DRIVE: a reconfigurable testbed for advanced vehicular services and communications

Carolina Pinart
Telefónica I+D
c/ Emilio Vargas 6 28043 Madrid
+34 9133 74005
cpg@tid.es

Iván Lequerica
Telefónica I+D
c/ Emilio Vargas 6 28043 Madrid
+34 9133 74005
ilr@tid.es

Isaac Barona
Telefónica I+D
c/ Emilio Vargas 6 28043 Madrid
+34 9133 74005
ibarona@tid.es

Pilar Sanz
Telefónica I+D
c/ Emilio Vargas 6 28043 Madrid
+34 9133 74005
psv@tid.es

Daniel García
Telefónica I+D
Parque Tec. Boecillo (Valladolid)
+34 9833 67798
dgf@tid.es

Diego Sánchez-Aparisi
Telefónica I+D
c/ Emilio Vargas 6 28043 Madrid
+34 9133 74005
dsam@tid.es

ABSTRACT

This paper introduces DRIVE, a reconfigurable demonstrator that will allow for experimental validation of vehicular networking research, as well as proof of concept and realistic performance studies of advanced services for vehicular users. The testbed features an OSGi-based software platform that allows easy service development through bundles. DRIVE’s hardware platform is based on a carPC, wireless sensors and rooftop antennae and uses wireless and mobile technologies, e.g., WiFi, UMTS and Bluetooth. On top of these platforms, a hybrid communication node enables always-on, QoS-aware connectivity by using the Always Best Connected concept, and a service platform enables seamless car, driver and passenger services.

Categories and Subject Descriptors

C.2.1 (Wireless communication), H.5.2 (Interaction styles), J.0 (Computer Applications).

General Terms

Design, Experimentation, Human Factors, Verification.

Keywords

Vehicular networks, demonstrator, VANET, added-value services, wireless networks, software platform, carPC.

1. INTRODUCTION

Vehicular networks and services are becoming a reality, driven by safety requirements and by the investments of car manufacturers and Public Transport Authorities. Consequently, much research effort is being put into defining radio technologies (e.g., IEEE802.11p), network models and mechanisms (e.g., vehicular ad-hoc networks, IEEE802.21), and into developing services for the vehicular environment (e.g., eCall). As in most network environments, modelers, emulators, simulators and experimental implementations are the main tools used for testing these aspects.

Modelers represent phenomena as a set of mathematical equations, which enables the visualization of network applications or protocols' performance, as well as 'what-if' analysis. Simulators generate test conditions approximating actual or operational conditions, and rely on mathematical formulae to determine behavior. These analytical methods are limited by the mathematical complexity and the dimensionality of systems, which usually leads to simplifications and approximations that may increase the distance between predictions and real behavior. Emulators imitate the function of other systems, as by modifications to hardware, software, or network activity that allow the imitating system (emulator) to accept the same data, execute the same programs, and achieve the similar results as the imitated system.

The advantages of experimental implementations (i.e., testbeds) over the above-mentioned approaches are the gain of hands-on experience in close-to-real or real performance and behavioral issues, as well as favoring intuition shape practice. However, because of their associated cost, there are not many experimental studies of vehicular networks and services. In this context, this paper presents the DemonstRator for Intelligent Vehicular Environments (DRIVE), which is an experimental implementation featuring both advanced vehicular services and vehicular networking (i.e., intra-vehicle, vehicle-to-vehicle [V2V], vehicle-to-roadside [V2R] and vehicle-to-infrastructure [V2I] communications). Thanks to a modular, reconfigurable architecture, this testbed can integrate different radio interfaces, communications schemes and services easily and readily.

The remainder of the paper is organized as follows. Section 2 overviews the major vehicular testbeds found in the literature. In Section 3 we describe the architecture of the DRIVE testbed, both in software and hardware terms. Section 4 describes in-vehicle, V2V, V2R and V2I communications in DRIVE from the on-board and infrastructure perspectives. Section 5 overviews the services available in the testbed, which cover car, driver and passenger applications. In Section 6, we describe a usability study for multimodal human-machine interfaces applied to DRIVE. Finally, in Section 7 we summarize the key features of the DRIVE testbed.

TRIDENTCOM 2008, 17th – 20th Mar 2008, Innsbruck, Austria.

ISBN 978-963-9799-24-0
DOI 10.4108/weedev.2008.3141

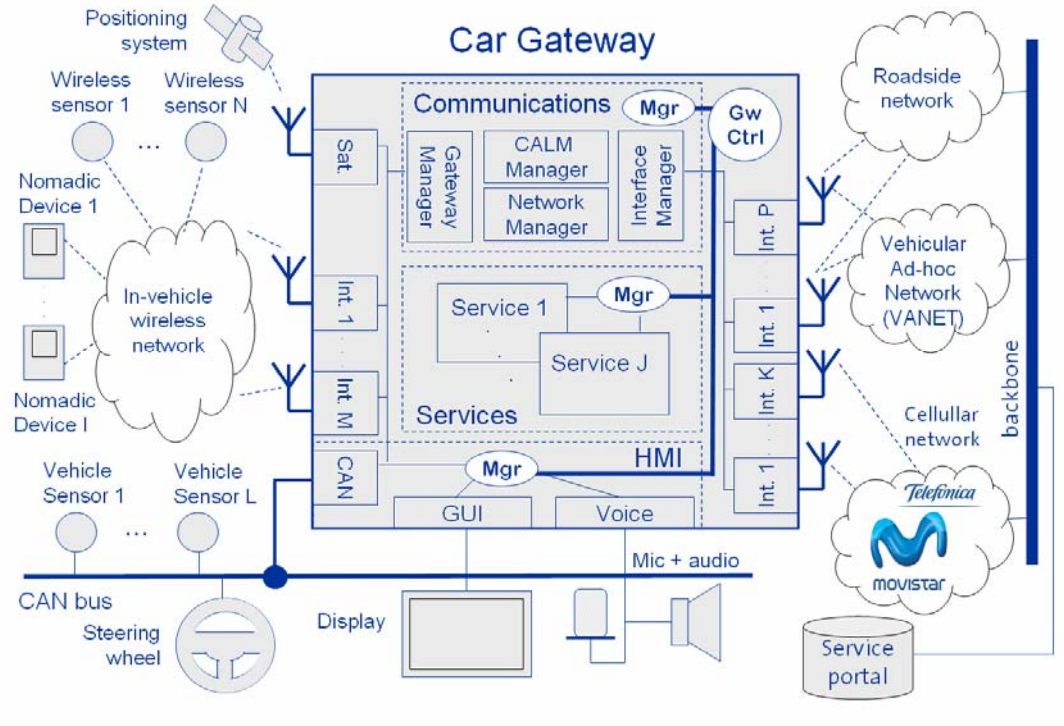

Figure 1. Architecture of the DRIVE testbed (on board vehicles)

2. RELATED WORK

Vehicular testbed implementations range from physical-layer demonstrators, such as Short-Range Communications (SRC) [1], to performance experimentation of applications in the vehicular environment, e.g., content sharing [2]. Since the DRIVE testbed is focused on network-layer aspects of communications and on advanced vehicular services, here we only report (not exhaustively) on those testbeds related to these issues, which can be categorized as vehicular ad-hoc network (VANET) testbeds and vehicular service platforms.

To the best of our knowledge, the *VANET testbeds* found in the literature prove the feasibility of cooperative communications (V2V, V2R), and test VANET open issues, such as routing protocols or mobility. Feasibility is usually proven through measurements in realistic conditions (on the road) and in several scenarios, such as urban or highway [4-5]. Such VANET testbeds consist of a convoy of two to several vehicles, each equipped with a car gateway with some functionality. Car gateways are typically laptops equipped with wireless network cards and GPS receivers.

For example, FleetNet's car gateway [3] is composed of two interconnected laptops, one acting as a Linux-based router (V2V and V2R communications using IEEE 802.11b) and the other serving as a Windows-based graphical interface to the user and to the GPS receiver. In [5], the car gateway is a laptop running Linux equipped with a PCMCIA 802.11b/g card with an external antenna, a Bluetooth-based GPS receiver and a multi-band WiFi driver for to monitor the packets that reach the network card.

Another example is the testbed of the US Department of Transportation, National Highway Traffic and Safety Administration (NHTSA), which comprises a GPS receiver, a Windows-based laptop, an IEEE802.11a radio device adapted to the dedicated SRC standard and simulation software to configure and test VANET performance [6].

The testbeds described above are similar to other relevant implementations, such as Carnegie Mellon University's testbed [7], the European projects COM2REACT [8] and MYCAREVENT [9], and ongoing implementations such as the CVeT testbed [10]. A more advanced demonstrator is that of the European CVIS project, currently under implementation, which will be based on the Communications, Air Interface, Long and Medium Range (CALM) architecture [11]. Finally, the Japanese Internet Intelligent Transportation Systems (ITS) Consortium (www.internetits.org) is also active in vehicular platform and field test activities.

As for *vehicular service platforms*, most testbeds use highly-demanding applications to analyze the factors that affect the quality of transmitting data from these applications over different types of vehicular networks and in different scenarios in terms of connection time, packet loss, throughput, etc. For example, in [2] the authors provide content sharing over a VANET, while [5] analyzes the performance of video streaming over a similar network. In these testbeds, services are typically only provided in terms of traffic, and sometimes emulated by using traffic generation tools such as Iperf. Other testbeds provide applications, e.g., FleetNet's cooperative driver assistance or decentralized floating car data services [3]; service platforms, e.g., after-sales and car maintenance in MYCAREVENT [9]; and intermediate service layers, e.g., the COM2REACT's virtual sub-center, which allows peer-to-peer implementation of traffic applications. It is worth noting that the CVIS project also considers applications for monitoring, urban, interurban, freight and fleet scenarios [11]. Moreover, there is also a number of disruption-tolerant network testbeds that provide Internet connectivity to moving vehicles via open access points in rural and urban environments. Examples of this are DieselNet, CarTel and Drive-Thru [12].

3. THE DRIVE TESTBED

The DRIVE testbed consists of a dedicated hardware platform embedded with a software platform containing the communications node, the vehicular services and the Human-Machine Interface (HMI) to be integrated on-board a vehicle. Moreover, DRIVE includes communication and management elements in the network infrastructure. DRIVE's on-board configuration can be used standalone or in a distributed way by setting up connections with other DRIVE-equipped vehicles. Apart from providing proof of concept, DRIVE collects network- and service-related parameters to analyze the performance of vehicular networking and advanced services.

3.1 Generic architecture

Fig. 1 illustrates the generic architecture of the on-board DRIVE demonstrator. The infrastructure elements are described in Section 4.2. DRIVE's core is the car gateway, which integrates the hardware and software components that enable intra-vehicular, V2V, V2R and V2I communications, advanced services and HMI. The in-vehicle wireless network considered is composed of wireless sensors located in the vehicle and nomadic devices of the driver and/or passengers.

The car gateway may also have access to some of the vehicle sensors and actuators, and has access to the steering wheel and audio system of the vehicle for HMI purposes. As for communications, DRIVE's architecture considers radio interfaces for in-vehicle communications (Int 1 to M in Fig. 1), positioning (Sat in Fig. 1), inter-vehicle and roadside communications (Int 1 to P in Fig. 1) and communications with cellular networks (Int 1 to K in Fig. 1). To handle all these communications, the car gateway includes a hybrid communications node, which is described in Section 4.1.

Time-to-market constraints limit the number of radio interfaces in DRIVE. For this reason, the testbed currently integrates mature technologies, such as Bluetooth, Zigbee, GPS, WiFi or UMTS (Table 1), and will integrate WiMAX for low-mobility V2I communications in the near future. We are also evaluating other established technologies for in-vehicle and V2R communications, such as RFID or Near-Field Communications (NFC), as well as emerging technologies for V2V (e.g., IEEE 802.11p, also known as WAVE) and positioning (e.g., Galileo), which will be integrated when available in the market.

The car gateway also includes a block to handle services, which is described in Sections 3.3 and 5, and an HMI block for multimodal interfacing with the driver and passengers. The HMI modes considered in DRIVE encompass a Graphical User Interface (GUI), voice technologies and steering wheel interactions, which are designed to act as single uniform assistant to the user and turn interaction into a more efficient and less driver's attention-consuming task. Finally, a service portal is included in Fig. 1. This portal hosts advanced added-value services and complex speech-based services (e.g., natural language interpretation), and will allow the provision of an open service access interface for third-party developments.

Table 1. Communications technologies of DRIVE (phase 1)

Positioning	In-vehicle	V2V and V2R	V2I
GPS	- Bluetooth, Zigbee, WiFi - CAN bus	WiFi: 802.11a/b/g	UMTS WiMAX*

* Technology under evaluation, to be integrated in 2008-2009.

3.2 Hardware platform

There are several implementation scenarios for the generic architecture described in Section 3.1: embedded, aftermarket device/s and nomadic. In its first phase, DRIVE is being implemented in an aftermarket scenario. In this context, laptops are usually employed in vehicular testbeds because they allow for easy implementation and modification of network characteristics and they facilitate the development of testing software. However, the use of laptops 'hides' important constraints of systems to be integrated in vehicles, such as power, processing capacity, storage, memory or size. For this reason, the car gateway is based on a powerful, extensible and flexible carPC architecture with:

- A central processing unit,
- HMI interfaces,
- communications interfaces and
- in-vehicle and roadside sensors.

The *central processing unit* is composed of a motherboard (VIA-EPIA CN-1300 mini-ITX with 1.0 GHz VIA C7 processor, 1GB DDR2 RAM and 80 GB hard disk), a rugged aluminum Voom-PC2 enclosure, and is powered by a M2-ATX Smart Automotive supply (140W). As for the HMI interfaces, they are currently a touch screen monitor for the GUI (7" Xenarc), a microphone and several speakers for the voice HMI, and Car Area Network (CAN)-based HMI (steering wheel). Fig. 2 shows the two implementations of DRIVE: integrated in a vehicle (Fig. 2a) and laboratory demonstration (Fig. 2b).

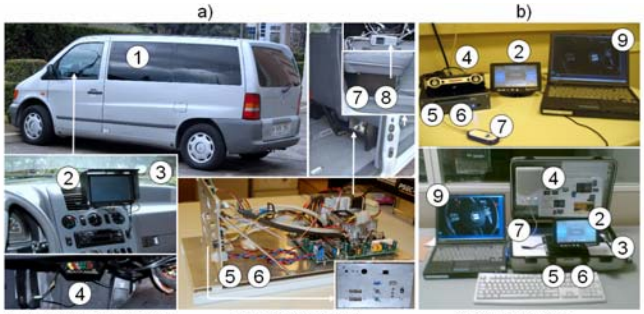

Figure 2. a) In-vehicle and b) lab implementations of DRIVE

DRIVE's *communications interfaces* are connected to the processing unit through USB or wireless-USB adapted ports. Table 2 illustrates the interfaces and adapters that will be available in the testbed. Currently, the in-vehicle integration of

DRIVE features WiFi for V2V and V2R, 3G for V2I and Bluetooth for in-vehicle communications, while the lab implementation features only V2I technologies. Apart from the car gateway equipment, DRIVE's hardware will include in-vehicle wireless sensors and roadside infrastructure for V2R, still under deployment. To this end, several Zigbee motes are being integrated to simulate different in-vehicle wireless sensors (Fig. 3). Moreover, the V2R communications scenario of the testbed, depicted in Fig. 3, considers rugged dual radio (WiFi 802.11a and 802.11b/g) access points, as well as further Zigbee motes and gateways to simulate roadside sensors.

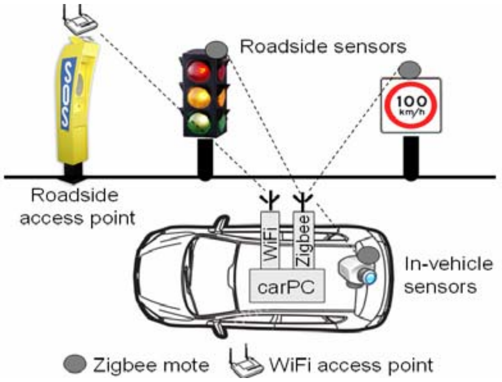

Figure 3. Example of in-vehicle and road sensors in DRIVE

Table 2. Communications interfaces for DRIVE

Positioning	In-vehicle	V2V and V2R	V2I
5 different GPS receivers: - Garmin 18 Deluxe USB, - TomTom Bluetooth, - Holux GR-213 RS-232 - San Jose GM-48 - GlobalSat Bluetooth BT-308	2 USB-Bluetooth adapters: - SMC BT10 - Conceptronic cbtu2 Zigbee: motes Tmote Sky and Tmote Connect gateway: 2 USB-WiFi adapters: - Netgear MA111 - Conceptronic c54ru	On board: USB-WiFi adapters Netgear MA111, Conceptronic c54ru Outdoor (infrastructure): rugged access points Colubris InReach CN-330	3G modem: Novatel MC950D HSPDA

3.3 Software platform

The standard approach for car manufacturers is to develop systems by assembling components, which are increasingly being designed and provided by external suppliers. In this context, the AUTOSAR partnership (www.autosar.org) is developing an open industry standard for automotive software architectures to be integrated in electronic controller units. AUTOSAR seeks modularity, scalability, function reusability and transferability [13]. In line with these technical goals, the software platform of the DRIVE testbed is designed as an open, modular, reusable architecture focused on the vehicular environment. This software architecture shall allow for the rapid development and integration of services with guaranteed resilience and reliability requirements.

As depicted in Fig. 4, DRIVE's software platform is based on middleware by the Open Services Gateway initiative (OSGi), which brings about a Java-based framework for the development, deployment and monitoring of services (bundles). Bundles are interfaced with one another, and they can be uploaded or downloaded dynamically with the tools provided by OSGi. DRIVE's software architecture can be divided into five levels, which are described at the following from top to bottom.

The first level is formed by the services, which use the HMI provided by the platform. After that, we find the OSGi bundles, which can deal with control, resources or service interfacing. Control bundles encompass administration, i.e., management of bundle (de)activation; remote administration and multimodal event management. Resource bundles deal with available devices and resources in DRIVE, such as Text to Speech (TTS), Automatic Speech Recognition (ASR), screen, GPS, database access or WebService clients, among others. Finally, service interfacing bundles include an application engine and bundles with basic services, i.e., part of other services, such as a browser or a telephone. The third level is the Java virtual machine (JVM). Below that, we find the operating system, with native drivers for resources and devices. DRIVE runs on Linux and uses C and C++ based drivers for TTS and ASR. Finally, we find the hardware platform (Section 3.2).

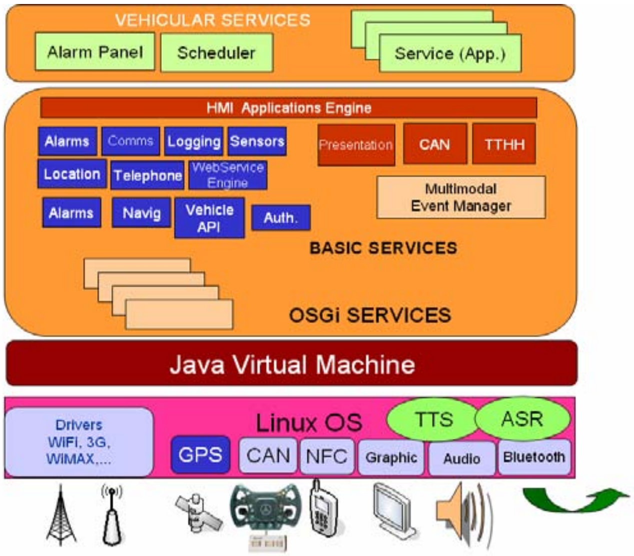

Figure 4. Architecture of the DRIVE software platform

4. VEHICULAR COMMUNICATIONS

The particular requirements of vehicular services make the design and development of an appropriate communication system for this environment a very complex task. A single wireless solution would be ideal to provide on-board nodes with communications, but this is not possible, at least now, because of the special characteristics inherent to the vehicular environment, which include unpredictable network topology changes, high speeds of the nodes, variable node density depending on traffic situation, multiple simultaneous accesses to the network, etc. For this reason, wireless nomad, cellular, ad-hoc and broadcasting technologies shall be included in a car gateway to fulfill the requirements of both the services and the environment. This leads to a *hybrid node* architecture

4.1 The on-board hybrid node

The architecture of DRIVE's hybrid node is depicted in Fig. 5, which details the modules of DRIVE's Communications block in Fig. 1. It is aligned with the CALM architecture [11] and its associated technologies, which are being standardized by the ISO TC 204 Working Group (WG) 16 [14]. The hybrid node also follows the recommendations of the European eSafety Forum, particularly of its Communications WG [15].

The hybrid node is the module in charge of communicating with any device in the vehicle and with any other node in the surrounding networks: other vehicles (V2V), road infrastructure elements (V2R) and the Internet (V2I). To this end, the hybrid node uses the best choice among the available technologies (Table 2) for every situation. In other words, the rationale behind DRIVE's hybrid node is the well-known concept of Always Best Connected (ABC), i.e., always choose the best communication choice possible or available for the given service requirements and user preferences. Examples of ABC study cases are rapid information exchange among vehicles, and high-quality (QoS) services. Rapid exchange of information, e.g., warning of incidents or dangerous situations on the road, requires a fast mechanism to send alert messages, which is best provided by a vehicular ad-hoc network (in DRIVE, WiFi-based VANET). On the contrary, high-quality demanding or security-critical services need network infrastructures such as cellular networks.

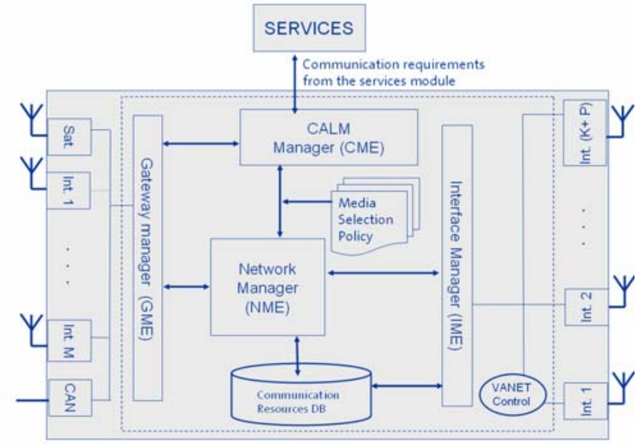

Figure 5. Architecture of the hybrid node in the car gateway

Using the interfaces provided by the software and hardware platforms described in Section 3, the hybrid node will be able to monitor the quality of all the available links to a given destination at any moment. Quality parameters may range from loss of coverage or availability to transmission parameters such as throughput, packet loss rate or signal-to-noise ratio. Based on this monitoring, and taking into account the service requirements, DRIVE decides dynamically which link is the best to use, and then routes the information through the chosen link. This is done by the different managers in Fig. 5, namely CALM, network and interface managers. Moreover, the hybrid node has a separate module to provide the functionalities for VANET deployment, i.e., ad-hoc routing mechanisms, gateway discovery, configuration and service discovery.

Note that ABC requires handling of horizontal and vertical handovers between different wireless technologies, which must be transparent for the user in the sense that the hybrid node will perform the choice automatically. For example, if a priority link goes down, DRIVE will change the status and the information will be sent using the best suitable network with neither an actuation by the user nor service disruption. To this end, proposals such as the Media Independent Handover (MIH) are being studied for future inclusion in DRIVE.

4.2 Communications architecture

Apart from the hybrid node, which will be located on board vehicles, DRIVE's communications system provides all the functionalities needed for the appropriate performance of a heterogeneous network from the infrastructure point of view. These functionalities, depicted in Fig. 6, are the following:

- VANET gateways,
- AAA servers,
- IMS enablers and
- management platforms.

VANET gateways are network access elements equipped with two wireless interfaces and with the processing capabilities to communicate with the vehicular nodes using VANET protocols and with the Internet using a cellular link as a backhaul, e.g., UMTS. These gateways are autonomous, which makes the installation of DRIVE easy and portable. The servers for Authentication, Authorization, and Accounting (*AAA server* in Fig. 6) are needed for controlling the access to the vehicular services, as well as for authorization and accounting. Concerning the *IMS enablers*, the new IP Multimedia Subsystem (IMS) will be adopted in most telecom operator networks to provide control functionalities for service deployment, such as presence, location, terminal profiling or group management, among others. IMS enablers are being standardized by the 3rd Generation Partnership Project (3GPP) [16] and the Open Mobile Alliance (OMA) [17]. Finally, the communications network must have tools to monitor and manage all the technologies involved. To this end, a *management* platform is needed to make the heterogeneous system work as a single network with integrated provisioning mechanisms, QoS and security aspects.

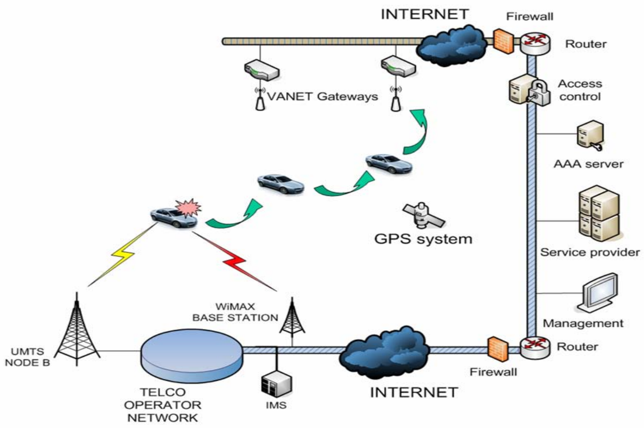

Figure 6. Architecture of DRIVE's communications system

4.3 Intra-vehicular, V2V, V2R and V2I communications

With the elements described in Sections 4.1 (on board vehicles) and 4.2 (network infrastructure), DRIVE is capable of performing intra-vehicular, V2V, V2R and V2I communications depending on the QoS needs of services. Concerning *intra-vehicular communications*, DRIVE supports gateway functionality (gateway manager in Fig. 5) to provide on-board sensors and nomadic devices used by passengers with communications. The latter is especially useful in public transportation, where numerous users want to get connectivity, and also in scenarios where external services need to collect data from sensors on board vehicles, e.g., to measure weather conditions. Currently, DRIVE features Bluetooth, WiFi and Zigbee for internal communications in the vehicle. Bluetooth is a mature, SRC technology appropriate for communication with on-board wireless sensors and for accessing the gateway by nomadic devices on private vehicles. Zigbee is a simple SRC commonly used to communicate with wireless sensors. WiFi is used here for longer-range, in-vehicle communications, e.g., nomadic devices on buses or trains.

DRIVE also supports communications with other vehicles (V2V) and with roadside equipment (V2R). Both *V2V and V2R communications* are achieved through VANETs. Currently, the testbed integrates IEEE 802.11a/b/g (WiFi) for VANET communication. Note that in-vehicle WiFi communications only use IEEE 802.11a. It is worth mentioning that these wireless technologies were not designed to support ad-hoc vehicular communication requirements, but only “fixed” nomadic and mobile network schemes. For this reason, the performance of a VANET using WiFi is not optimal. To fill this gap, the Institute of Electrical and Electronics Engineers (IEEE) is finalizing a standard that will cover specifically the VANET challenges: the IEEE 802.11p or Wireless Access for Vehicular Environment (WAVE) [15] [18]. However, until WAVE is commercially available, DRIVE will use WiFi for V2V and V2R communications. For this reason, DRIVE currently uses a VANET over WiFi for safety-related services, where every vehicle shall receive warning messages with very low delays. Section 6 summarizes an experimental study of WiFi-based VANETs in urban and interurban environments, which was performed with part of DRIVE’s hybrid node in the context of the COM2REACT project [8].

Last but not least, DRIVE features *V2I communications* through third-generation (3G) cellular technologies and IEEE 802.16 d/e (WiMAX, in the near future). UMTS is used by DRIVE when a service must be provided with high quality or security requirements. The rationale behind the adoption of WiMAX is to provide broadband access and higher rates, which is useful for multimedia services (infotainment, as described in Section 5). The applicability of WiMAX as a V2V/V2R technology will also be studied in DRIVE.

5. VEHICULAR SERVICES IN DRIVE

We envision future vehicular services as a co-operative network of distributed agents (vehicular, infrastructure, servers, as depicted in Fig. 6) that will be able to collaborate intelligently in the generation, distribution and presentation of the information associated to any kind of service. Under this scheme, each node will play the role of producer and/or consumer of information (e.g., vehicles as floating sensors) and will implement some service logic to process, filter, store and forward any kind of information that will finally shape the end-user services.

On top of this service architecture we foresee the deployment of a mix of safety-related free services, probably fostered by regulations (e.g., the future European eCall), and a full set of added-value services that the customer will subscribe to and pay for. We believe that this approach will help create and deploy novel services adapted to the special characteristics of the vehicular environment and, at the same time, establish smart business models that increase service acceptance by the users (vehicular service customers).

5.1 Service classification

Among there many classifications of ITS services that can be found in the literature, DRIVE’s services are grouped into:

- Car and driver,
- passenger,
- infrastructure and
- fleet management services[1].

Car and driver services are mainly aimed to help the driver in his/her main tasks (driving safely and comfortably, taking better decisions, being informed at all times), and they also include new functionalities into the car or improve the existing ones, e.g., remote diagnosis, physical placing of elements. Note that safety-related services are fostering the design and deployment of vehicular services, especially through initiatives by Public Authorities, e.g., the i2010 initiative by the European Union, which includes the emergency call (eCall) service, being currently standardized. DRIVE considers five relevant sub-categories in this group:

- *Safety:* eCall, virtual traffic signaling, incident warning, collision avoidance.
- *Adaptive driver assistance systems* (ADAS)
- *Security:* anti-theft, parental control, car location.
- *Driver information:* up-to-date maps, address look-up, smart navigation, geo-referenced information.
- *Comfort:* personalization-oriented services based on user identity management, allowing customization of logical and physical environments within the vehicle.

Passenger services are aimed to any passenger in the vehicle (including the driver) and, provided the right conditions (interaction facilities and capabilities) are given, are able to be personalized and customized according to an individual profile. These services, built upon the concept of Mobile Personalized Environment [19], are closely related to other environments, such as the home, office or entertainment, and are gaining relevance as users spend more time in their vehicles and demand “always-on” services. This kind of services also require novel interaction facilities such as enhanced terminals (e.g., with keyboard), user-friendly interfaces or voice HMI (TTR and ASR). DRIVE considers three categories here:

- *Infotainment*, which includes information and entertainment (mainly multimedia) services. Examples of information services are customized information channels (news, sports,

---

[1] Fleet management is out of the scope of DRIVE. Hence, this service group will not be described any further in this paper.

business, etc.) and location-based information, e.g., traffic or weather. Multimedia services include any audio/video service from in-vehicle (CD/DVD/HD/MP3) devices or from network sources, e.g., through download or streaming.

- *Productivity*, which encompasses office-like services aimed to professional customer groups such as email (e.g., Blackberry-like), calendar management, traditional web-browsing or Microsoft Office-like.
- *Communications*, which includes well-known services such as Bluetooth-like voice-call control, contacts and agenda management, SMS or chat, among others.

Last but not least, *infrastructure services* are mainly oriented towards Traffic Authorities and infrastructure managers. The final goal is to help vehicles on the road make a better and more efficient use of the infrastructure facilities. These services can be divided into public and private. Public services shall be offered by Public Authorities, and are usually related to public infrastructures. Examples of these are traffic management, special emergency management or dangerous goods transportation [20]. Private services are offered by private infrastructure managers, bussiness or enterprises, e.g., parking reservation and payment and remote toll payment.

Table 3. Current services in DRIVE

Car and driver	Passenger
- Basic eCall - Driver self-help browser - Remote vehicle diagnostics, advice and help	- General & sports news channels - GPS-based weather and traffic - Audio & video play. Content purchase/synchronization - Office: email access, web browsing - Communications: phone, SMS

5.2 Services implemented in phase 1

Table 3 lists the services already deployed or being deployed in DRIVE at the time of writing, according to the classification described in Section 5.1. The infrastructure elements that host and provide these services are depicted in Fig. 1 (service portal) and Fig. 6 (service provider), while the in-vehicle service platform is described in Section 3. Fig. 7 illustrates the service environment of DRIVE, which includes the technologies of Table 1 and the service portfolio of Table 3. In the car and driver group, self-help services are worth noting. These range from on-line manuals to help drivers in troubleshooting to a call-center-based help service for remote diagnostics, which was developed in the framework of the MyCarEvent project [9] and is now part of DRIVE. Apart from the services listed in Table 3, we are developing the following portfolio for car and passenger services:

- *Intelligent driving assistance system:* advanced navigation system that will warn the driver about any potential dangerous situation on the road and will provide relevant context-based information at the right moment and in a correct manner.
- *Enhanced eCall:* starting from the available basic eCall (Table 3), this service will include relevant information about any accident (e.g., real-time information about the occupants in the vehicle, black-box information for accident post-analysis, etc.).
- *Smart driving monitoring system:* automatic driver profiler that will allow insurance and/or car rental companies to charge drivers according to their driving manners and offer some bonuses for responsible driving.

Through the graphical interface developed, Fig. 8 illustrates some of the services available in DRIVE. Note that usability aspects have been taken into account to design the GUI, with of simplicity and intuition always in mind to achieve ease of use. The services depicted include eCall, telephony (voice and SMS), information channels, multimedia and Internet (web browsing and e-mail access).

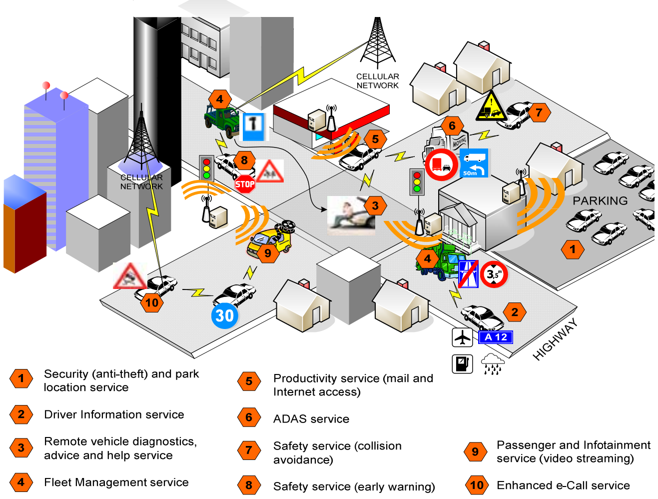

Figure 7. DRIVE's environment for future vehicular services

Figure 8. eCall, infotainment and communication services

6. HUMAN-MACHINE INTERFACES

The HMI is probably one of the most crucial technologies in the design of vehicular services due to its huge impact on security issues, which are being watched so closely by Public Traffic Authorities. In fact, there is a consensus that only services fully compliant with shortcoming HMI regulations will be allowed in vehicles. Hence, multimodality and HMI usability are key aspects to take into account.

More services with enhanced functionality are pushing to be included in the available vehicle interfaces. Moreover, car manufacturers and service providers want to differentiate from competitors by including (sometimes) fancy applications. This is resulting in an increased degree of complexity in the on-board HMI, which demands more and more attention from the driver and thus might distract him/her from what should be his/her main and almost only role: driving safely. DRIVE's approach concerning HMI is based in a two-sided strategy:

- *Overall usability review*: within DRIVE, a usability study has been done for each on-board HMI interface. The stress is put on the interfaces demanding an increased degree of interaction with the driver, such as GUI. Through this general study, the impact on the overall system is checked. Additionally, the study has to be customized on a per-service basis to cope with particular characteristics, user profiles and regulations.
- *Fully multimodal interface*: DRIVE considers the use of all the interaction channels present in the vehicle to match the best of the driver's interaction capabilities for each situation. As described in Section 3, three main HMI channels have been identified: voice, GUI and in-car-control interfaces.

Voice-based interfaces allow the driver to access all the services and facilities present in the vehicle with a minimum guaranteed degree of distraction. Still, they have problems that need to be taken care of, e.g., the difficulty to keep a context in cases where many nested "menu" levels are needed, or in very open interfaces (e.g., maps), in which a GUI is a much powerful interface. As for GUI, traditional graphical interfaces are in a clear withdrawal, but can still be used on board with restrictions, i.e., if used by the driver, they will be disabled when a certain speed is reached. Touch screens are the most commonly used GUI devices as they integrate both the input and output interaction devices, eliminating the need for external input devices. It remains an open question whether new sorts of GUI interfaces, e.g., head-up displays, will be able to overcome the constraints of GUI usage. Finally, in-car control interfaces include all sorts of integrated controls in the car, which are usually accessible through keys and pads in the steering wheel. Although these are already being widely used in vehicles for fixed applications, they also have been proven to work better and to be much more flexible when used in conjunction with oral interfaces for vehicular services, giving the user an oral feedback and oral context information.

7. SUMMARY

In a single architecture and platform, DRIVE integrates the features that other testbeds only provide in a partial way, namely: the use of a carPC to implement the car gateway, which influences the performance capabilities of the system; the combination of in-vehicle, V2V, V2R and V2I communications according to the ABC concept; the real implementation and provisioning of services for the end-user; and a flexible implementation that can be easily adapted and updated to technology changes without changing the hardware installed in vehicles. The latter is especially important for the car industry, in which cars have a 10- to 15-year lifespan while the communications devices used by common people last no longer than 2-3 years.

On the other hand, thanks to its distributed, ubiquitous nature, the DRIVE environment needs a minimum set of users/nodes to reach its full potential. For instance, ad-hoc mesh networks take advantage of the number of nodes to build a more resilient and flexible network topology. Regarding services, user-generated content (e.g., reporting incidents) is built upon the collaboration of multiple users to share certain information. Hence, DRIVE is implemented taking into account restrictions such as cost, inter-operability or usability to ensure that a critical mass of users is achieved that generates synergies and allows building pervasive novel services for all kind of drivers. Finally, DRIVE also integrates multimodality aspects of the HMI. In the near future, DRIVE will be used in field tests to conduct experimentation on vehicular communications, applications and user experience.

8. REFERENCES

[1] H. Wu, *et al.*, "An empirical study of short range communications for vehicles", in Proc. ACM VANET, September 2005.

[2] K.C. Lee, *et al.*, "First experience with CarTorrent in a real vehicular ad hoc network testbed", in Proc. VANET MOVE, May 2007.

[3] A. Festag, H. Füssler, H. Hartenstein, A. Sarma, R. Schmitz, "FleetNet: Bringing carto-car communication into real world," in Proc. 11th ITS World Congress, October 2004.

[4] V. Bychkovsky *et al.*, "A measurement study of vehicular internet access using in situ Wi-Fi networks", in Proc. MOBICOM, Sept. 2006.

[5] M. Jerbi, S.-M. Senouci, M. Al Haj, "Extensive experimental characterization of communications in vehicular ad hoc networks within different environments", in Proc. IEEE VTC 2007.

[6] US Department of Transportation, National Highway Traffic and Safety Administration, "Vehicle safety communications project: Appendix C,", September 2006.

[7] R. Mangharam, J.J. Meyers, R. Rajkumar, D.D. Stancil, J.S. Parikh, H. Krishnan, C. Kellum, "A multi-hop mobile networking test-bed for telematics", in Proc. SAE World Congress, 2005.

[8] Website IST-FP6 COM2REACT Project, www.com2react-project.org.

[9] Website MyCarEvent Project, www.mycarevent.com.

[10] Website CVeT Testbed, www.vehicularlab.org/testbed.php.

[11] P. Kompfner, "Cooperative vehicle-infrastructure systems. European vision for V2I communications", 4th ACM Workshop on Vehicular Ad Hoc Networks, Sept. 2007.

[12] J. Ott, D. Kutscher, "Drive-Thru Internet: IEEE 802.11b for 'Automobile' Users", in Proc. IEEE INFOCOM, 2004.

[13] A. Sangiovanni *et al.*, "Embedded system design for automotive applications", in IEEE Computer, Vol. 40, no. 10, pp. 42-51, Oct. 2007.

[14] Website of the ISO TC204 WG 16, www.calm.hu.

[15] eSafety Forum, "Final Report and Recommendations of the Communications Working Group". September 2007.

[16] The 3rd Generation Partnership Project (3GPP), "3GPP Specification Series 22, 23, 24 and 29", www.3gpp.org.

[17] The Open Mobile Alliance (OMA), "IP Multimedia Subsystem", available on www.openmobilealliance.org.

[18] The the Institute of Electrical and Electronics Engineers (IEEE), "IEEE 1609 - Family of Standards for Wireless Access in Vehicular Environments (WAVE)", in progress.

[19] Website Ambient Networks Project, www.ambient-networks.org.

[20] Website IST-FP6 GOODROUTE Project, www.goodroute-eu.org.