La Rochelle small demo system design

<table>
<thead>
<tr>
<th>Deliverable no.</th>
<th>D1.7.2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dissemination level</td>
<td>Public</td>
</tr>
<tr>
<td>Work Package</td>
<td>WP 1.7.2.1: La Rochelle small demo system design</td>
</tr>
<tr>
<td>Author(s)</td>
<td>Fawzi NASHASHIBI</td>
</tr>
<tr>
<td>Co-author(s)</td>
<td>Carlos HOLGUIN, Laurent BOURAOUI, François CHARLOT, Tony NOEL, Paulo RESENDE</td>
</tr>
<tr>
<td>Status (F: final, D: draft)</td>
<td>F</td>
</tr>
<tr>
<td>File Name</td>
<td>CITYMOBIL_1.7.2.1_ La_Rochelle_demo_systems_design_INRIA_v2.doc</td>
</tr>
<tr>
<td>Project Start Date and Duration</td>
<td>01 May 2006 - 31 December 2011</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

1 Introduction 5

2 System Constraints & general design 5
   2.1 First and second versions of the cybercars circuit 7
   2.2 Final version of the cybercars circuit 9
      2.2.1 General design in demonstration Version 1 11
      2.2.2 General design in demonstration Version 2 12

3 Subsystems design 15
   3.1 Vehicle architecture (V1 & 2) 15
      3.1.1 Sensors (V1 & V2) 16
      3.1.2 Low Level architecture (V1 & V2) 19
      3.1.3 High Level architecture (V1 & V2) 21
   3.2 Stations 22
      3.2.1 Accessibility 23
      3.2.2 Power supply 24
   3.3 Localization V1 - V2 24
   3.4 High level control 24
      3.4.1 Global Planning 25
      3.4.2 Local Planning 27
      3.4.3 Trajectory coordination 28
   3.5 Safety subsystems (Obstacle detection) 30
   3.6 Communications V1 – V2 31
   3.7 VMS 33
   3.8 HMI 34
      3.8.1 On-Board HMI 34
      3.8.2 External road users HMI 36

4 Conclusions and future work 37
   4.1 Technical perspectives 37
   4.2 General perspectives 38
TABLES

TABLE 1. VEHICLES TRAJECTORIES PER ORIGIN/DESTINATION PAIR (FIRST VERSION OF THE CIRCUIT)................................................................. 8
TABLE 2. DESTINATION AND WARNING MESSAGES TABLE.................................................................................................................... 36

FIGURES

FIGURE 1 : EVOLUTION OF THE CYBERCARS CIRCUIT (RED) .................................................................................................................. 6
FIGURE 2 : WEST-EAST SECTION VIEW OF AVENUE DES AMÉRIQUES ....................................................................................................... 7
FIGURE 3 : CIRCULATION SCHEME - FIRST VERSION OF THE DEMONSTRATION CIRCUIT ................................................................. 8
FIGURE 4 : CIRCULATION SCHEME - SECOND VERSIONS OF THE DEMONSTRATION CIRCUIT .............................................................. 9
FIGURE 5 : CIRCULATION SCHEME - FINAL VERSION OF THE DEMONSTRATION CIRCUIT ................................................................. 10
FIGURE 6 : GENERAL DEMONSTRATION’S SYSTEM ARCHITECTURE ....................................................................................................... 11
FIGURE 7 : CIRCULATION SCHEME – SIMPLIFIED VERSION OF THE DEMONSTRATION CIRCUIT ............................................................ 12
FIGURE 8 : SCENARIO 1 DIAGRAM .............................................................................................................................................................. 13
FIGURE 9 : SCENARIO 2 DIAGRAM .............................................................................................................................................................. 13
FIGURE 10: SCENARIO 3 DIAGRAM ............................................................................................................................................................ 14
FIGURE 11: SCENARIO 4 DIAGRAM ............................................................................................................................................................ 14
FIGURE 12: SCENARIO 5 DIAGRAM ............................................................................................................................................................ 15
FIGURE 13: SCENARIO 6 DIAGRAM ............................................................................................................................................................ 15
FIGURE 14: EVOLUTION SINCE THE BEGINNING...................................................................................................................................... 16
FIGURE 15: LOCALIZATION + OBSTACLE DETECTION CONFIGURATION V1 ............................................................................................. 17
FIGURE 16: NEW LIDAR CONFIGURATION V2 ........................................................................................................................................ 18
FIGURE 17: RTK GPS ...................................................................................................................................................................................... 19
FIGURE 18: LOW LEVEL ARCHITECTURE OF THE CYBUS ...................................................................................................................... 19
FIGURE 19: SYDEX SOFTWARE................................................................................................................................................................. 20
FIGURE 20: LOW-LEVEL <-> HIGH LEVEL ........................................................................................................................................... 21
FIGURE 21: THE WOODEN STATION WITH ITS HMI AND ITS LCD MONITOR ......................................................................................... 22
FIGURE 22: ONE OF THE FIVE ACCESS STATIONS .................................................................................................................................. 23
FIGURE 23: FINAL CONFIGURATION OF THE LASER-BASED SLAM ON THE CYBUS: TWO IBEO LUX LASER SENSORS AND A LOW COST INERTIAL MEASUREMENT UNIT WERE USED.................................................................................................................. 24
FIGURE 24: EXAMPLE OF PATHS PROVIDED BY THE ELEVATOR ROUTE PLANNER ................................................................................ 26
FIGURE 25: TRAJECTORY (RED LINE), PATH (GREEN LINE) AND VEHICLE (WHITE BOX) ........................................................................... 27
FIGURE 26: INITIAL TRAJECTORY (RED LINE), FINAL TRAJECTORY (BLUE LINE), PATH (GREEN LINE) AND COMMUNICATING VEHICLE (WHITE BOX) .................................................................................................................. 29
FIGURE 27: ARCHITECTURE OF A TRAJECTORY PLANNING APPROACH FOR A GENERIC ROBOT ............................................................. 29
FIGURE 28: 4G CUBE ...................................................................................................................................................................................... 31
FIGURE 29: COMMUNICATION EVOLUTION ........................................................................................................................................... 31
FIGURE 30: LA ROCHELLE CITYMOBIL MESH NETWORK ........................................................................................................................ 33
FIGURE 31: VMS ARCHITECTURE ............................................................................................................................................................... 34
FIGURE 32: SCREENSHOT OF THE VISUAL HMI INSIDE THE CYBUS .................................................................................................... 35
FIGURE 33: MESSAGE DISPLAY AND PLAYBACK BASED ON THE SPEED PROFILE .................................................................................. 36
1 Introduction

This deliverable aims at describing the architecture of the system used for the La Rochelle small demonstration of the European CityMobil project. The deliverable will focus on technical components and aspects. It will describe the functional and the operational constraints but does not expose the functioning results and does not tackle the legal issues or the juridical aspects.

After exposing the system’s constraints we will describe the general design of the system and then detail each of the subsystems developed for this purpose.

2 System Constraints & general design

The La Rochelle’s demonstration site physical and geographical conditions present several issues that have a strong impact into the design of the system. These conditions have evolved together with the design of the circuit from the moment when the decision of organizing the demonstration was taken in late 2008 until the moment in which the final configuration of the circuit was defined in November 2010.

In the initial version of the circuit, the cybercars should circulate on Avenue des Amériques, Michel Crépeau Avenue, Sénac de Meilhan street and Loup Marin street. However, due to a building construction on Loup Marin street, it was decided to abandon this section in June 2010. Finally, during a meeting with the technical services of the City of La Rochelle in November 2010 it was decided to exclude from the circuit the section in Sénac de Meilhan street, since it required several infrastructural adaptations to allow the circulation of cybercars, such as removing parking places and moving trees and street barriers. At this time it was also decided to move the station at the end of this section to the corner of Sénac de Meilhan street and Michel Crépeau Avenue. Figure 1 : shows the evolution of the cybercars circuit.
The main physical constraint for the circulation of the cybercars, which remains true throughout the different versions of the design of the circuit, is the width of the street in the longest section of the circuit in which the cybercars will have to drive, Avenue des Amériques, as shows the West-East street section view in Figure 2.

Figure 1: Evolution of the Cybercars circuit (red)
The width of the street being 5.3 m. creates the constraint that the vehicles cannot make U-turns without a heavy infrastructural intervention. On the other hand, the configuration of the sidewalks, 4.3 m. on the West side (including the greenery) and 1.35 m. on the East side results in the constraint that the cybercars stations can only be installed on the West side of the street. Since it has always been considered that the vehicles to be used are somewhat open, and given that the dominant winds come from the West, the City of La Rochelle had initially considered that the vehicles’ openings and doors should face East and be closed on the side overlooking at the West. For this to be true, the stations should be installed on the East side of the street. The fact that the stations could only be installed on the West side of Avenue des Ameriques had not been taken into consideration by the City of La Rochelle in the first and second versions of the circuit, and it only became evident when INRIA made a detailed study of the site. The constraint of the unavailability of U-turns combined with that of the implementation of the stations only on one side of the street resulted in a particular circulation scheme design. To allow the circulation of several vehicles, the vehicles have to drive forward in the North-South direction (on the Westernmost lane of Avenue des Ameriques), switch to the East lane when driving from South to North but return to the West lane to pick up or drop passengers. This cybercar circulation scheme poses risks as it is not a behaviour road users external to the cybercar can expect in a motor vehicle. Therefore, this fact has to be included as a Human-Machine Interface constraint for road users outside of the vehicle.

2.1 First and second versions of the cybercars circuit
In the first (until June 2010) and second (until November 2010) versions of the circuit, the circuit design was still based on La Rochelle’s requirements for the stations to be installed on the East side of Avenue des Ameriques. In these two versions of the circuit, it was decided to drive on Sénac de Meilhan street, which created an additional constraint. Unlike all the sections of the circuit, where two driving lanes are
available, in Sénac de Meilhan street there would be only one lane for cybercars. The City should enable this lane after several infrastructural modifications. Figure 3 shows the circulation scheme in the first version of the demonstration circuit.

**Figure 3 : Circulation scheme - first version of the demonstration circuit**

In the first version of the demonstration circuit, all the vehicles shared Sénac de Meilhan street, which had to be a two-way lane. In this case, the access to this section had to be controlled remotely by the vehicle management system, increasing the risk of deadlock in case of failure of the communications system and creating long waiting times to go to station e. Besides this, an inversion station [z] was required for vehicles to switch direction and dock with the door on the proper side of the station when going to/from station e. **Table 1** indicates the Vehicles trajectories per Origin/Destination pair (first version of the circuit).

**Table 1. Vehicles trajectories per Origin/Destination pair (first version of the circuit)**

<table>
<thead>
<tr>
<th>O/D</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>X</td>
<td>AB</td>
<td>ABC</td>
<td>ABCD</td>
<td>ABZZE</td>
</tr>
<tr>
<td>B</td>
<td>BA</td>
<td>X</td>
<td>BC</td>
<td>BCD</td>
<td>BZZE</td>
</tr>
<tr>
<td>C</td>
<td>CA</td>
<td>CB</td>
<td>X</td>
<td>CD</td>
<td>CDE or C</td>
</tr>
<tr>
<td>D</td>
<td>DA</td>
<td>DB</td>
<td>DC</td>
<td>X</td>
<td>DE</td>
</tr>
<tr>
<td>E</td>
<td>EZZA</td>
<td>EZZB</td>
<td>EC</td>
<td>ED</td>
<td>X</td>
</tr>
</tbody>
</table>
In the second version of the circuit (Figure 4), in which the section on Loup Marin street was abandoned, the need for the inversion station (Z) disappeared. Yet, the two-way lane on Sénac de Meilhan street and the risk of deadlock and long waiting times to go to station e remained true.

**Figure 4 : Circulation scheme - second versions of the demonstration circuit**

---

### 2.2 Final version of the cybercars circuit

When INRIA studied in detail the site since June 2009, it was concluded that the stations could only be installed on the West side of Avenue des Amériques. Under this constraint, it was no longer possible to keep station e in the Sénac de Meilhan street, since it would have been required to install the station on the East side of the cybercars circuit, therefore, on the street itself. Besides this, the City lacked the budget for the infrastructural works required to adapt this section for the circulation of the cybercars. Therefore, this section was eliminated and station e was moved to the corner of Sénac de Meilhan street and Michel Crépeau Avenue. This was the final modification made on the circulation scheme.
Once the cybercars circulation scheme was defined, it was necessary to define the whole system architecture, which included the management and following of the cybercars traffic and the possibility of calling the cybercars from the stations. The presence of mobile barriers on Avenue des Amériques created another constraint, since the vehicles should communicate with these barriers in order to pass across them. Figure 6 shows the final architecture of the system.
The components of the system are:

1. Vehicles;
2. Mobile barriers, on-board radio emitters and magnetic loops;
3. Stations, acting as communication relays;
4. Stand-alone communication relays;
5. Vehicle Management System (VMS);

Each of these components and the relationship between them will be described in chapter 3.

2.2.1 General design in demonstration Version 1

Due to the delay of an INRIA subcontractor (Induct) to provide the main demonstrator vehicle (CyberGo) in time for the start date of January 2011, only two INRIA prototype vehicles were available at the moment of the opening of the demonstration in May 2011. In order to have a backup in case of malfunction, only one of these vehicles ran each day. Therefore, it was not necessary to execute the cybercars circulation scheme, as it was designed for more than one vehicle, and a simplified version of this was carried out. In this simplified version, the vehicles did not require changing lanes to drive in the South – North direction, behaving exactly like a horizontal elevator, even though the “lane change” algorithm was already available. Figure 7: shows the simplified version of the demonstration circuit.
2.2.2 General design in demonstration Version 2

Due to the availability of only two INRIA prototype vehicles in the first version of the demonstration in May 2011, the driving scheme was not executed as defined in the final circulation scheme. The final circulation scheme was only executed in the extension of the demonstration, after the increase in the budget allocated by the CityMobil project to La Rochelle demonstration. This demonstration was executed in INRIA's Rocquencourt site, but in total respect of the circulation scheme designed for La Rochelle's site.

In order to execute the final circulation scheme, all the possible conflict scenarios between the cybercars were considered. Scenarios of conflict between the cybercars and external obstacles were considered but were not implemented. An example of these conflicts is the presence of a static obstacle on the cybercars’ path, which would require overtaking it. This kind of scenario was not implemented since it required a more complex set of sensors, in particular to determine the length of the obstacle.

Cybercars conflict scenarios

In order to execute the final circulation scheme with several vehicles, it is necessary to consider all the potential conflict scenarios in which cybercars interact. The conflict management in the final version of the demonstration is centralized by the Vehicle Management System (VMS). Cybercars are considered as “communicating obstacles”, whose nature and dimensions can be known by the other cybercars via the VMS, provided that the communication system works properly. It is important to remark that some scenarios
were not considered because of the fact that the demonstrated cybercars don't have motorized doors nor the presence of detectors on-board.

The following are the conflict scenarios implemented for the extension demonstration.

**Scenario 1**

Scenario 1 represents the most basic functionality necessary for the other scenarios to operate. In this scenario, all the vehicles must be capable of communicating their positions to the VMS, and the stations are capable of sending and receiving messages to and from the VMS in order to call a vehicle and to provide the station's Human-Machine Interface (HMI) with waiting time information. The diagram in Figure 8: represents scenario 1.

**Scenario 2**

In scenario 2, vehicle 2 is stopped at a station, and vehicle 1 was requested by a user to go to the same station. In this situation, vehicle 1 had to change lane to reach the selected station. In this case, the VMS will order vehicle 1 to stop until vehicle 2 has left the station. In case of using a vehicle with motorized doors and presence sensors, the VMS could determine if vehicle 2 is empty and order it to go to another station. Otherwise, it is necessary to wait until the user on board vehicle 2 selects a destination station. The diagram in Figure 9: represents scenario 2.
Scenario 3
Scenario 3 is quite similar to scenario 2. The only difference is that in this case, vehicle 2 did not have to change lane to reach the station occupied by vehicle 1. In this case, the VMS will order vehicle 2 to stop until vehicle 1 has left the station making a lane change manoeuvre. Just like in the previous scenario, in case of using a vehicle with motorized doors and presence sensors, the VMS could determine if vehicle 1 is empty and order it to go to another station. Otherwise, it is necessary to wait until the user on board vehicle 1 selects a destination station. The diagram in Figure 10 : represents scenario 3.

![Scenario 3 diagram](image)

Figure 10 : Scenario 3 diagram

Scenario 4
Scenario 4 presents the ability for the VMS to determine whether a vehicle crossing another vehicle, especially when stopped at a station, has to be stopped or not. In this case, vehicle 1 is driving passed the station where vehicle 2 is stopped. The VMS must be aware that vehicle 1 is not on the same lane as vehicle 2 and allow vehicle 2 to keep driving uninterruptedly to its destination. The diagram in Figure 11 : represents scenario 4.

![Scenario 4 diagram](image)

Figure 11 : Scenario 4 diagram

Scenario 5
Scenario 5 represents a conflict situation in which vehicles 1 and 2 drive at the same time towards the same destination station. In this case, the VMS will determine, based on the
proximity, which vehicle will have the priority. The closest vehicle will be allowed to continue and dock at the station, while the other vehicle will be ordered to stop. In case of a vehicle with presence detection, the presence of a passenger on board would be another criteria to give priority to a vehicle above the other. The diagram in Figure 12 : represents scenario 5.

**Figure 12 : Scenario 5 diagram**

Scenario 6
In scenario 6, vehicle 2 has to drive to a station and encounters a vehicle stopped in the same lane. In this case, the VMS will order vehicle 2 the authorization to overtake, which vehicle 2 will do based on the position of vehicle 1 (provided by the VMS). To avoid a conflictive situation, the VMS will order vehicle 1 to remain stopped until vehicle 2 has finalized the overtaking manoeuvre. The diagram in Figure 13 : represents scenario 6.

**Figure 13 : Scenario 6 diagram**

3 Subsystems design

3.1 Vehicle architecture (V1 & 2)
The CyBus is based on a regular Yamaha golf cart (model number G17-E) that has been completely modified by INRIA and Yamaha at the early stages of the 2000s. The base of the
mechanical parts (rolling chassis, propulsion assembly, steering assembly …) were kept and the bodywork was completely re-designed in order to create a new concept of autonomous vehicles. In addition, all the electronics and cabling were modified to accomplish at this time some specific demonstrations using wire-guidance technology.

Due to the delay of the CyberGo, it was then decided to use 2 CyBus vehicles for the demonstration. The rolling chassis, the motors, their amplifiers and the bodywork were kept and refurbished, but it was necessary to re-design all the electronic side of the vehicles in order to achieve the desired goals (for maintainability, reliability and safety reasons). New sensors and computer systems were also integrated into the vehicle to use the latest available technology.

Vehicle and main sensors evolution since the beginning of the project:

![Figure 14: Evolution since the beginning](image)

At the beginning localisation was only a visual line following and the LIDAR was used only for obstacle detection, for a short period we used RTK GPS but after the Finland showcase this technique was decided not reliable for the next showcases. At the end of the project we finally used LIDAR (as well as a GPS as a help) to perform both tasks: localization and obstacle detection.

### 3.1.1 Sensors (V1 & V2)

In Version 1 of the demonstration, a minimal set of sensors was integrated into the vehicles to achieve an accurate localization and safe obstacles detection. This first set of sensors was basically acceptable, but the running of the demonstration highlighted some weakness of the system. That is the reason why a second set of devices were added in Version 2 to provide a more robust localization needed for the second part of the event.

V1:
The main devices for localization and obstacle detection were 2 Ibeo Alasca XT laser sensors. They are located at the front and at the rear end of the CyBus, at a height of 80cm from the ground. The Alasca is a 4-layers laser-based range finding device, which measures the distances to objects in the surroundings of the sensor. The coverage of the beams is around 180° and the maximum distance that can be measured is around 200m. Both sensors are connected to an embedded computer (ECU) that achieves the fusion of the measurements and send them through an Ethernet connection at a frequency of 12.5Hz.

The localization process is also using data from an IMU (Inertial Motion Unit), which is located on the rear axle of the vehicle. The IMU used onboard the CyBus is the IMU440CA from Crossbow. This device is an inertial system that utilizes MEMS-based gyroscopes and accelerometers to provide inertial measurements in the three dimensions at a frequency of 100Hz. Due to the kind of information that the IMU is providing, it is mainly used to improve the time to time relative position and the orientation changes of the vehicle (Extended Kalman Filter).

The CyBus has also a "low-cost" GPS (a Garmin GPS18), which is located at the back of the roof. This sensor can provide a rough estimation of the position, because its accuracy is lower than 15m. Even if the sensor is not used for this specific demonstration, it is available and can be used by the system to check the SLAM\(^1\) based localisation, or to solve ambiguities.

Another device that's has not been really used for this Version 1 but was available on the vehicle, is an Axis 215 PTZ camera that was fitted under the roof at the rear of the CyBus. It's a 360° IP camera that is mainly used for video surveillance purposes. It can be remotely activated and monitored by a central station, but as an operator was always present in the vehicle, it was not really necessary to use it. In any case, it has a clear deterrent potential.

\[^1\text{SLAM} : \text{Simultaneous Localization And Mapping}\]
The running of the Version 1 of the demonstration highlighted some small problems that were not experimented before. These problems were mainly disturbing the localization and obstacle detection processes. The main reasons for this were: lack of reliability from the Ibeo ECU, lack of performance from the 2 laser sensors and a non-appropriate position of these 2 sensors. That is the reason why other sensors were considered to improve the system.

The Sick LMS511 is a single layer laser-based range finding device. The coverage of the beam is around 170° and the scan frequency goes from 25Hz up to 100Hz. For practical and logistical reasons, it was not possible to integrate them on the CyBus, but it is planned to do it as soon as possible to replace the two Alasca XT. That will make the obstacle detection module much more robust and more reliable.

The Ibeo Lux is another laser device which is much more compact and lighter than the LMS511. Like the Alasca XT, it provides 4 layers data but on a narrower range (between 80° and 100°). The scan frequency goes from 12.5Hz up to 50Hz. Like the other ones, the maximum range of the sensor is 200m. They have been integrated on the top of the roof of the CyBus (around 2m height) at the front and the rear end of the vehicle. With this configuration, the laser beams are not affected by the presence of people directly around the car. That makes the localization process much more robust than with the previous configuration, even with a smaller range covered.

In order to avoid any problem with the localization, a RTK (Real Time Kinematiks) GPS has also been integrated into the vehicle. The goal of this device was to replace the 2 laser sensors in case of any doubt concerning the localization. The GPS used for doing this job is an Ashtech Z-Xtreme running at a frequency of 10Hz. Practically, we never had to use the data provided by the GPS.
3.1.2 Low Level architecture (V1 & V2)

CyBus Low Level Architecture Description:

The CyBus is a small urban vehicle powered by a 72V lead-battery pack and propelled up to 15km/h by a 5.625HP separately excited electric motor. The vehicle steering system has been modified to become fully drive-by-wire, supplying an absolute encoder to measure steering lock position and an assisting 12V power amplifier for the control of the steering motor. Vehicle instantaneous speed and odometric measurements are performed using two magnetic sensors mounted with 90° phase difference on the vehicle’s back axle. Standard automotive lighting system in conjunction with a horn and two emergency rotating lights enable low-level basic interactions with other road-users.

CyBus Electronic Architecture:

Two prototypes of controllers have then been produced to allow complete and precise control of the vehicle steering and propelling systems respectively. The architecture of the controllers has been designed to be as generic as possible made up a mother-board based on a low-cost 16-bit microcontroller with digital signal processing additional functions supplied by daughter-boards holding all peripheral physical interfaces: one for the steering system’s sensors & actuators and another doing the same for the propelling system. Two industrial
CAN bus interfaces, located on each mother-board, enable dual 1Mbps real-time data-packet exchanges to take place between controllers.

CyBus Alternate Emergency System – V1:
Stating that the vehicles are still in experimental configuration, an alternate independent emergency system has been added to the vehicle’s architecture, allowing an external human operator to stop with no delay vehicle’s low-level operation. When activated or when the vehicle is out of reach, electromechanical braking system is engaged and electric power sources are cut off. Once the system fault recognized and corrected, the vehicle can be restarted manually. A certified commercial system is used, constituted by a wireless remote control associated with an unique receiver communicating on a proprietary encrypted channel.

CyBus Software Development – V1:

![Figure 19: Syndex software](image)

CyBus electronic architecture is distributed by design: having controllers close to vehicle’s sensors and actuators limits wiring complexity and improves electromagnetic compatibility; system modularity is increased allowing future low-level architecture extensions to be envisaged. On the other side, complexity is reported on software development to correctly manage inherently distributed functionalities of the whole system. With this in mind, long term collaboration with INRIA project team AOSTE has been maintained, promoting the use of the SynDEx EDI to implement the low-level control firmware of the CyBus. SynDEx EDI is dedicated to rapid prototyping and hardware/software co-design of complex real-time distributed systems. Once proved formal assumptions on whole system correct behaviour, safe deadlock-free code is automatically generated allowing the controller network to be re-programmed in-the-loop. This functionality considerably improves system development life-cycle reducing together with product time-to-market.

CyBus Motor Control Loops – V1:
Two real-time control loops have been implemented within a 100ms cycle period. Position control of the steering angle is done by a PID algorithm running on the steering lock controller. Another PID algorithm running on the propelling controller regulates the speed of the vehicle. Each of them runs in parallel and independently from the other. The control algorithm’s gain coefficient parameters can be modified in run-time allowing quick on-the-road reconfiguration of vehicle’s current behavior.

CyBus Motor Control Loops - V2
During the La Rochelle demonstration, tedious oscillations were noticed when the CyBus was moving on long straight trajectories. After numerous trials and experiments, it has been shown that the low-level control algorithm was not responsive enough and that the current implementation was error-prone. Once this incorrect behavior demonstrated, control system has been reviewed and re-configured in less than a day thanks to the highly modular design of the whole hardware/software system.

**CyBus Low-level To High-Level Communications - V1**

![Diagram](image)

**Figure 20 : Low-Level <-> High Level**

High-level computer cluster communicates in real-time with the low-level controller network using a single CAN bus interface. A concise communication protocol has been defined allowing high-level system to set vehicle speed and steering angle references in a single CAN frame. Low-level then returns current vehicle odometric counter and steering lock position. As explained, parameters of each control algorithm can also be configured during run-time through three other CAN bus messages. Due to this implementation, high-level and low-level systems are fully independent of each other enabling high-level system to be modified or replaced without intervention on the low-level architecture.

### 3.1.3 High Level architecture (V1 &V2)

The high level architecture was mainly based on an embedded single computer. With the experience of the Version 1 of the demonstration, it was decided to add another computer dedicated to the localization.
With the experience of the Version 1 of the demonstration, it was decided to add another computer dedicated to the localization: there was no direct real problem with this module, but we discovered that the computer could be overloaded in case of unexpected events. These overloads were causing perturbations to the control process resulting in some dangerous delays. As a starting point, we decided to use exactly the same computer that was proven to be reliable enough. The 2 computers are connected with a simple Ethernet line and exchange data within the RTMaps environment.

3.2 Stations

The stations used for the La Rochelle small demos are wooden platforms specially designed for the demonstration purpose. Each station is made of three components:

1. An access platform made of a planar basis and a ramp. This is a wooden platform with a basic design and is equipped with wood stud to protect against accidental drops.

2. A wooden station equipped with a touch screen holding the man-machine interface and the embedded communications components. The station also carries a notice paper containing the user information. This is the "instruction" as well as the "usage conditions".

3. A wireless Wi-Fi network relay with a 8dB antenna put at 4 meters high.

Figure 21: The wooden station with its HMI and its LCD monitor.
3.2.1 Accessibility

As it can be seen on Figure 22, the stations were designed in order to enable the access to elderly people and to wheeled chairs. A ramp of nearly 4° allows moderate inclination; this ramp is equipped with a basic anti-skid material to avoid slippery surface. Two wooden amounts are also available to prevent the users from falling.

Figure 22: One of the five access stations
3.2.2 Power supply

In order to provide the station – especially the LCD touchscreen monitor – with electricity a power supply had to be present. Because no nearby electricity access was present it was decided to use ordinary rechargeable batteries in La Rochelle I demo. However for La Rochelle II a nearby electric outlet was available.

3.3 Localization V1 - V2

Localisation uses mainly the LIDAR because GPS had caused us a lot of trouble at the beginning of Citymobil. Since the Trondheim (Norway) showcase (2009), this technique had been improved to be robust in urban environments. For V1 and V2, we used Laser based SLAM for providing an accurate Map, and it use a Laser based Mapping Localization Algorithm which is fused With IMU Data Using EKF for Improving Localization precision.

The Simultaneous Localization and Mapping (SLAM) problem is one of the fundamental issues in mobile robotics. SLAM requires a mobile robot to increasingly build a consistent map of an unknown environment using on-board sensors while concurrently localizing itself relative to this map. A solution to the SLAM problem has been regarded as an important prerequisite for autonomous robots as it would provide the means to make a robot truly autonomous under unknown environments.

We have developed for CityMobil project a high-efficient reliable SLAM for outdoor application and shown it working on a challenging large-scale dynamic environment in real time. Our SLAM algorithm is able to incrementally build a consistent map for large-scale outdoor environment and close the loop.

3.4 High level control

This subsection describes the RTMaps motion planning components used in the navigation task of the vehicles used in the CityMobil European projects.

The motion planning components are divided into 2 groups:

- Global planning: route planning;
3.4.1 Global Planning

Route planning

Purpose
The route planning consists in determining the path to go from the current vehicle position to a requested destination. This destination can be chosen by a user in the vehicle using the internal HMI, or externally set at a station booth or a mobile device. The vehicle itself behaves like a horizontal elevator serving the several destination requests reason by which the component was named ElevatorRoutePlanner.

Implementation
The elevator route planner component determines the path to be followed by the vehicle when he receives a new destination request. This is done by solving the travelling salesman problem [RD001] using the nearest neighbour algorithm. By doing so, it determines, at any time, the next destination to be reached by the vehicle.

Destinations are handled much like an elevator does except the vehicle does not always travel in the same direction while there are remaining request in that same direction. This means that the vehicle may change direction if there are requests in the opposite direction that are closer.

The route planner also takes into account the deceleration of the vehicle and will only provide a path to a destination if it has enough time to stop at that destination. The vehicle will not stop at a destination if the request was done too late – the vehicle does not have enough time to stop smoothly at the station – much like when a bus driver does not stop when someone pushed the stop button very close to a station. The difference is that, in the automated vehicle, the destination is memorized and will be fulfilled as soon as possible – the vehicle will reverse direction and will go back to the requested destination – as for the bus will probably stop in the next station or will not stop at all.

The elevator route planner was implemented taking into consideration that the destinations (typically stations) were always situated on one same side of the road and that the vehicle should respect the traffic direction of the lanes. This assumption was done taking into account the topology of the demonstration sites of La Rochelle and INRIA Rocquencourt. In both sites the roads are composed of two lanes with opposite traffic directions and there is usually very limited space in one side of the road to implant new infrastructure like the stations and access ramps. For this reason, contrary to a traditional horizontal elevator that follows always the same path when travelling forward or backward, the elevator route planner was implemented in a way that the vehicle can follow 2 different paths depending on the direction of the movement: forward or backward. The initial path (recorded waypoints) follows a lane (green line) and the second path (red line) is generated from the initial path. This is done by bending the initial path to follow the second lane and perform the necessary lane changes to leave/arrive at the corresponding station. The following image illustrates the described principle:
Inputs
The elevator route planner component has three inputs:

- **Status**: confirmation of the destination by the user (in vehicle). This input is only necessary because our vehicle do not have door sensors installed to indicate that there is no one entering or exiting the vehicle. A confirmation is then necessary by a human action in the vehicle for safety.
- **Vehicle state**: the vehicle pose (position and heading) and velocity;
- **Destination**: position (point x,y);

Outputs
The elevator route planner component has four outputs:

- **Path**: a set of waypoints to go from the current vehicle position to a destination
- **Number of points**: the total number of waypoints existing in the path
- **Next destination**: information about the next destination like its position, distance, and time to arrival.
- **Blinkers**: the vehicle blinkers command. For example: when turning right the right blinkers are activated and when the vehicle is moving at a very reduced speed or stopped (due to an obstacle for example) the four blinkers are on.

Properties
The elevator route planner component properties are set directly at the component properties dialog box:

- **Debug report**: enables/disables the writing of formatted data (string) to the standard output.
- **Vehicle name**;
- **Logs path**: the directory where the trips logs will be recorded;
- **Popup file**: the graphical interface that will be launched when the status input is enabled – allows the confirmation of the destination by the user.
- **Path file**: the recorded waypoints that cover all the stations network;
- **Number of destination inputs**: total number of sources that can provide destination requests to the route planner;
- **Vehicle length**;
- **Maximum steering wheels angle**;
- **Default road speed limit**;
- **Vehicle deceleration**;
- **Resting time**: time to wait before providing a new path;

Figure 24 : Example of paths provided by the elevator route planner
Blinker velocity threshold: the maximum velocity to which the fours blinkers will be turned on;
Bend route: enables/disables the bend of the path (subset of the recorded waypoints) to change lane when moving in the direction opposite to the sense of the recorded waypoint;
Bend width: the lateral distance between lanes;
Bend length: the longitudinal distance that the vehicles should perform to go from one lane to another;
Bend side: specifies if the lane change should be done to the left or to the right;
Bend direction: specifies if the lane change should be done when moving forward or when moving backward.

3.4.2 Local Planning

Trajectory planning

Purpose
The trajectory planning provides a time parameterized reference (i.e., a geometric path with an associated timing law) for the vehicle to follow.

Implementation
The trajectory planner implementation is very simple: the trajectory generated is just a portion of the geometrical path provided by the route planner to which is added a speed profile taking into account the current vehicle speed and the maximum allowed speed in that path. See the following image:

Figure 25 : Trajectory (red line), path (green line) and vehicle (white box)

The following steps describe how the trajectory is generated:
1. reading of the reference path provided by the route planner;
2. localization of the vehicle in the reference path;
3. extract a partial path (portion of the reference path) until a certain maximum distance (local sensors range);
4. determine if the vehicle is to move forward or move backward;
5. add speed profile and time to the partial path taking into account vehicle movement direction and speed limits.

The speed in the end of the trajectory is always zero (brick wall stopping criteria).

In the current implementation of the trajectory planning component the collision with obstacles is not being managed since it is already handled by the speed guard component.

Inputs
The trajectory planning component has three inputs:
• Vehicle state: the current vehicle pose (position and heading), steering angle, and speed;
• Path: the geometric path produced by the route planner;
• Number of path points

Outputs

The trajectory planning component has one output:

• Trajectory: each trajectory element is a pose in time – x, y, heading, speed, time

Properties

The trajectory planning component has the following properties:

• Debug report: enables/disables the writing of formatted data (string) to the standard output.
• Maximum velocity: maximum vehicle velocity;
• Minimum velocity: defines the minimum velocity command necessary for the vehicle to move;
• Maximum acceleration;
• Maximum deceleration;
• Security distance: stopping distance from last trajectory point;
• Output delay: specifies the minimum computation time of the component.

3.4.3 Trajectory coordination

Purpose

The trajectory coordination allows the negotiation of trajectories between communicating vehicles. It provides the final time parameterized reference (i.e., a geometric path with an associated timing law) for the vehicle to follow.

Implementation

The trajectory coordinator relies on a centralized control centre and therefore simplifies greatly its implementation. This component has two main functions:

• update the speed profile: the speed profile of the trajectory provided by the planning component is updated accordingly with the received reference speed from the control centre. For example: if the reference speed is set to zero, the new speed profile will make the vehicle stop, even if the initial trajectory speed profile specifies that the vehicle should accelerate.
• overtake vehicle: the initial trajectory points are modified (bended) to overtake a static communicating vehicle.

The final trajectory will be eventually a modification of the initial trajectory depending on the decisions taken at the centralized control centre.
In order to perform the trajectory deformation it is necessary to take into account the front vehicle dimensions so that the ego vehicle can respect safe margins and avoid any possible collision. Communications between the vehicles provide these parameters to the ego vehicle.

In order to decide the way and amount of deformations while respecting the non-holonomic vehicle constraints a new path planning approach was designed based on the elastic bands well known technique.

**Elastic bands** are proposed to close the gap between global path planning and real-time sensor-based robot control.

An elastic band is a **deformable collision-free path**. Subjected to artificial forces, the elastic band deforms in real time to a short and smooth path that maintains clearance from the obstacles. While providing a tight connection between the robot and its environment, the elastic band preserves the global nature of the planned path.

Another advantage of this technique is that it is adapted to holonomic and non-holonomic vehicles. The general architecture of a trajectory planning approach combining a global planning and a reactive planning can be illustrated in Figure 27.

**Figure 27**: Architecture of a trajectory planning approach for a generic robot.
Inputs
The trajectory coordinator component has four inputs:

- Vehicle state: the current vehicle pose (position and heading), steering angle, and speed;
- Trajectory: the initial trajectory produced by the trajectory planner;
- Maximum velocity: provided by the centralized control centre;
- Overtake: overtaking order (yes/no) with the communicating vehicle position.

Outputs
The trajectory planning component has one output:

- Trajectory: each trajectory element is a pose in time – x, y, heading, speed, time.

Properties
The trajectory planning component has the following properties:

- Debug report: enables/disables the writing of formatted data (string) to the standard output.
- Deceleration: vehicle deceleration;
- Radius: the radius of the circle that encloses the communicating obstacle;
- Overtaking distance: the distance from which the overtaking is started/finished.

3.5 Safety subsystems (Obstacle detection)

The safety subsystem designated here is obviously the obstacle detection module. This system aims at ensuring safety for the vehicle and the environmental elements: the obstacle. These are pedestrians, other Cybuses, vehicles but also the static environment.

In order to avoid the collisions with the static and dynamic obstacles it is necessary to have two functionalities: obstacle detection and obstacle handling (stop on obstacle and avoid the obstacle).

The obstacle avoidance is a planning problem which consists in deforming the existing path or in elaborating a new trajectory in order to avoid the collision with an eventual obstacle. This function is described in section 3.4.3.

As for “obstacle detection”, this function is composed of two sub-systems that are decorrelated. The first sub-system is embedded in the SLAM component which is responsible of the environment mapping and the vehicle localization. Indeed, in order to achieve this performance the SLAM has to match the local perception models built with the laser data with the registered map. This enables the classification of real data into: static obstacle and mobile objects. Static obstacles are compared to the registered data to determine if they belong to the known environment or if they correspond to static obstacles. The mobile objects are surely obstacles the vehicle has to deal with. The position and speed of these obstacles can be determined by the SLAM in real time. A filtering procedure enables to make trajectory predictions in order to identify a collision situation.

The second sub-system is a part of a basic anti-collision system which aim is to detect in real time a very close obstacle on the trajectory of the vehicle.

The first detection system uses the two laser scanners that are on the top of the Cybus while the second system uses the laser scanners that are at the knee level close to the bumpers. The sensors on the top have to be more accurate (high angular resolution) and with a high
depth range since they are responsible of performing accurate localization. The lower sensors are mid-range homologated sensors but have a high frame rate since they are dedicated to reactive detections. As seen in Figure 16, upper sensors are IBE0 ALASCA that were used for the La Rochelle I demo while for La Rochelle II demo held at INRIA those sensors were replaced by SICK LMS 511 sensors.

3.6 Communications V1 – V2
Since the beginning of the project we have developed a specific communication that use today standardized modules. A dedicated hardware was designed to help the deployment of the communication on the demo’s site.

![Figure 28: 4G Cube](image)

This Hardware is a Linux embedded pc with 2 mini pci free slot for adding different communication cards. Since Cybercars 2, we had developed a service discovery based architecture to make vehicle communicate and to simplify network based development for ITS. We started for Cybercars2 and the beginning of CityMobil with homemade solution's, “scope server framework” + OLSR + IPv4 to finally converge to a fully IPv6 system based on “cables framework” + BATMAN mesh network. And on the top of that a VMS (Vehicle management system) was coupled to manage all our automated vehicles. Also the radio frequency of 5Ghz was used in order to be more close to the future standard 802.11p, a dedicated communication system for ITS. New antennas and WIFI card have been used for the last Showcases.

![Figure 29: Communication évolution.](image)
Cables:
Originally developed for the automotive industry (AROS project with VALEO), Cables has proved itself to be very useful for developing distributed network based applications.

- **Cables** is very lightweight: 400kB of binaries approximately. It's just a library written in plain C.
- It's also a full featured web server, supporting HTTP 1.1: keep-alive connections, pipelining and data compression. Indeed, in Cables, each component is a web-server. **Cables** also supports AJAX, which eases the development of nice web based graphical interfaces.
- **Cables** also proposes its own proprietary protocol, which is used when communicating between Cables components. This allows to establish subscribe / push communications, but also to use UDP transport, which maximizes data streaming bandwidth.
- It can run on any 32 or 64-bits platform, including Windows, Linux, MacOS X, iOS (iPhone and iPad) or Android.

Mesh Network:
Our network is self organised and it use layer 2 based mesh algorithm:
B.A.T.M.A.N. (better approach to mobile ad-hoc networking) is a routing protocol for multi-hop ad-hoc mesh networks.

Most other wireless routing protocol implementations (e.g. the batman daemon) operate on layer 3 which means they exchange routing information by sending UDP packets and bring their routing decision into effect by manipulating the kernel routing table. Batman operates entirely on ISO/OSI Layer 2 - not only the routing information is transported using raw Ethernet frames but also the data traffic is handled by batman-adv. It encapsulates and forwards all traffic until it reaches the destination, hence emulating a virtual network switch of all nodes participating. Therefore all nodes appear to be link local and are unaware of the network's topology as well as unaffected by any network changes.

This design bears some interesting characteristics:
- network-layer agnostic - you can run whatever you wish on top of batman-adv: IPv4, IPv6, DHCP, IPX,…
- nodes can participate in a mesh without having an IP
- easy integration of non-mesh (mobile) clients (no manual HNA fiddling required)
- roaming of non-mesh clients
- optimizing the data flow through the mesh (e.g. interface alternating, multicast, forward error correction, etc)
- running protocols relying on broadcast/multicast over the mesh and non-mesh clients (Windows neighborhood, mDNS, streaming, etc)
3.7 VMS

A vehicle management system based on the communication architecture has been developed to achieve the problem of managing vehicle reservation and safety in critical sections.
The VMS have a state machine that managed all vehicles demands, and it is distributed to be run anywhere on the network. The database is standardised (SQL LITE) and it can store any kind of vehicle information. The VMS can take control of every car on the path to achieve dedicated manoeuvres to make the system run smoothly or to avoid collisions.

3.8 HMI
Because of the unavailability of motorized doors on the demonstrated INRIA's cybercars, the visual and audio Human-Machine Interfaces (HMI) are extremely important for user’s safety during the public demonstration in La Rochelle. In case of a problem causing the vehicle to stop, it is necessary to prevent users from alighting the vehicle, since the vehicle could start moving suddenly again. However, as mentioned in chapter 2, the circulation scheme which includes vehicles driving backwards and changing lanes also required the design of an external HMI for road users outside of the vehicle. However, this was limited by the fact that INRIA's cybercars have no external HMI devices other than headlights, brake lights and blinkers. Therefore, the lights, plus an audio warning device, were the only external HMI hardware available.

3.8.1 On-Board HMI
The main purposes of the on-board HMI are to allow users to select a destination and to provide users with visual and audio messages in function of the status of the vehicle (starting trip, stopping at the station, stopping and not at a station). As mentioned above, some messages are essential since they provide warning messages to prevent users from alighting the vehicle in a position different than a station. The visual HMI evolved throughout the different showcases that INRIA carried out in the framework of the CityMobil project but some extra features were added for the specific operating conditions in La Rochelle. One of these new functions was the ability to provide status information to the operators, so to ease the diagnosis of any problem that might occur while INRIA staff was not present.

Visual and audio HMI
The visual HMI is a separate application written in C programming language that receives information from the vehicle status. It can be configured for different network schemes, languages, etc. through an XML configuration file. In this configuration file can be included the number of stations required, the position of the buttons to be displayed in the visual HMI, the actual positions of the stations, and the messages to be displayed in different languages.
The HMI playbacks and displays two kinds of messages depending on the vehicle’s status (position, speed, distance to destination): Warning messages (WM) and Destination messages (DM).

**Figure 32 : Screenshot of the visual HMI inside the Cybus**

The visual HMI is divided in 5 zones, each with a particular function. The visual HMI zones (from top to bottom) are the following:

1. **Language selection**: This area contains the buttons to change language and lock the HMI. The latter was added for La Rochelle demonstration, so that the operator could lock the HMI (and therefore the vehicle) in case he had to leave it empty.

2. **Destination selection**: The map of the served site is represented in this area by means of lines and dots. The dots represent the stations. They can be pressed to select the station and they use a particular colour code to indicate the current station (blue), to indicate whether a station is not selected (white), if a station is selected and is the next stop (red) or if a station has been selected but is not the next station (orange).

3. **Destination message area**: The messages about the next destination (time to arrival) are displayed in this area.

4. **Warning messages area**: In this area are displayed warning messages, which are also played back as audio. The purpose of this area is to provide warning messages to handicapped users.

5. **Status information area**: This area contains a series of indicators of the vehicle’s subsystems status. This area is mainly intended to provide the cybercars’ operators in La Rochelle with a quick diagnosis about the status of each subsystem in case of breakdown, which would otherwise take too long.

A diagram representing the speed profile of a trip was made in order to define the situations in which messages are played back and displayed in the visual HMI. Figure 33 indicates the Message display and playback based on the speed profile of the vehicle.
The most critical messages, created specifically for La Rochelle demonstration, are WM3 and WM4 (WM = Warning Message), which are played back when the vehicle stops in a position different than a station. In INRIA's cybercars, which have manually operated doors and don’t have door sensors to feedback to the vehicle whether the doors are open or not, these messages prevent users from stepping off the vehicle while it has not reached its destination. This could lead to a risk situation since the vehicle could start moving with passengers alighting. As mentioned before, these messages would not be required in a vehicle with motorized doors. The detail of Warning and Destination messages and whether audio messages are played back is presented in Table 2.

### Table 2. Destination and Warning messages table.

<table>
<thead>
<tr>
<th>Code</th>
<th>Short name</th>
<th>Actual text</th>
<th>Visual</th>
<th>Audio</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM1</td>
<td>Next stop ___ in ___ minutes</td>
<td>Next destination: ___. Arrival in less than ___ minutes.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DM2</td>
<td>Vehicle available</td>
<td>Vehicle available. Waiting for new destination</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>WM1</td>
<td>Imminent departure</td>
<td>Imminent departure, DO NOT LEAVE THE VEHICLE!</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>WM2</td>
<td>Vehicle accelerating</td>
<td>Attention, the vehicle is accelerating. Please hold the handles.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>WM3</td>
<td>Vehicle slowing down</td>
<td>Attention, the vehicle is decelerating. Please hold on tight.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>WM4</td>
<td>Vehicle is stopped for a technical reason. DO NOT OPEN THE DOORS!!!</td>
<td>The vehicle has not stopped at the station. PLEASE DO NOT OPEN THE DOORS!!!</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>WM5</td>
<td>Arriving to destination</td>
<td>We are approaching the next stop. Please do not open the doors until the vehicle has come to a complete stop.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>WM6</td>
<td>Arrived to destination</td>
<td>We have reached the destination.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### 3.8.2 External road users HMI

As mentioned in chapter 2, the physical constraints of the demonstration site constrained INRIA to design a circulation scheme in which vehicles must drive forward on one lane, and change lanes to drive backwards, so that multiple vehicles can use the site. It was thus essential to prevent other road users about the lane change manoeuvre, since this is not a regular manoeuvre in a motor vehicle. Therefore, it was also required to design an external HMI for road users outside of the vehicle.

The requirements for this external HMI are the following:
1. Turn on 2 white lights (headlights) in the front of the vehicle (relative to the vehicle's direction of movement).

2. Turn on red lights in the back of the vehicle (relative to the vehicle's direction of movement).

3. Turn on one side's blinkers before the vehicle makes a turn or before it changes lanes (back and front).

4. Turn on both side's blinkers before and when the vehicle stops (back and front).

5. Increase the red light intensity (relative to the vehicle's direction of movement) in the back of the vehicle when the vehicle brakes.

6. Provide a yellow flashing light (road work vehicle) located at a height visible from any direction to external road users (between 1.2 and 3 m).

7. Operate a yellow flashing light (road work vehicle) 5 seconds before the vehicle starts moving and permanently while the vehicle is moving.

Unfortunately, due to the design of INRIA's cybercars, requirements 1 and 2 could not be fulfilled since these vehicles only have headlights in the front and brake lights in the back.

4 Conclusions and future work

The service provided at Lo Rochelle from May to July 2011 provided interesting feedbacks regarding the improvements to achieve in order to have a better functional system.

From a technical point of view we have identified a number of perspectives.

4.1 Technical perspectives

Extended perception:
- In order to handle complex situations it is necessary to identify these situations and be able to interpret adequately the environing context. For example, detecting a barricade or a temporary working zone is necessary in order to identify a bottleneck situation. Thus, more intelligent overtaking can be performed using extended perception. For example, overtaking of stopped aligned vehicles or crossing pedestrians enables the system to identify tricky situations where the vehicle is not supposed to be blocked.

- Introduction of vision technology: in order to ensure redundancy but also to recognize elements of the environments that cannot be identified by the laser sensors alone, vision can be of great help. For example, vision, can not only detect obstacles but also classify the nature of the objects. This has an implication on the way the vehicle would deal with the obstacles (e.g. avoidance strategy). The fusion of laser sensors and vision will also be very helpful in better estimating the shape parameters of the objects.

Deployment of advanced communications:
New wireless communication boxes under design. These are adopting the newest standards and norms such as 802.11p under 5GHz radio frequency and IPv6. The 4G cube described in section 3.6 is a first version of this dedicated device.

**Advanced docking:**
In order to allow safe access to the Cybus stations especially for elderly and handicapped or wheel chaired people, it is necessary that perform a very accurate docking. The current system uses the accuracy of the localization obtained by the SLAM system. We are willing to improve the docking by the use of different means:

- Achieve sensor-based docking : e.g. magnets
- GPS based docking using a centimetric RTK version
- Vision based docking using visual features or beacons

**Compliance and fault tolerance:**
It is mandatory that any system dealing with human safety must be fault tolerant. One way to deal with this issue is to build redundant systems like in the aeronautics. Duplicating the perception system is a step forward but this approach should be generalized to other subsystems especially those dealing with safety.

**Autonomy vs. VMS based architecture:**
Today's system is based on a centralized architecture where the VMS plays the role of the decision system. The VMS communicates with all the operating vehicles and decides in case of conflictual situations and bottlenecks. However in some tricky situations it is necessary that the vehicle rely on its own decisional system especially in case of a VMS shortage or in case of a reactive navigation (e.g. reactive obstacle avoidance or overtaking).

### 4.2 General perspectives

It is interesting to inform that, following the two La Rochelle small demonstrations, a permanent transportation service will be developed at INRIA in Rocquencourt based on the platforms developed for CityMobil. Besides being a technological demonstrator the system is intended to provide a true service to disabled people working on the site and to visitor and employees in general.

In France, new projects dealing with innovative large scale multimodal and clean transportation systems are launched and financed by public funds: Mobilité 2015 and SYSMO 2015 are the latest examples of French projects under submission and dealing with these issues.

In Europe, the CityMobil-2 has been submitted very recently. Besides extending the functionalities and principles of CityMobil project, its main objective is bring even closer such systems in the heart of the European cities and make large scale demonstrations / services using pre industrial platforms. Since the deployment of such systems is not only a technical issue CityMobil-2 will focus on legal issues at the European level, as a first step to legalize the similar transportation systems.