



HAL
open science

Construction of database management system for the vehicle-to-vehicle communications

Cen Jiang

► **To cite this version:**

Cen Jiang. Construction of database management system for the vehicle-to-vehicle communications. Databases [cs.DB]. 2010. dumas-00530718

HAL Id: dumas-00530718

<https://dumas.ccsd.cnrs.fr/dumas-00530718>

Submitted on 29 Oct 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Construction of database management system for the vehicle-to-vehicle communications

Supervisors: Michel PARENT
Laurent BOURAOUI
Paulo RESENDE

Author: Cen JIANG
TELECOM-Bretagne

August 2010

Table of content

List of figures.....	3
Abstract.....	4
Introduction.....	5
1. Cybercar-2 communication architecture.....	8
1.1 Context.....	8
1.2 Communication V2V.....	9
1.3 Cybercar-2 communication architecture.....	10
1.4 RT-Maps and its own database(STDB).....	13
1.5 Problem and innovation.....	14
2. Database and web applications.....	16
2.1 Formalization of the data structure.....	16
2.2 Development of RT-Maps components.....	19
2.3 Web applications.....	23
3. Data interchange for complicated decisions.....	26
3.1 Modelization of the global map.....	26
3.2 Example 1: Overtaking decision.....	27
3.3 Example 2: Intersection decision.....	30
3.4 Extended applications of database system.....	32
Future work.....	34
Conclusion.....	35
References.....	36

List of figures

Figure 1 Stanford Racing Team's Junior car.....	6
Figure 2 INRIA's Cycab.....	6
Figure 3 Cybercar-2 communication architecture.....	11
Figure 4 Driveless Vehicle Architecture.....	12
Figure 5 Software RT-Maps(Real Time, Multi-sensor, Advanced Prototyping Software)	13
Figure 6 New data management structure.....	15
Figure 7 Flow chart about how a automated vehicle runs.....	16
Figure 8 Database Schema for the vehicle management system.....	19
Figure 9 RT-Maps Components: RecorderMYSQL.....	19
Figure 10 RT-Maps Components: PlayerMYSQL.....	19
Figure 11 Usage of recorder and player.....	20
Figure 12 Data stocked in the XML file.....	21
Figure 13 Result of the simulation.....	21
Figure 14 Vehicle's trajectory during a time slot on the Google Maps.....	22
Figure 15 Vehicles management system – vehicles' real-time position in Google Maps.....	23
Figure 16 Vehicle management system – vehicles' speed chart.....	24
Figure 17 GPS position stocked in the MYSQL database.....	24
Figure 18 Modelization of Map system.....	27
Figure 19 Map structure schema including notes and edges.....	27
Figure 20 Overtaking scene.....	28
Figure 21 Overtaking decision flow chart.....	29
Figure 22 Intersections decision flow chart for a vehicle arriving to the intersection.....	32

Abstract

Cybernetic Transport Systems (CTS) based on fully automated urban vehicles (the cybercars) will be seen on city roads and on new dedicated infrastructures in the short term future. The group IMARA of INRIA-Paris is on the research of the development, integration and experimentation of Information and Communication Technology (ICT) in order to improve road transport, especially on vehicle-to-vehicle communication and vehicle-to-infrastructure communication. The first chapter of my report will give the context of our project Cybercar 2, about the vehicle-to-vehicle communication, OLSR protocol and so on. To measure the performance of vehicle communication, we need to process large quantities of data. RT-Maps (Real Time, Multisensor, Advanced Prototyping Software) is used to collect different kinds of information(Speed, GPS position...), but it has limited capacity. New database as well as some RT-Maps components are created to record the data, and improve the performance of communication, also some web applications are designed to show the information dynamically and automatically for a better vehicle management system, the second part will explain it in detail. And some further research work about the complicated decisions as overtaking, intersection will be explained in the third part.

Key words: Cybercar 2, vehicle-to-vehicle communication, database, RT-Maps

Introduction

Our living environments are facing some serious problems which are caused by the private automobile controlled by human beings. Accidents, energy congestion and noise greatly degrade the quality of life. The consequence is that: the traditional trade declines, people like to live in the suburbs and cities become less attractive to tourists. Although public transport systems have made many recent improvements by information technology to reduce the use of private vehicles, in many cases, the car still offers a much better service at the individual level.

An alternative solution to private car is the small automatic vehicle that could offer the same flexibility but much less nuisances. It is visible that in the short term future, Cybernetic Transport Systems (CTS) [1] based on fully automated urban vehicles (the Cybercars) [3] will be seen on city roads and on new dedicated infrastructures and become part of public transport system and non-motorized transport, providing passenger services anywhere at any time. These systems can also evolve to ensure the delivery of goods door to door or garbage collection automatically which will change our daily life greatly.

During the recent years, many countries have invested heavily in research of the automated vehicles and got good results. The advantages of autonomous driving and new transport system are numerous. First, they reduce congestion and produce a better traffic flow, air quality and energy conservation. Second, the system is much safer than manual driving. Third, the cars can drive autonomously to a remote parking area when it is not needed, thus leaving the space free for pedestrians and cyclists. Finally, it could take into account the requirements of the private consumer to optimize system performance, improve people's lives.



Figure 1 *Stanford Racing Team's Junior car* won a good result-- without human intervention.

Stanford's autonomous car has a number of GPS sensors and lasers, a camera, and other equipment to help it make its way through the course. In 2007, during the DRAPA (Defense Advanced Research Projects Agency) urban challenge, it drove around a parking lot in Mountain View, made three-point turns, and followed the rules at a four-way stop,

In INRIA-Paris, the most representative driverless vehicle model in our project Cybercar-2 for our research work is called CyCab. It has been developed and evaluated in the draft CyberCars (www.cybercars.org) and are currently deployed. The Cocab is a four wheeled electric vehicle which has robotic abilities, it can drive fully autonomously. It has been designed to transport up to two persons in downtown areas, pedestrian malls, large industrial or amusement parks and airports. It has a camera for remote control, a Joystick for manual driving, multimedia terminal, linear camera for platoon driving, infrared targets for platoon driving, Ibeo Laser for collision avoidance, Steering actuator, one electric drive motor per wheel, one electric brake motor per wheel and eight lead-acid batteries.



Figure 2 *INRIA's Cocab*

The goal of project Cybercar 2 is to introduce cooperation in cybernetic transport systems (CTS) through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, in order to enable higher traffic flows through improved network efficiency and more complex networks.

For the vehicle-to-vehicle and vehicle-to-infrastructure communication, there are many different kinds of information that should be considered to control and manager, like the speed, GPS position, with considering the future usages, some private services like the reservation could be taken into account. RT-Maps is a widely used software in the automated vehicle communication. It

is used to collect different kinds of real-time information from different vehicles at the same time. As more co-operations between vehicles is needed, it has considerable limits: it is hard to reuse the data for the further analysis; for the demonstration, with only the number to show to the users; it is less useful than using some tools like charts and Google Visualization API to show these data dynamically; it is hard to provide some expanded services.

In this case, a new vehicle information management system is designed to stock the information which is collected from the Cycab by RT-Maps. And in this way, the data could be reused easily for the further analysis when necessary, also it is better to monitor the real-time events of different vehicles. After designing the database to stock different kinds of real-time events in a unified model for the management of vehicle communication, some RT-Maps components are designed by Microsoft Visual Studio C++ to let the database server easily collect to different driverless vehicles, and also web applications are created to show them in different ways (show the speed in the chart, show the GPS position in the Google Maps and Google Earth...), as the database is designed in a unified form for different data, it is easy to expand its application, for the short term future, when all the necessary information is well collected, it could be expanded to be a widely used vehicle management system, to control and manage vehicle communications.

In my report, the first chapter will give the fundamental context of our project, about the Cybercar 2, the vehicle-to-vehicle communication, OLSR protocol, the software RT-Maps and its own database. The second part focuses on the development of the new database system for the automated vehicle and the realization of web applications. And the third part explains some further research work about the complicated decisions.

1. Cybercar-2 communication architecture

1.1 Context

The Project Cybercar-2 is driven by the vision that, in the short term future, Cybernetic Transport Systems (CTS) based on fully automated urban vehicles (the Cybercars) will be seen on city roads and on new dedicated infrastructures. Such systems have been developed and evaluated in the scope of the CyberCars (www.cybercars.org) and CyberMove (www.cybermove.org) projects of the 5th FWP and are now being deployed.

However, presently these CTS can only operate in low demand environments where little interaction between vehicles is anticipated. In order for these systems to address high demands, more cooperation between vehicles is needed. This is the topic of this Project, based on vehicle-to-vehicle and vehicle-to-infrastructure communications and vehicles coordination. We will address in particular the cooperation between vehicles running at close range (platooning) and at intersections (merging,crossing).

This research is closely linked to research on the cooperation between driver operated vehicles with Advanced Driver Assistance Systems (ADAS) to help the drivers in the improvement of safety and efficiency in the operation of their vehicles. Therefore, the project will keep close links to projects on driver assistance systems based also on close range communication. As far as

possible, the same technologies will be used for the communications and exchange of information such as the CALM(Communications Access For Land Mobiles) architecture. The work in CyberCars-2, however, will focus on fully automated vehicles. We believe that some of the algorithms developed for the CTS, will be applicable for driving assistance when the control of the vehicle is taken away from the vehicle drivers in case of some improper actions.

The transition between Cybercars and ADAS vehicles will be made easier with the forecasted arrival of "dual-mode" vehicles which will offer an automatic mode in specific situations such as platooning and in specific locations such as automated parking lots and manual-assisted mode in regular situations.

The most representative model of the project Cycab has two mode: aided-conduct mode and autonomous mode. The former is aided by the assisted use of a joystick and a touch screen. The joystick is connected to the computer that controls the vehicle and also provides the convenience and safety, the speed can be limited in the corners and special areas. The touch screen allows the user to communicate with the system to obtain information such as location, the autonomous vehicle, or other services such as tourist information. In INRIA there are 6 Cycabs that we could use for our experience environment.

In order to achieve the autonomous driving, there are numerous data that should be processed for the V2I/V2V communication, including the map information, road information, GPS position, Speed, traffic situation, and so on. To process these huge data, and also for the replay and comparison for the further analysis, using a center database management system with a good record format to organize the data is inevitable .

1.2 Vehicle-to-vehicle / vehicle-to-infrastructure communication

Large scale control of traffic requires us to construct a scalable, cooperative global system to control the vehicles in different levels: over an urban, metropolitan or regional area by a regional control center (RCC), local sub-centers, and individual vehicles. The use of the vehicle-to-vehicle (V2V) [5] and vehicle-to-infrastructure (V2I) communication systems could improve the road efficiency and traffic safety on urban, intercity arterial and rural roads[9]. It chooses the best available wireless technologies (focusing on 802.11 devices) and the network operation is ensured at a higher level by routing protocols, adaptable to mobility situations based on the existing Optimized Link State Routing Protocol (OSLR) [4], which is developed by INRIA for Ad-hoc mobile networks.

OSLR operates as a table driven and proactive protocol, exchanges topology information with other nodes of the network regularly. Moreover, the protocol does not require reliable transmission of control messages. In fact, the ad-hoc networks for vehicles can be created quickly and the impact of the car to and from the network are minimal. In addition, the protocol OSLR is designed for multi-hop networks [8] with a strong and effective mechanism for the flow of data across the network. Because large networks may be inappropriate for V2V communication, a series of changes to OSLR is proposed to limit the information relays of the topology and quickly create suitable Ad-hoc networks for the dynamic communication vehicle . The control information sent by vehicle is relayed in a confined space in order to obtain the routing information and network topology nearby. This information is sufficient to allow broadcasting and data relayed to a larger area nearby known as topology.

1.3 Cybercar-2 communication architecture

The Cybercar-2 communication architecture comprises Physical Layer, MAC Layer, Network Layer, System Service Layer and Control Component / Management Plane [6] .

In the physical layer, three communication technologies have been considered according to

this classification: IEEE 802.11p V2V/V2I for direct communication, WLAN technology (Wireless Local Area Networks) based on IEEE 802.11a/b/g to provide information support; WWAN technologies (Wireless Wide Area Networks) provides comprehensive coverage to monitor traffic flow and improve traffic efficiency. The MAC layer is based on the IEEE 802.11 MAC and the algorithm is the Carrier Sense Multiple Access with Collision Avoidance (CSMA / CA). In the network layer, to achieve dynamic routing in this scenario, the Optimized Link State Routing rotocol (OLSR) is chosen because its behavior is appropriate for proactive communication vehicle. OLSR mesh network allows the vehicle to be quickly created and reconfigured dynamically, because it is designed for multi-hop networks with a strong and effective mechanism for delivery of data streams. Two modules were identified in the service layer: one for service discovery and another for the decision on creating the communication links.

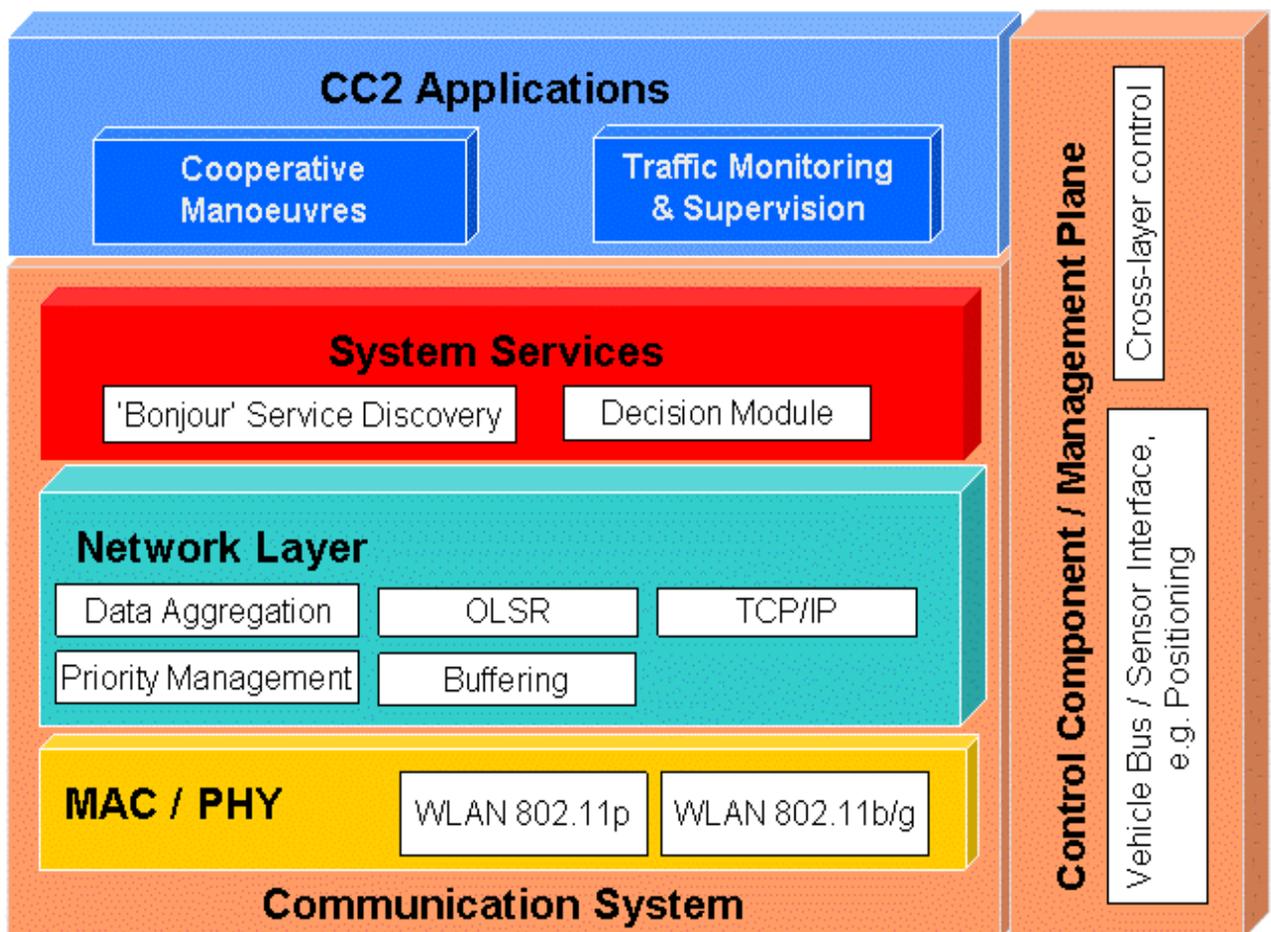


Figure 3 *Cybercar-2 communication architecture*

For the software architecture, it includes HMI(Human-Machine Interface) module, control

module, perception module, planning module and so on. HMI module provides information, warning and advises to the user access to geographically referenced content, including road network, current vehicle state and planned route. The perception module is to transfer the sensor data into consist and useful information. The global route planning module is to find the shortest path between the current position and the destination and output a temporal target, also it provides route information to HMI. The local trajectory planning module computes the trajectory that respect the vehicle constraints, avoids collisions and moves towards the temporal target. All of these modules have interactions with the database called Knowledge Base as shown in Figure 4.

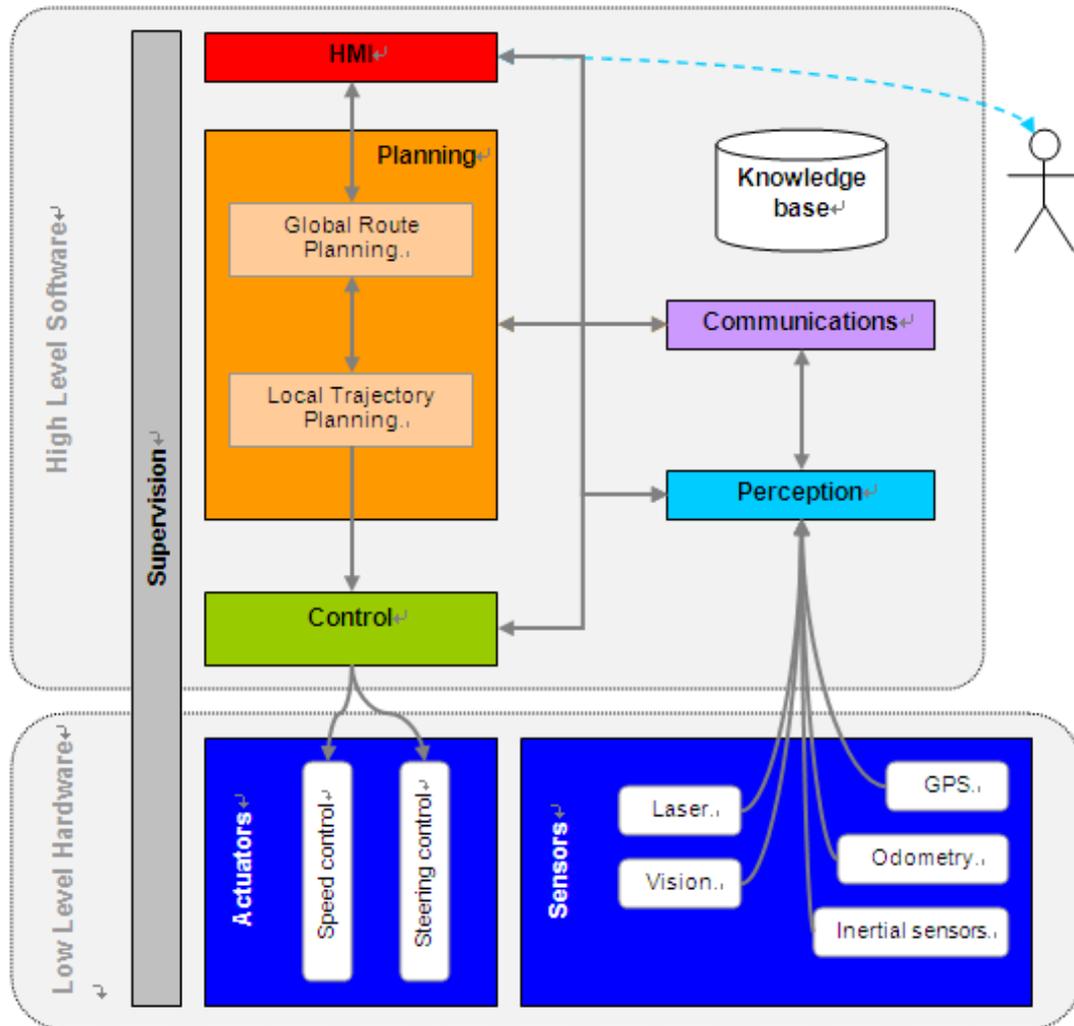


Figure 4 Drive-less Vehicle Architecture

During my internship, I focus on the development of database for stocking the different

kinds of data with a good structure for the real-time control and the manage, considering how to connecting the software RT-Maps by some components provided by RT-Maps' company and also I developed some components based on researching the working principles of RT-Maps, with the help of these components connecting to the Cycab to get different types of data from different vehicles. Also for the visualization, I created some web applications for better managing the data, like showing the trajectory on Google Maps and Google Earth.

1.4 RT-Maps (Real Time, Multi-sensor, Advanced Prototyping Software) and its own database (STDB)

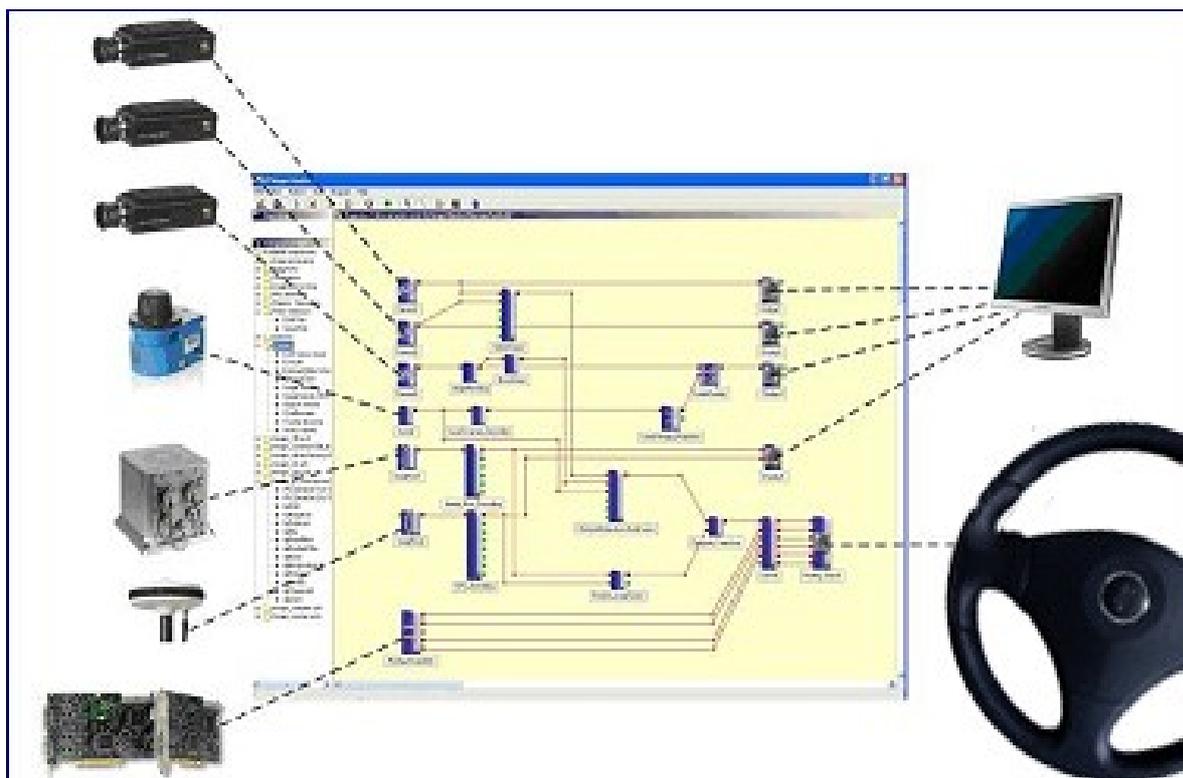


Figure 5 Software RT-Maps(Real Time, Multi-sensor, Advanced Prototyping Software)

RT-Maps (Real Time, Multi-sensor, Advanced Prototyping Software) is a software to stock large amounts of data. It is an integrated development environment provided by Intempora and is based on binary modules that can be linked to build a control application or register synchronously a set of sensors. Any suitable device for connecting to a computer can be caused by RT-Maps. The information from any type of sensors is acquired. The connections between the various graphical elements are made without any difficulty. The substitution of a sensor is quick. The comparison of information acquired by various types of sensors is simple: video cameras, analog methods, digital, CAN bus, GPS, radar ... The module "Recorder" allows the simultaneous recording tracks for a variety of information.

The information is stored in a STDB (Synchronized Time stamped Databases) format. When replayed, the sequence information is reproduced using the timestamps of the data. It is also possible to read information at a different speed: faster, slower or step by step. This database also allows synchronization between other cars using GPS signal. It is easy to demonstrate the synchronous data of different cars on the same computer and to evaluate the communication time or quality of the wireless link. So it is widely used in our vehicle communication system.

1.5 Problem and innovation

Although RT-Maps has already been widely used in the vehicle communication system for collecting different communication data from different vehicles, it has some disadvantages. First, when we save some data by timestamps, the size is too large, also some of the data is not that useful. For example, when the vehicle runs with the same speed, it is not necessary to save all the data as well as the timestamps to the database, just the start time and end time is enough, and if possible data mining algorithms could be applied for this case. Second, the data stocked in its own database is not well organized, so hard to manage with comparing to the principal database software. Third, when we want to expand its functions, RT-Maps is too limited as it provides less interfaces to other applications like web than the principal database software. As to expand its functions for the future application, new database management system is created and the data

structure is reorganized.

As shown in the graph, in the new vehicle management system, new RT-Maps components (Recorder and player) are designed to record and replay the different kinds of data from different vehicles at the same time. And if the database server doesn't work, it could be directly save to the XML files, else the data will be well stocked to the database MYSQL. For the player component, it gets the data from the XML file or from the MYSQL database, and show it. For the web platform, it gets different kinds of vehicle communication information from the database, and show it in different ways so it is easier to control and manage the different vehicles.

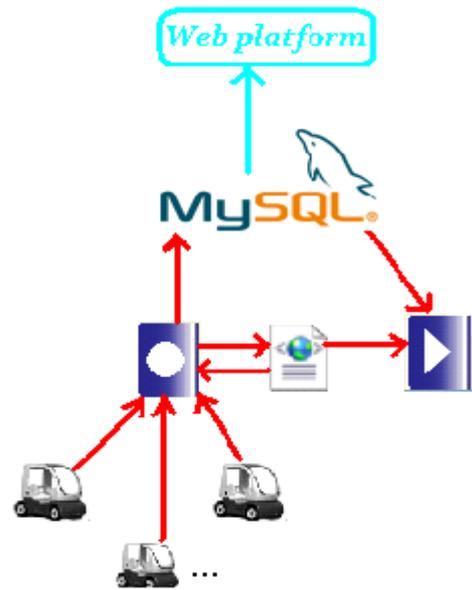


Figure 6 *New data management structure*

2. Database and web applications

2.1 Formalization of the data structure

In order to improve the vehicle-to-vehicle and vehicle-to-infrastructure communication for the project Cybercar-2, the data should be organized with a suitable structure which means that it's easy to stock, easy for the managers to understand, easy to get and use, when there is some mistake, the system manager could find it easily and avoid the accidents, of course we hope that there is no accident any more if all the components work together in a good state.

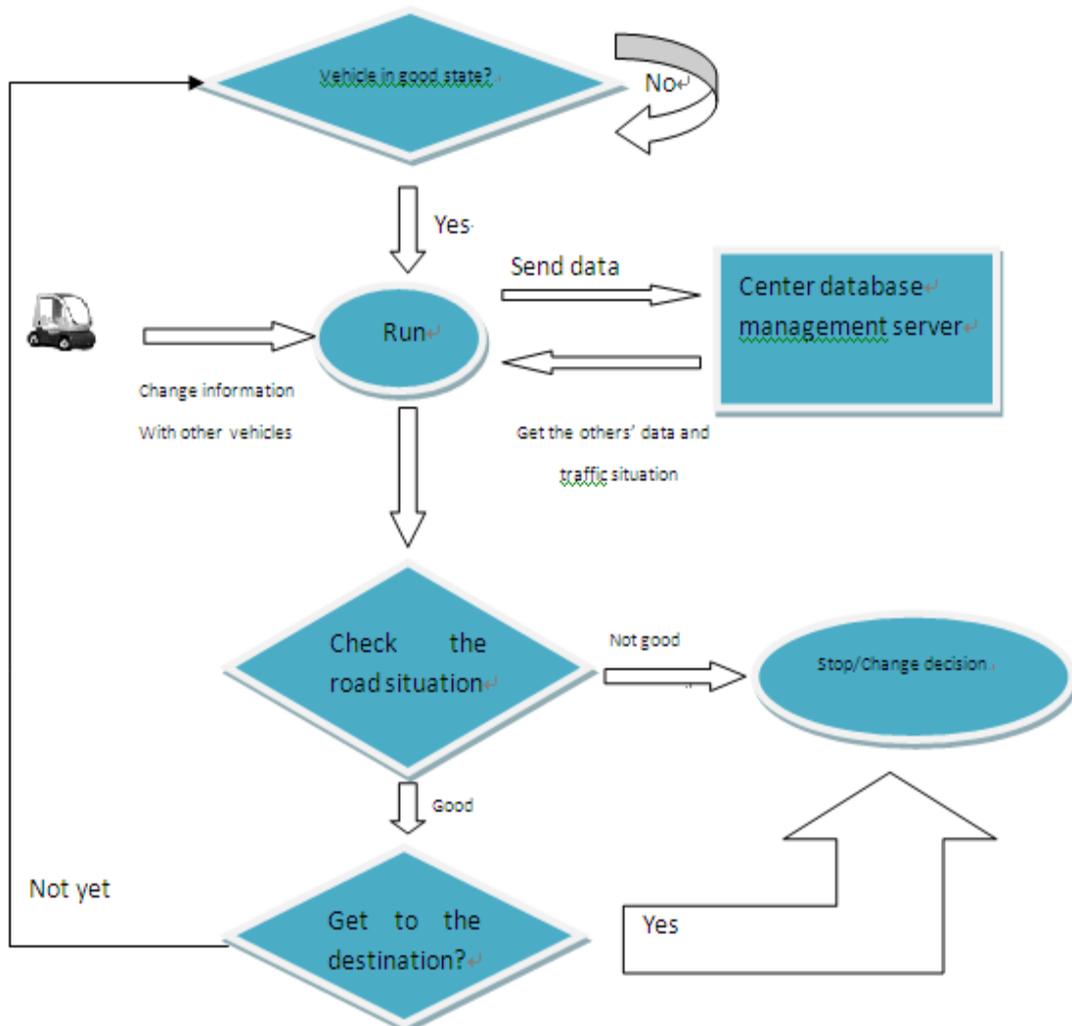


Figure 7 Flow chart about how a automated vehicle runs

Imaging that for a automated vehicle, when it runs on the road, it should first check its own

parameters, if all of them are in a normal state, it could start. And then it should know its precise position, its speed, and other relative parameters for driving. For being better controlled by the vehicle management system, it should send all these data to the center database management system. And of course it should receive information about all the circulating vehicles as well as the traffic situation, detect the obstacles around to decide which trajectory to take, whether to brake/Stop or accelerate, or if some of the roads are too crowded, it could change its decision immediately and choose other roads. All of these information should be processed in time and let the vehicle make decisions quickly. Otherwise it is very dangerous to the others. Check that if it has already reached the destination, if not, it continues running. So it is really a much harder case than the auto-control subway of which the environment is fixed and easy to be managed. But for the vehicles, the road situation is changing every day, even every second. In this case the word « real-time » has a really important meaning for the data processing. So we must have a suitable database to satisfy the requirements.

For a vehicle, the necessary information that should be shared with other vehicles and should be sent to the center server are as follows:

1. Vehicle identifier
2. GPS position
3. Speed
4. Destination
5. Emergency flag
6. Intentions (Overtaking, platooning, roundabout, intersection
7. Timestamps

Some of the information changes each second (GPS position, speed), others may stay the same during the whole period. So the data should be processed respectively: The values of speed, GPS position should be updated every time, with only the change of the vehicle identifier during some special situation. Of course for the attributes which change frequently, sometimes they keep the same value for a long time, for example, if we want to stock the speed values by fixed timestamps to the database, often we stock all the data, but if it keeps the same value, we could make some optimizations: compare the new value with the old one by timestamps, if they are the same, just save the old one, only when the value changes, save the new one. So in this way for the period the vehicles keep the same value, the database just save the values with its start timestamps and end timestamps.

The new database is still based on the RT-Maps, because there is no yet other mature softwares that could help getting different vehicles' data from their lasers, sensors, cameras at the same time by the connectors, and also RT-Maps has already developed many components for the data and image processing, it is really reliable for our team. Even when the new database was created, we developed the RecorderMYSQL and PlayerMYSQL components to get and replay the data from the vehicle or from the database.

After considering all of these, there are two solutions for designing the database. First we use the one that for each vehicle we have its id, registration, speed, GPS position, Overtaking, platooning, roundabout, intersection information and so on, and almost each attribute is a foreign key and connect to another table. All the other tables are different and doesn't have a unified format, also the main table VEHICLE wastes too much space, as just some of the data is changing all the time, it should record all the other useless values together. As we want to show each different value with its own application for the future analysis, control and manage, it's better to turn to the other solution.

The other solution is that the table VEHICLES just saves the basic parameters which don't change frequently like registration, manufacturer, Model, Seats, Image URL, Icon URL,Blocked. For each other event, a table is created with 4 attributes: ID, VID, Timestamps UTC, Value and it has the same prefix called event_. It is reasonable to formalize all the tables: the most important one is that each table will be used for the web application for further research work, like showing the speed with different charts, showing the GPS position with the Google Maps and Google Earth API. And in this way when there are some new parameters that we want to pay attention to, it is easier that we just create a table and connect it to the table VEHICLES with the foreign key. Also for the RT-Maps, it could process the unified data format easier.

For this solution, we have different tables for different events, using the VID (means the vehicle ID) to have relation with the table VEHICLES. The timestamps UTC attribute gets the system time from the vehicle, and could be reset by us. For different events, the only difference is the value. For the event Speed, the value type is float or double, but for the event GPS position, it has two values: latitude and longitude that we use a comma to separate. Here we decide to use only one attribute instead of two or even more because if every event is in the same format, it is good for the expansion and easy for the RT-Maps to process which I will explain later, as all the attribute VALUE of event is String. I just show some of the tables as follow, as the other tables of different

events are almost with the same attributes, and the expansion capacity is important for the further research.

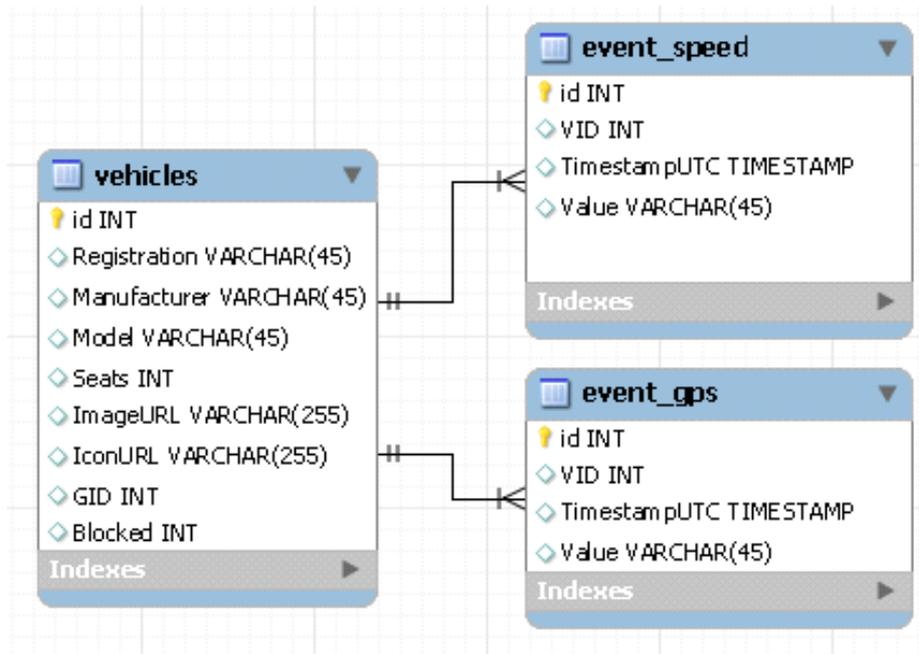


Figure 8 Database Schema for the vehicle management system

2.2 RT-Maps components and web applications

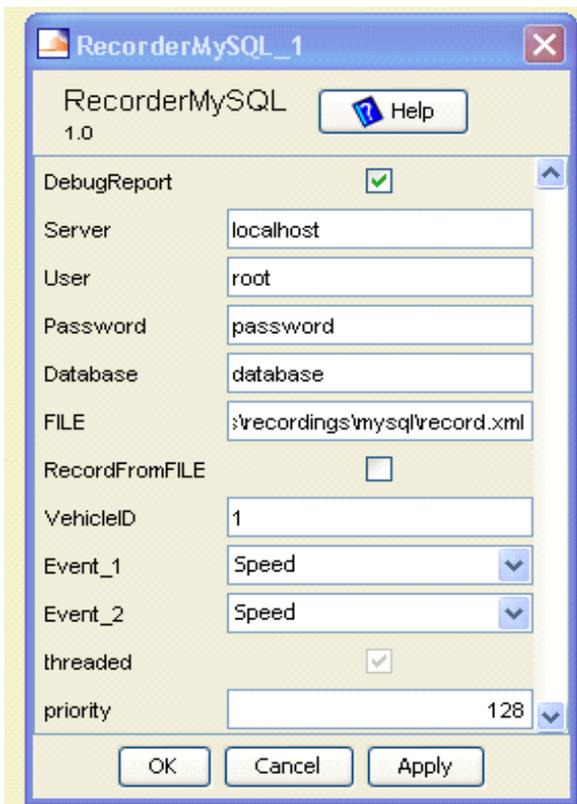


Figure 9 RT-Maps Components: RecorderMYSQL

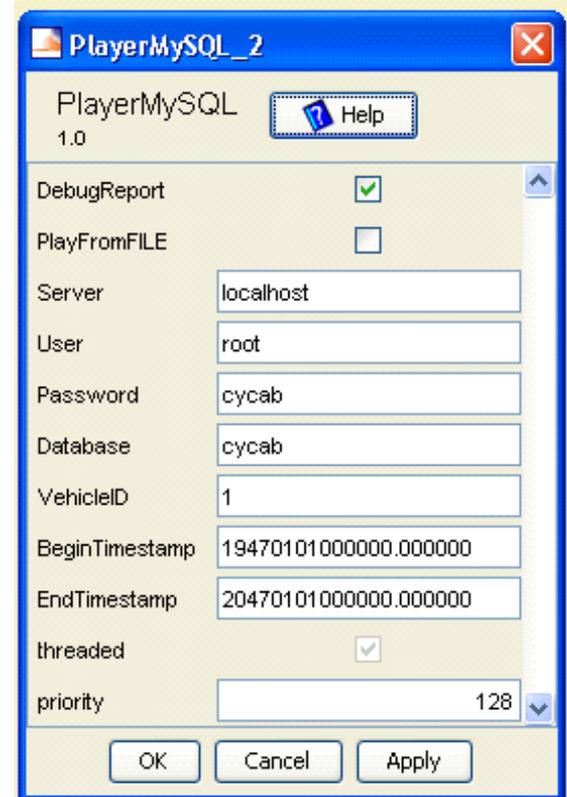


Figure 10 RT-Maps Components: PlayerMYSQL

The two RT-Maps components: RecorderMYSQL and PlayerMYSQL, designed by me using Microsoft Visual Studio C++, are used to record the data from the vehicles to the database or from the XML files when the database doesn't work and replay it from the database or the XML files.

RecorderMYSQL:

Debug Report: choose to get the debug report

Inputs: Parameters for database connection

Vehicle ID (could choose many vehicles)

File path (when the database server doesn't work, the data will be saved in a XML file, so no data will lose)

Event type(choose different types of event that we want to get from the vehicles, for example: speed, position)

Debug report (choose to show it or not)

Priority

Outputs: The relative values to the XML file or to the database

PlayerMYSQL:

Debug Report: choose to get the debug report

PlayFromFile: to get the data from the database and replay it directly with RT-Maps

Inputs: Parameters for database connection

Vehicle ID: Vehicle ID (could choose many vehicles)

BeginTimeStamp

EndTimeStamp

Priority

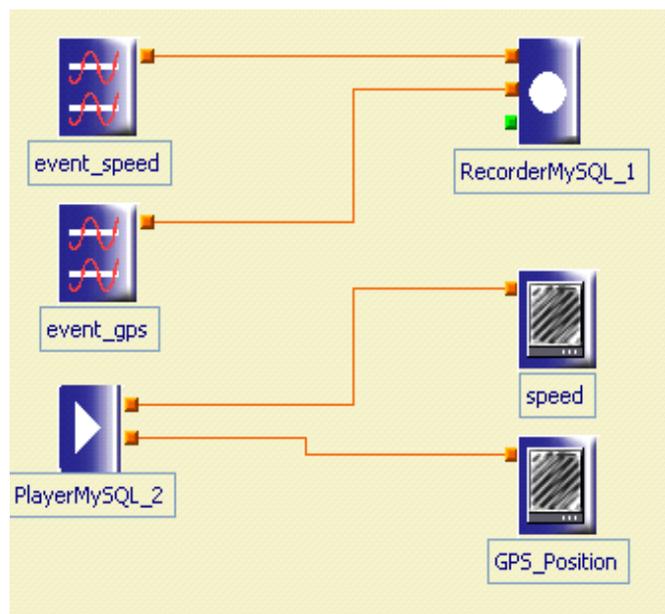


Figure 11 Usage of recorder and player

Figure 11 uses the data generator and RecorderMYSQL components in RT-Maps to generate data and stock it in the database, also uses the playerMYSQL components to play the data from the file or from the database and shows it with RT-Maps' own viewer.

```

<event table="event_speed" vid="1" timestamp_utc='20100629141521.453000' value='0' />
<event table="event_gps" vid="1" timestamp_utc='20100629141523.703000' value='48.8376415,2.10148416666667' />
<event table="event_speed" vid="1" timestamp_utc='20100629141523.703000' value='0' />
<event table="event_gps" vid="1" timestamp_utc='20100629141525.953000' value='48.837642,2.1014845' />
<event table="event_speed" vid="1" timestamp_utc='20100629141525.953000' value='0.278469' />
<event table="event_gps" vid="1" timestamp_utc='20100629141528.203000' value='48.8376426666667,2.10148533333333' />
<event table="event_speed" vid="1" timestamp_utc='20100629141528.203000' value='0.754359' />
<event table="event_gps" vid="1" timestamp_utc='20100629141530.453000' value='48.837643,2.10148583333333' />
<event table="event_speed" vid="1" timestamp_utc='20100629141530.453000' value='0.968406' />
<event table="event_gps" vid="1" timestamp_utc='20100629141532.703000' value='48.8376431666667,2.1014865' />
<event table="event_speed" vid="1" timestamp_utc='20100629141532.703000' value='0.714875' />
<event table="event_gps" vid="1" timestamp_utc='20100629141534.953000' value='48.8376416666667,2.10148933333333' />
<event table="event_speed" vid="1" timestamp_utc='20100629141534.953000' value='1.03906' />
<event table="event_gps" vid="1" timestamp_utc='20100629141537.203000' value='48.8376376666667,2.1014965' />
<event table="event_speed" vid="1" timestamp_utc='20100629141537.203000' value='0.810469' />

```

Figure 12 Data stocked in XML File

Using the tinyXML API, we save all kinds of values in a unified format, with the table name, Vehicle ID, timestamp and the value. When the database server doesn't work, the data is stocked in the XML file.



Figure 13 Result of the simulation

This is just a simulation that used to test if the database and the RT-Maps components RecorderMYSQL and PlayerMYSQL work well or not. When connecting to the automated vehicles Cycab, much more components should be used: data processing, viewer, image processing, data conversion, generators. The schema is very huge, so I don't want to show it in this report, but the principle is more or less the same, and the important part that we focus on is shown in Figure 11. Using different components to get the same type of value from different vehicles for the comparison or different types of value from the same vehicle. Using data processing/conversion components to transform the data types. Using recorder to record and player/ viewers to get the results.

As RT-Maps just could show the values, its function is limited. For the web applications, different frames are created by PHP/JavaScript to show the different types of values. As shown in the figure 15, the report type could be chosen: Speed, GPS position, reservation information and so on. Also there are different kinds of display mode: time slot, real-time, replay. We could show the real-time values from the vehicles, could choose the event start time and end time to just replay the data for a certain period of time, also by the time slot, the trajectory could be shown between the start date and the end date. Different period of auto update could be chosen, and in the right there is a vehicle list showing the ones which have already registered in the database management system.

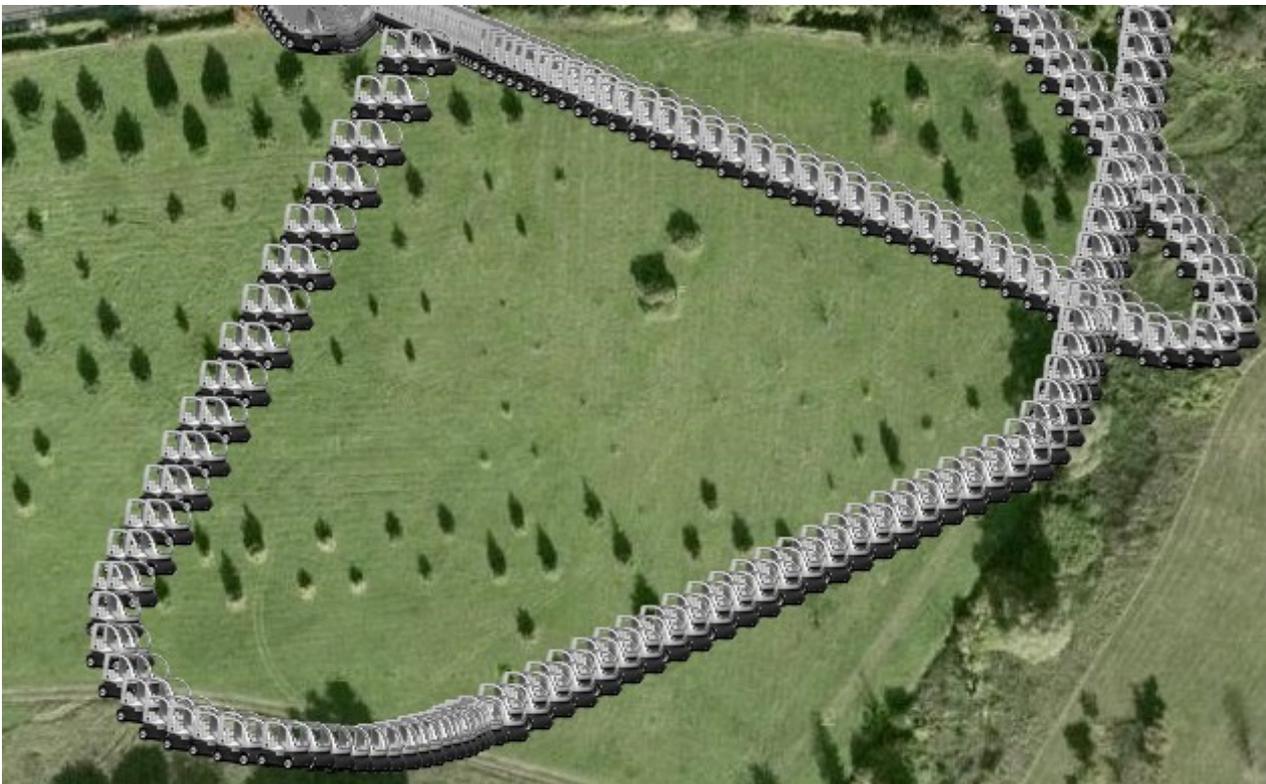


Figure 14 *Vehicle's trajectory during a time slot on the Google Maps*

2.3 Web applications

Report Type:

Display mode:

Auto Update: None 1s 5s 10s 30s

July 2010

Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Start date:

HH:mm:ss : :

End date:

HH:mm:ss : :

Check all

- Citroen C3 Red
- Citroen C3 Green
- Citroen C3 Blue
- Citroen C3 Black
- Robosoft CyCab 1
- Yamaha CyBus Closed

Powered by Google
 Données cartographiques ©2010 AND, Geocentre Consulting, MapData Sciences Pty Ltd, PSMA, MapLink, Tele Atlas - [Conditions d'utilisation](#)

Figure 15 Vehicle management system – vehicles' real-time position in Google Maps

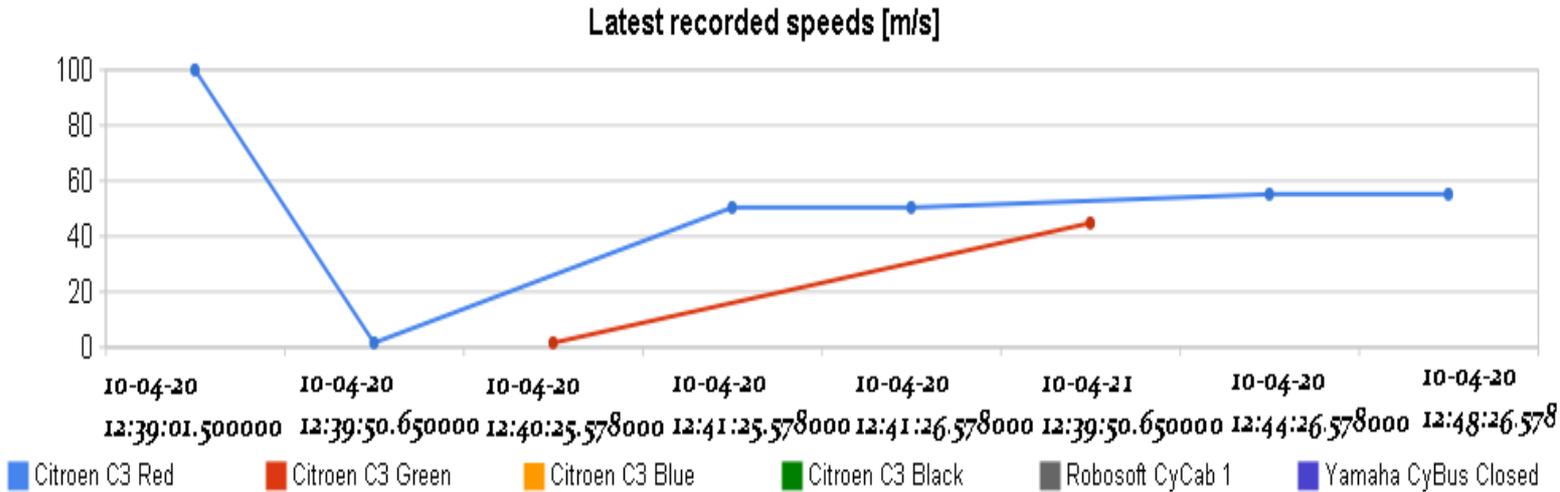


Figure 16 *Vehicle management system – vehicles' speed chart*

Figure 15 and Figure 16 show some of the web applications by Google Maps and charts. The data is from Cycab, we did some experiments and got it in INRIA-Paris, Rocquencourt.

			ID	VID	TimestampUTC	Value
<input type="checkbox"/>		<input checked="" type="checkbox"/>	2042	1	20100621143249.703000	48.8379612160181,2.10435477270935
<input type="checkbox"/>		<input checked="" type="checkbox"/>	2043	1	20100621143249.828000	48.8379612160181,2.10428662266528
<input type="checkbox"/>		<input checked="" type="checkbox"/>	2044	1	20100621143249.875000	48.8379800709018,2.10428662266528
<input type="checkbox"/>		<input checked="" type="checkbox"/>	2045	1	20100621143249.921000	48.8379800709018,2.10421552905'13
<input type="checkbox"/>		<input checked="" type="checkbox"/>	2046	1	20100621143250.406000	48.8379972601094,2.10421552905'13
<input type="checkbox"/>		<input checked="" type="checkbox"/>	2047	1	20100621143250.453000	48.8379972601094,2.10413986320'64
<input type="checkbox"/>		<input checked="" type="checkbox"/>	2048	1	20100621143252.953000	48.8380124083684,2.10413986320'64
<input type="checkbox"/>		<input checked="" type="checkbox"/>	2049	1	20100621143253.015000	48.8380124083684,2.10405965577572
<input type="checkbox"/>		<input checked="" type="checkbox"/>	2050	1	20100621143253.640000	48.8380256506945,2.10405965577572
<input type="checkbox"/>		<input checked="" type="checkbox"/>	2051	1	20100621143253.906000	48.8380256506945,2.10397516815386

Figure 17 *GPS position stocked in the MYSQL database*

Because the data of GPS position is changing all the time when the automated vehicle runs on the road, during each time stamp, the data is stocked in the database. If the vehicle stops, the value rests the same, and in the database we only have its start time and end time, as shown in

Figure 17. It is impossible to have three continuous rows with the same value.

This is the basic event database for the automated vehicle communication, and it is still possible to design some data mining algorithm to better process different types of data. For the complicated decisions based on these data, we will discuss in the next chapter.

3. Data interchange for complicated decisions

3.1 Modelization of the global map

Beside of the basic actions like getting the value of the Speed, GPS Position, we will focus on some complicated decisions: the intentions about Overtaking, platooning, roundabout, intersection and so on. For making these decisions, the automated vehicles should get the values from many event tables, analyze and compare them, if possible some additional information should be added in order to complete these actions. Take the map for example, for the intentions, the map information including real-time road traffic situation is necessary, but the precise GPS position is not that useful, just knowing that which road the vehicle is located is enough, and sometimes Google Maps' updating is not that prompt. For the control and manage system, create our own map structure is a good solution that we don't always need to rely on Google produces.

For construct the map model, we support to use to nodes, edges instead of precise positions. As for the manage and control, just know that which vehicle is in which road, and whether the traffic situation of the road and the surrounding environment is good enough for the vehicle to drive. A node stands for the crossing, and a edge represents a road with the start node and end node, so the path could be a set of relative edges. In this way, the map could be constructed in a simplified way but it satisfies the requirements and we could have our own map system which comprises the road traffic information as we could give each edge a attribute named Color to represent its traffic situation, red as block, green as normal, yellow as busy. Sometimes there exists many trajectories

from the source to the destination, and the length of each road which makes up of the trajectory and the traffic situation should be taken care of to find the most optimized path to the destination by some algorithms. When suddenly a traffic accident happens, the road color will immediately turn red, it means block, the vehicle will get the information, and change its way. So it must exchange the road information with the center server all the time and change its decision in time.

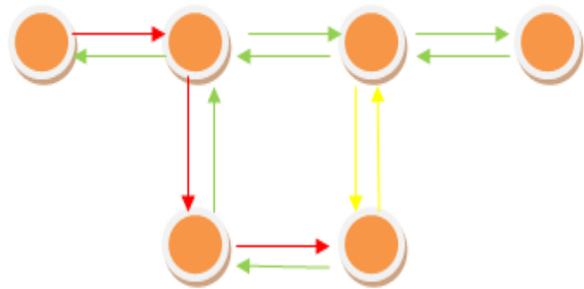


Figure 18 Modelization of Map system

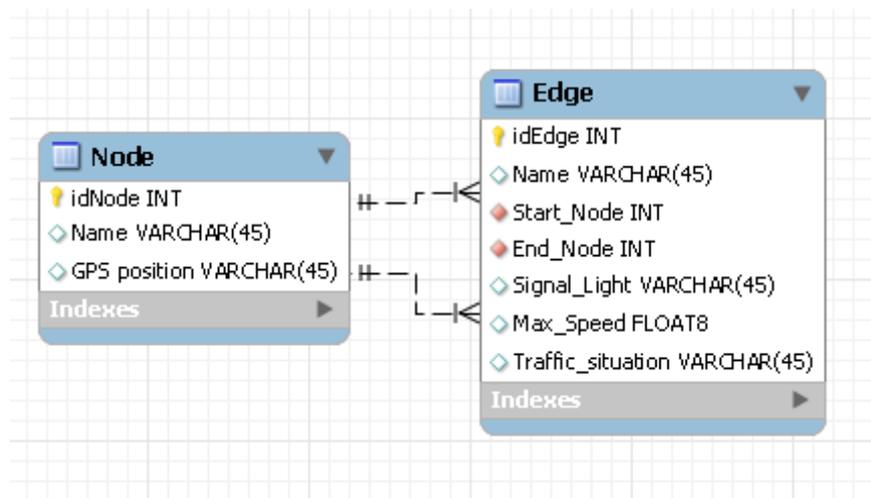


Figure 19 Map structure schema including notes and edges

3.2 Overtaking decision

After having this new structure, the complicated decisions could be made with the help of the information stocked in these tables. Take overtaking for example, if one vehicle want to overtake the one in front of it, it should first know the one's intention(accelerate or brake or suddenly stop) also the distance to the next crossing as well as if some pedestrians wants to go across the road, and get their relative speed. The pedestrian problem is really hard to process so

maybe we should rely on some sensors to make sure of it. After getting all of these values, it couldn't avoid to do some calculations in order to know that whether it has the possibility to overtake the one in front of it and what is the probability.

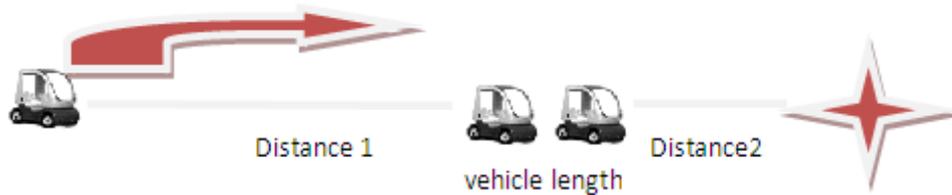


Figure 20 Overtaking scene

Once a vehicle finds that the one before it goes at a lower speed it should:

1. Check that if there is more than one preceding vehicle
 If there is only one vehicle then go to step 2
 If there are several vehicles, and none is going to overtake the one before, it can overtake all of them, and if the vehicles are closed, they could be considered as one vehicle with an exceptional length, and it should ask the last to get the relative distance, and one by one, for the calculations of the total length, then go to step 2.
2. Check that if the remaining distance is still enough for the vehicle to overtake, and at the same time, to know the front one's intention is necessary. If the one just get to its destination, it will suddenly stop, and before that it should send some signals to inform other vehicles behind it.
3. Check that if there comes another vehicle in the lane that it wants to use to overtake.

So with the values: its own speed V_0 , other vehicles' average speed V_1 , the distance from the current position to the next crossing D_0 , the distance from the current position to the front one's position D_1 , the vehicle length L , the distance for the overtaken vehicles to get to the next crossing D_2 , it could calculate like this:

Time to get to the next crossing $T_0 = D_0 / V_0$;

Time to overtake $T_1 = (L + D_1) / (V_0 - V_1)$;

Time for the overtaken vehicles to get to the next crossing $T_2 = D_2 / V_1$;

If $T_1 > T_0$ or $T_1 > T_2$, it should cancel the intention of overtaking or accelerate.

In order to complete this decision, the useful information that should be got from the database are :

Its own speed, position, road information;

Overtaken vehicles' speed, intention, position and destination.

All these data have been stocked in different tables that we have created before. So they could be reused to construct the new table OVERTAKING.

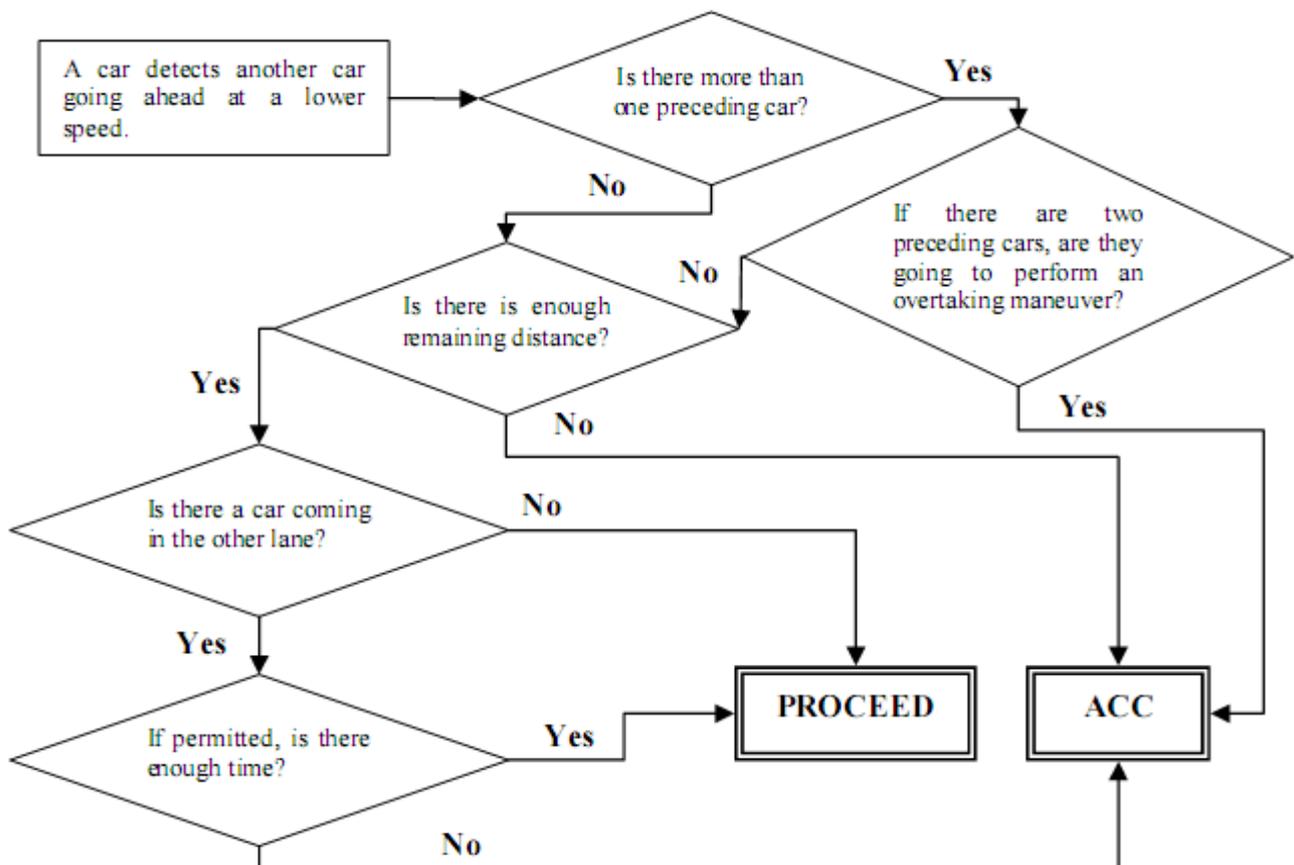


Figure 21 Overtaking decision flow chart

3.3 Intersection decision

Intersection decision is another complicated decision, for that when two or even more vehicles is going to pass the crossing, if they keep their own speeds without considering each other, the traffic accident will happen. Actually most of the accidents happen at the crossing.

When a vehicle arrives to an intersection, first it must know if there are vehicles already in it, or if they are arriving. Also it should know the direction that the other vehicles turn to. If the intersection is free and there is no vehicles around, the vehicle will continue its way. If there is a vehicle already in the intersection that interferes the vehicle's driving direction, it will have to stop before the crossing and wait until the intersection is free.

The case above is the simplest. Considering some complicated cases. When several vehicles arrive at the same time at an empty intersection, or when the intersection is free and there are several vehicles waiting to enter, a further evaluation is needed.

1. Check for the signal light
if there is a stop signal the vehicle must stop before the crossing and must re-evaluate its situation.
If there is not a stop signal it can proceed to the step 2.
2. Check intersection occupation
If the intersection is free then proceed to step 3
If there is other vehicle already in the intersection and it does not interfere in car's desired, then proceed to step 3.
If there is a vehicle in the intersection, it interfere the car's trajectory, the car must stop before the intersection and re-evaluate its situation to check that if the intersection is free.
3. Check for vehicles in the other lanes that arrive to the intersection
If there are no vehicles around, the vehicle can continue its way
If there are vehicles waiting at the intersection or arriving to it, but its

trajectories do not interfere, the vehicle can proceed.

If there are vehicles waiting at the intersection or arriving to it, and the trajectories interfere, then it's needed a priority analysis to check which car must proceed, and which ones must remain stopped.

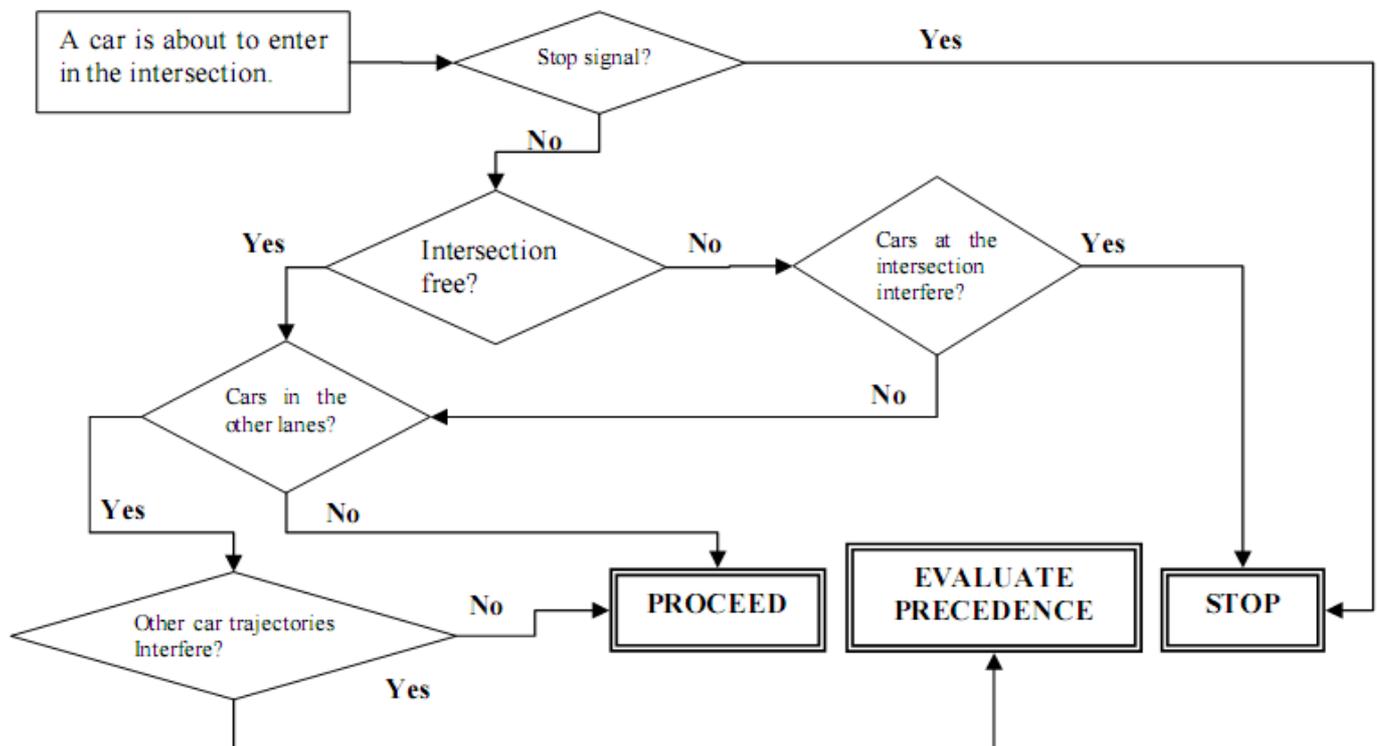


Figure 22 Intersections decision flow chart for a vehicle arriving to the intersection

If the car is already stopped before the intersection entrance, it must:

1. Check intersection occupation.

If the intersection is free then proceed to step 2.

If there is other vehicle already in the intersection and it does not interfere in vehicle's trajectory, then proceed to step 2.

If there is a vehicle in the intersection, and it interfere the vehicle's trajectory, the vehicle must remain stopped and re-evaluate its situation once the intersection is free.

2. Check for vehicles in the other lanes that arrive to the intersection.

If there are no vehicles around, the vehicle can continue its way.

If there are vehicles waiting at the intersection or arriving to it, but their trajectories do no interfere, the vehicle can proceed.

If there are vehicles waiting at the intersection or arriving to it, and the

trajectories interfere, then it's needed a priority analysis to check which vehicle must proceed, and ones must remain stopped.

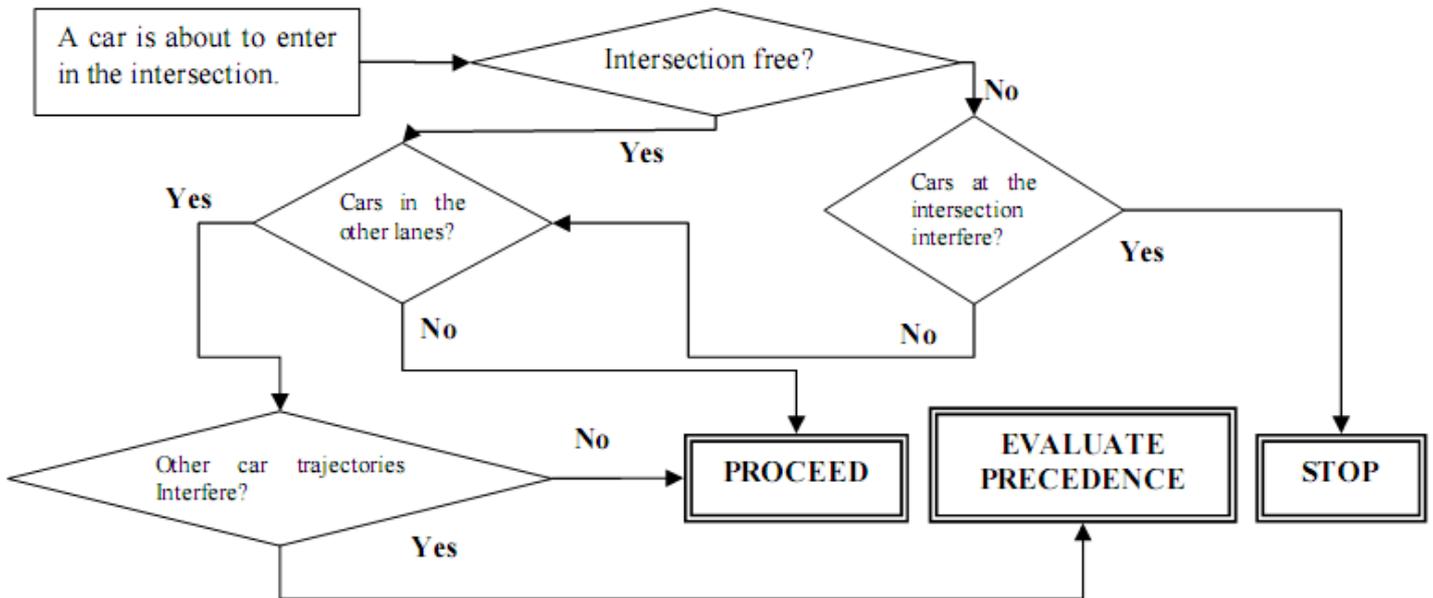


Figure 22 Intersections decision flow chart for a stopped vehicle before the crossing

For the database, considering to add to each road a attribute signal light (RED, YELLOW, GREEN) for controlling the traffic at the crossing. And also the priority should be given to the different kinds of vehicles if there are no signal light at the real environment. So when two vehicles are getting closed to the crossing almost at the same time, only the priority decides which vehicle should go first and which one should stop.

3.4 Extended applications of database system

Besides its function for the manage and control, the database could also provide some private services for the users with its abundant information. It could expand its applications for both the managers and the users. For example, the reservation. Users could get the information about at that moment which vehicle is the most closed to him and use it. Of course the vehicle should be free, by accessing to the database, getting the GPS position and comparing the distance. Once the vehicle is reserved by somebody, it could not be reserved by others, only in the case that

during a certain period of time, it is still not used after being reserved, the reservation will be canceled in a fixed period. And when something is wrong, the message will be sent to inform that the user should reserve others again.

For other applications like delivering goods, letters or pick up children, different types of automated vehicles could be granted for different usages. So there is no doubt that the automated vehicles will have wide applications in the future.

Future work

Transport will remain an essential part of economic sustainability and social cohesion in the future. The growth in transport needs will bring more energy congestion, traffic accidents. So the automated vehicles will be more and more taken seriously. As the calculator usually works better than the human brain, the introduction of new vehicles could surely reduce the number of traffic accidents by the human errors. But now as some of the parts are still in research, automated vehicles have not been widely used in the city, just in some suburbs. When all the control and management methods are considered, and the management system could work well without mistakes, automated vehicles will be widely produced and appears everywhere.

For my database management system, the basic event management as vehicle information, speed, GPS position has been tested and already realized. Also some data mining algorithms could be applied for the database management. But for some complicated decisions, like overtaking, intersection, they are still in the research and test period because the environment should be complicated and should be fully considered. For example, in the auto-route scene, where the vehicles drive at a terribly high speed, the information process could not catch up with the changing environment, so it is still needed to do more simulations and tests. I believe that it will have a good future.

Conclusion

As database provides a better control and management for different types of data, it has been widely used in many different domains. In our project Cybercar-2 the vehicle information management system requires a stable and real-time data exchange for the vehicle-to-vehicle communication and vehicle-to-infrastructure communication. The database I have created could be used to manage the basic vehicle communication events, with the RT-Maps components for connecting to the automated vehicle Cycab and web application for facilitating the data management and presentation. But for some complicated decisions, like platooning, overtaking, intersection, more simulations and tests are required to form a perfect system. Besides the data control and management, we could also imagine the usages for private users which makes the database multifunctional. It would be very useful for the future research work on the automated vehicle communication.

I must thank my tutors: Michel Parent, Laurent Bouraoui, Paulo Resende and all my teachers of TELECOM-Bretagne and RENNES-1, they give me many suggestions during the whole period of my stage, and I have learned lots of development and research methods.

Also I must say thanks to the IMARA group, it provides advanced environment for the experience of automated vehicles. It's a group full of energy, full of new ideas and it always chases for advanced technologies. I acquire lots of capabilities here. I'm so honored to have the opportunity to work in this group.

References

- [1] Michel Parent, Arnaud de La Fortelle **Cybercars : Past, Present and Future of the Technology** ITS World Congress 2005 (2005)
- [2] Thierry Fraichard, Michel Parent **Toward urban driverless vehicles** 2008
- [3] **CYBERCARS2 Project - Contract number 028062 Architecture of the Cooperative Cybernetics Transport System** 30/06/2008
- [4] Hassiba Asmaa Adnane **La confiance dans le routage Ad hoc : étude du protocole OLSR** 03/12/2008
- [5] Arnaud de La Fortelle, Claude Lurgeau, Paul Muhlethaler, Yasser Toor **COM2REACT: V2V COMMUNICATION FOR COOPERATIVE LOCAL TRAFFIC MANAGEMENT** 2007
- [6] Olivier Mehani, Rodrigo Benenson, S'everin Lemaignan and Thierry Ernst **Networking Needs and Solutions for Road Vehicles** Inria Rocquencourt, Imara team 2007
- [7] Asma MELKI **Système d'aide à la régulation et évaluation des transports multimodaux intégrant les Cybercars** Ecole Centrale de Lille 28/11/2008
- [8] Herv'e Rivano, Fabrice Theoleyre, Fabrice Valois **A Framework for the Capacity Evaluation of Multihop Wireless Networks** 19/03/2009
- [9] Thierry Ernst, Arnaud De La Fortelle **Car-to-car and car-to-infrastructure communication system based on nemo and MANET in IPV6** 2006
- [10] Jiazi Yi, Sylvain David, Hassiba Asmaa Adnane, Benoît Parrein, Xavier Lecourtier **Multipath OLSR: Simulation and Testbed** 02/10/2009
- [11] Natallia Kokash, Roberto Speicys, Pierre-Guillaume **A Flexible QoS-aware Routing Protocol for Infrastructure-less B3G Networks** 2009
- [12] Arun Kumar S P, Diganta Baishya, Amrendra Kumar **Link Enhancer for Vehicular Wireless ATM Communications Center for Artificial Intelligence and Robotics** Bangalore, India 24/04/2008

- [13] Philippe Owezarski **A database of anomalous traffic for assessing profile based IDS** 01/04/2010
- [14] Rodrigo Benenson, Thierry Fraichard, Michel Parent **Achievable safety of driverless ground vehicles** 10th IEEE International Conference on Control, Automation, Robotics and Vision (2008)
- [15] Rodrigo Benenson, Michel Parent **Design of an urban driverless ground vehicle** IEEE International Conference on Intelligent Robots Systems (2008)
- [16] J.Chao, Y. qi Chen, W. Chen, X. Ding, Z. Li, N. Wong, and M. Yu **An experimental investigation into the performance of GPS-based vehicle positioning in very dense urban areas** Journal of Geospatial Engineering, vol.3, pp. 59-66, 2001.
- [17] E. Yang, D. Gu, T. Mita and H. Hu **Nonlinear tracking control of a car-like mobile robot via dynamic feedback linearization** in Proceeding of Control, University of Bath, September 2004
- [18] Sebastien Boisse, Rodrigo Benenson, Laurent Bouraoui, Michel Parent, Ljubo Vlacic **Cybernetic Transportation Systems Design and Development: Simulation Software Cybercars** 2007 IEEE International Conference on Robotics and Automation - ICRA'2007
- [19] Laurent Bouraoui, Stephane Petti, Anis Laouiti, Thierry Fraichard, Michel Parent **Cybercar Cooperation for Safe Intersections** Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems (2006)
- [20] Plamen Petrov, Michel Parent **Adaptive Control for Reversing a Two-Vehicle Platoon Symposium on Control in Transportation** (2006)
- [21] A. Laouiti L. Bouraoui, A de la Fortelle. **OLSR improvement for distributed traffic applications. In Fourth Annual Mediterranean Ad Hoc Networking Workshop, MedHocNet' 2005, Porquerolles, France, June 2005**
- [22] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz and S. Thrun **Anytime dynamic a*: An anytime, replanning algorithm.** In Proceedings of the International Conference on Automated Planning and Scheduling(ICAPS), June 2005
- [23] Sebastian Bengochea, Angel Talamona, Michel Parent **A Software Framework for Vehicle-Infrastructure Cooperative Applications** IEEE ITSC ' 05 (2005)
- [24] Laurent Bouraoui, Philippe Jacquet, Anis Laouiti, Michel Parent, Laurent Viennot **Ad hoc communication between intelligent vehicles** 5th International Conference on ITS Telecommunications - ITST 2005 (2005)
- [25] Anjali Awasthi, Yves Lechevallier, Michel Parent, Jean-Marie **Proth Using data**

analysis to approximate fastest paths on urban networks

- [26] Cyril Duron, Michel Parent, Jean-Marie Proth **Analysis of the Balancing Process in a Pool of Self-Service cars** 2000
- [27] Vivien Delsart, Thierry Fraichard, Luis Martinez-Gomez **Real-Time Trajectory Generation for Car-like Vehicles Navigating Dynamic Environments** IEEE Int. Conf. on Robotics and Automation (2009)
- [28] Vivien Delsart, Thierry Fraichard **Reactive Trajectory Deformation to Navigate Dynamic Environments** European Robotics Symposium (2008)
- [29] Y. Kanayama and N. Miyake **Trajectory generation for mobile robots** In Proc. Of the Int. Symp. On Robotics Research, 1985
- [30] H. Delingette, M. Hebert and K. Ikeuchi **Trajectory generation with curvature constraint based on energy minimization** in IEEE-RSJ Int. Conf. On Intelligent Robots and Systems, 1991
- [31] Kristijan Macek, Dizan Alejandro, Vasquez Govea, Thierry Fraichard, Roland Siegwart **Safe Vehicle Navigation in Dynamic Urban Environments: A Hierarchical Approach** IROS Workshop on Planning, Perception and Navigation for Intelligent Vehicles (2008)
- [32] Christian Laugier, Dizan Alejandro, Vasquez Govea, Manuel Yguel, Thierry Fraichard, Olivier Aycard **Geometric and Bayesian Models for Safe Navigation in Dynamic Environments** *Intelligent Service Robotics* 1, 1 (2008)
- [33] Gang Chen, Thierry Fraichard **An Architecture for Automated Driving in Urban Environments** *Field and Service Robotics* Springer (Ed.) (2008)
- [34] Hanna Kurniawati, Thierry Fraichard **From Path to Trajectory Deformation** IEEE-RSJ Int. Conf. on Intelligent Robots and Systems (2007)
- [35] Rodrigo Benenson, Stephane Petti, Michel Parent, Thierry Fraichard **Integrating Perception and Planning for Autonomous Navigation of Urban Vehicles** IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2006)
- [36] Olivier Lefebvre, Florent Lamiroux, Cedric Pradalier, Thierry Fraichard **Obstacles Avoidance for Car-Like Robots. Integration And Experimentation on Two Robots** Proc. of the IEEE Int. Conf. on Robotics and Automation (2004) 4277--4282