* 2009] Hans Gellersen, Carl Fischer, Dominique Guinard, Roswitha Gostner, Gerd Kortuem, Christian Kray, Enrico Rukzio and Sara Streng. *Supporting Device Discovery and Spontaneous Interaction with Spatial References*. Personal Ubiquitous Comput., vol. 13, no. 4, pages 255–264, May 2009. 11

[Gerla *et al.* 2014] Mario Gerla, Eun-Kyu Lee, Giovanni Pau and Uichin Lee. *Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds*. In 2014 IEEE World Forum on Internet of Things (WF-IoT), pages 241–246. IEEE, March 2014. 12

[Gipps 1986] Peter G Gipps. *A model for the structure of lane-changing decisions.* Transportation Research Part B: Methodological, vol. 20, no. 5, pages 403–414, 1986. 64

[Giusto *et al.* 2010] D. Giusto, A. Iera, G. Morabito and L. Atzori. The internet of things. Springer, 2010. 10

[Google Self-Driving Car Project 2014] Google Self-Driving Car Project. `http://www.google.com/selfdrivingcar/how/`, 2014. Consulted on 05/21/2016. 106

[Guériau *et al.* 2015] Maxime Guériau, Romain Billot, Nour-Eddin El Faouzi, Salima Hassas and Frédéric Armetta. *Multi-Agent Dynamic Coupling for Cooperative Vehicles Modeling.* In The Twenty-Ninth Conference on Artificial Intelligence AAAI'2015-(DEMO Track), 2015. ix, 34, 35

[Guinard & Trifa 2009] Dominique Guinard and Vlad Trifa. *Towards the web of things: Web mashups for embedded devices.* In Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences), Madrid, Spain, page 15, 2009. 11

[Guinard 2011] Dominique Guinard. *A Web of Things Application Architecture – Integrating the Real-World into the Web.* Ph.D. thesis, ETH Zurich, 2011. 27, 28, 29, 30

[Guler & Menendez 2013] S. Ilgin Guler and Monica Menendez. *Empirical evaluation of bus and car delays at pre-signals.* In STRC, 13th Swiss Transport Reasearch Conferences, Monte Veritá / Ascona, 2013. 40

[Hidas 2005] Peter Hidas. *Modelling vehicle interactions in microscopic simulation of merging and weaving.* Transportation Research Part C: Emerging Technologies, vol. 13, no. 1, pages 37 – 62, 2005. 64

[Hodges *et al.* 2013] S. Hodges, S. Taylor, N. Villar, J. Scott, D. Bial and P. T. Fischer. *Prototyping Connected Devices for the Internet of Things*. Computer, vol. 46, no. 2, pages 26–34, Feb 2013. 80

[Huang *et al.* 2014] Di Huang, Huaxiong Ding, Chen Wang, Yunhong Wang, Guangpeng Zhang and Liming Chen. *Local circular patterns for multi-modal facial gender and ethnicity classification*. Image and Vision Computing, vol. 32, no. 12, pages 1181 – 1193, 2014.

[Hugot *et al.* 2015] Myriam Hugot, Fabien Moreau and Hélène Tattegrain. *Quelle signalisation pour les allocations dynamiques de voies réservées à la circulation des transports publics?* In ATEC ITS FRANCE, les rencontres de la mobilité intelligente, page 18 p, Paris, France, Jan 2015. 93, 94

[Hui & Culler 2008a] J. W. Hui and D. E. Culler. *Extending IP to low-power, wireless pesonal area networks*. IEEE Internet Computing, vol. 12, no. 4, pages 37–45, 2008. 13

[Hui & Culler 2008b] J. W. Hui and D. E. Culler. *IP is dead, long live IP for wireless sensor networks*. In Proc. 6th ACM Conf. Embedded Nework Sensor Systems, pages 15–28, 2008. 13

[IBM 2008] IBM. *Smarter Planet*. `http://www.ibm.com/smarterplanet/`, 2008. 1

[IPSO alliance] IPSO alliance. *Enabling the internet of things*. `http://www.ipso-alliance.org`. Consulted on 11/05/2016. 13

[Jacobsen 2001] H-Arno Jacobsen. *Middleware services for selective and location-based information dissemination in mobile wireless networks*. In Proceedings of the Workshop on Middleware for Mobile Computing, pages 311–321, 2001. 23

[Jacobsen 2004] Hans-Arno Jacobsen. *Middleware for Location-Based Services.*, 2004. 23

[Jiang & Yao 2006] Bin Jiang and Xiaobai Yao. *Location-based services and GIS in perspective.* Computers, Environment and Urban Systems, vol. 30, no. 6, pages 712–725, 2006. 8, 25

[Jin 2014] Huiliang Jin. *Proxemic Interaction and Migratable User Interface: applied to Smart City.* Ph.D.thesis, Ecole Centrale de Lyon, 2014. 1, 25, 26, 27

[Karimi 2004] HA Karimi. *Telegeoinformatics: Current trends and future direction.* In Telegeoinformatics: location-based computing and services, pages 2–20. Boca Raton, FL : CRC Press, 2004. 8

[Karpischek *et al.* 2009] Stephan Karpischek, Florian Michahelles, Florian Resatsch and Elgar Fleisch. *Mobile Sales Assistant - An NFC-Based Product Information System for Retailers.* In First International Workshop on Near Field Communication, 2009. NFC'09., pages 20–23. IEEE, Feb 2009. 34

[Kerner & Konhäuser 1994] Boris S Kerner and P Konhäuser. *Structure and parameters of clusters in traffic flow.* Physical Review E, vol. 50, no. 1, page 54, 1994. 36

[Kesting *et al.* 2007] Arne Kesting, Martin Treiber and Dirk Helbing. *General Lane-Changing Model MOBIL for Car-Following Models.* Transportation Research Record: Journal of the Transportation Research Board, vol. 1999, pages 86–94, 2007. 64

[Kesting *et al.* 2008] Arne Kesting, Martin Treiber and Dirk Helbing. *Agents for traffic simulation.* arXiv preprint arXiv:0805.0300, 2008. 35, 36, 37

[Kindberg *et al.* 2002] Tim Kindberg, John Barton, Jeff Morgan, Gene Becker, Debbie Caswell, Philippe Debaty, Gita Gopal, Marcos Frid, Venky Krishnan, Howard Morris*et al.* *People, places, things: Web presence for the real world.* Mobile Networks and Applications, vol. 7, no. 5, pages 365–376, 2002. 11

[Koeppel 2000] Ian Koeppel. *What are location services?–from a GIS perspective.* Environmental Systems Research Institute (ERSI) White Paper, 2000. 7

[Kolski 2011] Christophe Kolski, (Ed.). *Human-Computer Interactions in Transport*. John Wiley & Sons, Inc, 2011. 89

[Ksontini *et al.* 2014] Feirouz Ksontini, René Mandiau, Zahia Guessoum and Stéphane Espié. *Affordance-based agent model for road traffic simulation.* Autonomous Agents and Multi-Agent Systems, vol. 29, no. 5, pages 821–849, 2014. 54

[Kuechler & Vaishnavi 2008] Bill Kuechler and Vijay Vaishnavi. *On theory development in design science research: anatomy of a research project.* European Journal of Information Systems, vol. 17, no. 5, pages 489–504, 2008. 16

[Lighthill & Whitham 1955] Michael J Lighthill and Gerald Beresford Whitham. *On kinematic waves. II. A theory of traffic flow on long crowded roads.* In Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, volume 229, pages 317–345. The Royal Society, 1955. 36

[Luckenbach *et al.* 2005] Thomas Luckenbach, Peter Gober, Stefan Arbanowski, Andreas Kotsopoulos and Kyle Kim. *TinyREST-a protocol for integrating sensor networks into the internet.* In Proc. of REALWSN, pages 101–105, 2005. 12

[M. Treiber 2007] M. Treiber. `http://www.traffic-simulation.de`, 2007. Consulted on 03/21/2016. 37

[Mayer *et al.* 2014] Simon Mayer, Andreas Tschofen, Anind K. Dey and Friedemann Mattern. *User Interfaces for Smart Things – A Generative Approach with Semantic Interaction Descriptions.* ACM Trans. Comput.-Hum. Interact., vol. 21, no. 2, pages 12:1–12:25, Feb 2014. 11

[Metropole 2014] Lyon Metropole. *Smart City Lyon.* `http://www.business.greaterlyon.com/smart-city-lyon-process-47.html`, 2014. 3

[Miller & Horowitz 2007] J. Miller and E. Horowitz. *FreeSim - a free real-time freeway traffic simulator.* In 2007 IEEE Intelligent Transportation Systems Conference, pages 18–23, Sept 2007. 37

[Monteil *et al.* 2014] J. Monteil, R. Billot, J. Sau and N. E. El Faouzi. *Linear and Weakly Nonlinear Stability Analyses of Cooperative Car-Following Models.* IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 5, pages 2001–2013, Oct 2014. 34

[Mrissa *et al.* 2015] Michaël Mrissa, Lionel Médini, Jean-Paul Jamont, Nicolas Le Sommer and Jérôme Laplace. *An Avatar Architecture for the Web of Things.* Internet Computing, March 2015. ix, 32, 33

[Mulloni *et al.* 2012] Alessandro Mulloni, Hartmut Seichter, Andreas Dünser, Patrick Baudisch and Dieter Schmalstieg. *360 Panoramic Overviews for Location-based Services.* In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 2565–2568, New York, NY, USA, 2012. ACM. 106

[National Intelligence Council 2008] National Intelligence Council. *Disruptive Civil Technologies - Six Technologies with Potential Impacts on US Interests Out to 2025.* Conference Report CR, April 2008. 11

[Nouvier 2007] J Nouvier. *Entrez dans le monde des ITS = Enter the world of ITS.* Lyon: CERTU, CD-ROM TU CE12 10450, 2007. 38, 39

[Pasley 2005] James Pasley. *How BPEL and SOA are changing Web services development.* IEEE Internet Computing, vol. 9, no. 3, pages 60–67, 2005. 30

[Prehofer *et al.* 2007] Christian Prehofer, Jilles van Gurp and Cristiano di Flora. *Towards the web as a platform for ubiquitous applications in smart spaces.* In Second Workshop on Requirements and Solutions for Pervasive Software Infrastructures (RSPSI), at Ubicomp, volume 2007, 2007. 11

[Priyantha *et al.* 2008] Nissanka B. Priyantha, Aman Kansal, Michel Goraczko and Feng Zhao. *Tiny Web Services: Design and Implementation of Interoperable* *and Evolvable Sensor Networks*. In Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, pages 253–266, New York, NY, USA, 2008. ACM. 13

[Richards 1956] Paul I Richards. *Shock waves on the highway*. Operations research, vol. 4, no. 1, pages 42–51, 1956. 36

[Richardson & Ruby 2008] Leonard Richardson and Sam Ruby. Restful web services. " O'Reilly Media, Inc.", 2008. 30

[Ronzani 2009] Daniel Ronzani. *The battle of concepts: Ubiquitous Computing, pervasive computing and ambient intelligence in Mass Media*. Ubiquitous Computing and Communication Journal, vol. 4, no. 2, pages 9–19, 2009. 4

[Schilit & Theimer 1994] B. N. Schilit and M. M. Theimer. *Disseminating active map information to mobile hosts*. IEEE Network, vol. 8, no. 5, pages 22–32, Sept 1994. 5

[Schiller & Voisard 2004] Jochen Schiller and Agnès Voisard. Location-based services. Elsevier, 2004. 5, 7, 8

[Schmidt *et al.* 1999] Albrecht Schmidt, Michael Beigl and Hans-W Gellersen. *There is more to context than location*. Computers & Graphics, vol. 23, no. 6, pages 893–901, 1999. 27

[Simulation of Urban MObility 1998] Simulation of Urban MObility. `http://sumo.sourceforge.net`, 1998. Consulted on 05/21/2016. 37

[Spiekermann 2004] Sarah Spiekermann. *General Aspects of Location-Based Services*. Location-Based Services, vol. 9, pages 14–33, 2004. ix, 10, 19, 23, 24

[Tarapiah *et al.* 2014] Saed Tarapiah, Shadi Atalla and B Alsayid. *Smart on-board transportation management system Geo-Casting featured*. In 2014 World Congress on Computer Applications and Information Systems (WCCAIS), pages 1–6. IEEE, Jan 2014. 12

[The EPCglobal] The EPCglobal. `http://www.gs1.org/epcglobal`. Consulted on 11/04/2016. 13

[Treiber & Kesting 2013] Martin Treiber and Arne Kesting. *Traffic flow dynamics: data, models and simulation*. Springer, 2013. 105

[Treiber *et al.* 1999] Martin Treiber, Ansgar Hennecke and Dirk Helbing. *Derivation, properties, and simulation of a gas-kinetic-based, nonlocal traffic model*. Physical Review E, vol. 59, no. 1, page 239, 1999. 36

[Treiber *et al.* 2000] Martin Treiber, Ansgar Hennecke and Dirk Helbing. *Congested traffic states in empirical observations and microscopic simulations*. Physical Review E, vol. 62, no. 2, pages 1805–1824, 2000. 61

[Treiber 2002] Martin Treiber. *The Lane-change Model MOBIL*. `http://www.vwi.tu-dresden.de/~treiber/`, 2002. vii, 66, 67

[Treiber 2010] Martin Treiber. *Longitudinal Traffic Model: The IDM*. `http://www.vwi.tu-dresden.de/~treiber/`, 2010. vii, 63

[Viegas & Lu 2004] José Viegas and Baichuan Lu. *The Intermittent Bus Lane signals setting within an area*. Transportation Research Part C: Emerging Technologies, vol. 12, no. 6, pages 453 – 469, 2004. 38

[Viegas *et al.* 2007] J. Viegas, R. Roque, B. Lu and J. Vieira. *Intermittent Bus Lane System: Demonstration in Lisbon, Portugal*. Presented at 86th Annual Meeting of the Transportation Research Board, Washington, D.C., 2007. 38

[Walnes 2013] Joe Walnes. *About Xstream*. `http://x-stream.github.io/`, 2013. 70

[Wang *et al.* 2014a] Chen Wang, Bertrand David and René Chalon. *Dynamic road lane management: A smart city application*. In 3rd International Conference on Advanced Logistics and Transport (ICALT'2014), pages 72–77. IEEE, May 2014. doi: `http://dx.doi.org/10.1109/ICAdLT.2014.6864085`. 16, 41

[Wang *et al.* 2014b] Chen Wang, Bertrand David and René Chalon. *A Smart City Case Study: Dynamic Management of Road Lanes.* In Panos Markopoulos Norbert Streitz, (Ed.), Second International Conference, DAPI 2014, Held as Part of HCI Interational 2014, Lecture Notes in Computer Science, pages 629–640. Springer, jul 2014. doi: `http://dx.doi.org/10.1007/978-3-319-07788-8_58`. 16, 41

[Wang *et al.* 2015] Chen Wang, Bertrand David, René Chalon and Chuantao Yin. *Dynamic road lane management study: A Smart City application.* Transportation Research Part E: Logistics and Transportation Review, vol. 89, pages 272–287, 2015. doi: `http://dx.doi.org/10.1016/j.tre.2015.06.003`. 17, 53

[Wang *et al.* 2016] Chen Wang, Bertin Nayo, Bertrand T. David and René Chalon. *Allocation dynamique des voies de circulation : tests d'interprétation de la signalisation.* In Les Rencontres de la Mobilité Intelligente, Paris, France, Jan 2016. 17, 89

[Weiser & Brown 2014] M. Weiser and J. Brown. *The coming age of calm technology.* `http://www.johnseelybrown.com/calmtech.pdf`, 2014. Consulted on 11/05/2014. 10

[Weiser 1999] Mark Weiser. *The Computer for the 21st Century.* SIGMOBILE Mob. Comput. Commun. Rev., vol. 3, no. 3, pages 3–11, July 1999. 3

[Xiong 2012] Zhang Xiong. *Smart City and Data Vitalisation.* The 5th Beihang Centrale Workshop, January 2012. 5

[Xu 2013] Tao Xu. *The Context-aware Middleware in Ambient Intelligence.* Ph.D.thesis, Ecole Centrale de Lyon, 2013. 25, 26

dernière page de la thèse

AUTORISATION DE SOUTENANCE

Vu les dispositions de l'arrêté du 7 août 2006,

Vu la demande du Directeur de Thèse

Monsieur B. DAVID

et les rapports de

M. C. KOLSKI
Professeur - Université de Valenciennes - Le Mont Houy - 59313 Valenciennes cedex 9

et de

M. A. SEFFAH
Professeur - Lappeenranta University of Technology - Department of Software and Innovation
P.O. Box 20 - FI-53851 Lappeenranta - Finlande

Monsieur WANG Chen

est autorisé à soutenir une thèse pour l'obtention du grade de **DOCTEUR**

Ecole doctorale INFOMATHS

Fait à Ecully, le 23 mai 2016

P/Le directeur de l'E.C.L.
La directrice des Etudes

ECOLE CENTRALE DE LYON
Service
Scolarité
ECULLY

M-A. GALLAND