Signal Processing, Control and Coordination in an Intelligent Connected Vehicle

Manolo Dulva Hina[1](✉), Sebastien Dourlens[2], Assia Soukane[1], and Amar Ramdane-Cherif[2]

[1] ECE Paris School of Engineering, 37 quai de Grenelle, 75015 Paris, France
{manolo-dulva.hina, assia.soukane}@ece.fr
[2] LISV Laboratory, Université de Versailles St-Quentin-en-Yvelines, 10-12 avenue de l'Europe, 78140 Vélizy, France
sdourlens@hotmail.com, rca@lisv.uvsq.fr

Abstract. In this paper, we present the functionalities of an intelligent connected vehicle. It is equipped with various sensors and connected objects that enable communication between the driver and its environment. This system provides assistance towards safe and green driving. The driving assistance may be directed towards the driver (semi-autonomous vehicle) or completely towards the vehicle (self-driving, autonomous vehicle). The assistance is based on the driving context which is the fusion of parameters representing the context of the driver, the vehicle and the environment. This cyber-physical vehicle has three main components: the embedded system, the networking and real-time system and the intelligent system. The architecture for data transfer within the connected vehicle is implemented through publish-subscribed infrastructure in which services are transferred and controlled in an orderly manner. These functionalities are tested both in the laboratory and on the road with satisfactory results. This is the fruit of labor of a consortium composed of five industrial and two academic partners.

Keywords: Connected vehicle · ADAS · Internet of things · Intelligent vehicle

1 Introduction

Cyber-physical systems (CPS) are physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core. CPS is composed of interconnected clusters of processing elements and large-scale wired and wireless networks that connect a variety of smart sensors and actuators [1]. This paper is about a cyber-physical vehicle [2], an intelligent connected vehicle and its smartphone app created to contribute to the reduction of road traffic accident. This connected vehicle proposes safe driving and green driving [3] assistance based on the given driving context. The driving context [4] is the driving situation based on the fusion of various parameters (obtained from sensors, internet of things, etc.) representing the context of the driver, the vehicle, the environment and the infrastructure. An intelligent connected vehicle can be partitioned into three subsystems, each representing a distinct signal processing part. They are given below:

R. Zitouni and M. Agueh (Eds.): AFRICATEK 2018, LNICST 260, pp. 32–43, 2019.
https://doi.org/10.1007/978-3-030-05198-3_3

- *Embedded system*: it is that sub-system responsible for capturing data and signals from the vehicle, and environment. Various sensors, gadgets and actuators of different modalities are associated with embedded system. Altogether, these components form connected objects and Internet of things (IoT) [5].
- *Intelligent System*: it is that sub-system that is responsible for obtaining input data from the embedded system, fusion them in order to deduce the driving situation and determine what action must be undertaken to the specified driving situation. This assistance may be directed towards the vehicle (in which case, the vehicle becomes an *autonomous, self-directing machine*) or the driver (in which case the driver is in complete control of the vehicle, a case of *semi-autonomous vehicle*).
- *Network and Real-time System*: it is that sub-system concerned with communication protocols between components of the system. The communication between embedded system and intelligent system is handled by this component. The repository of vehicle data may be the vehicle itself or the Cloud computing infrastructure.

This paper is a continuation of our previous work [6, 7] only that here the focus is on the detailed treatment of the fusion of signals, the processing for the detection of driving context and the fission process to offer driving assistance.

2 Related Work

As early as 1920’s, various efforts have been made to automate vehicle driving [8]. Some promising trials happened in 1950’s and research on this has never ceased since then. The first autonomous vehicle was developed in 1980’s in Carnegie Mellon University Navlab [9] and AVL in 1984 [10]. It is followed by Mercedes-Benz and Bundeswehr University Munich’s Eureka Prometheus Project in 1987. In July 2013, Vislab demonstrated BRAiVE, a vehicle that moved autonomously on a mixed traffic route open to public traffic [11]. Connected and autonomous vehicles (CAV) are a technological revolution, combining radical changes in the design of the road vehicles and understanding of their interactions with the networked infrastructure.

Connected vehicles [12, 13] and autonomous vehicles [14, 15] face different obstacles for further development because their respective technologies are at different stages of development and implementation. Consumers express concern about cybersecurity and privacy issues while safety and consumer readiness to adopt are the perceived barriers for autonomous vehicles [16]. The core science and technology required to support CPS and cyber-physical vehicles (both for connected and autonomous vehicles) are essential for future economic competitiveness. Creating the scientific and technological basis for CPS can pay dividends across wide domains.

According to the survey conducted by Foley & Lardner LLP [16], the technologies for which the connected and/or autonomous vehicles are worth investing are as follows: cybersecurity protection, precision mapping and location technology, vehicle-to-vehicle (V2V) communication technology, advanced driver assistance system (ADAS), machine learning and driving data analysis, infotainment features, and self-driving vehicles for car-sharing. Indeed, this work on ADAS is on the right track. The machine learning [17–19] component of our system is integrated in the intelligent system. The V2V features are a priority by our networking and real-time system component. This paper contributes to the advancement of intelligent connected vehicles.

3 Driving Assistance for a Connected Vehicle

3.1 Objective and Offering a Technological Solution

There is a very wide range of mobility-related applications available in the market these days but most are not directly related to vehicle data. This is in contrast to the norm given that the use of vehicle data and its environment is at the heart of the Advanced Driver Assistance Systems (ADAS) [20, 21]. In order to test and validate the on-board driver assistance systems, it is necessary to equip the test vehicle with the correct sensors that will capture the perceived context of the driver, vehicle and the environment. Furthermore, these sensors should be of quality. The costs incurred in such configuration means that in most cases, ADAS functionalities are generally made available mostly on high-end vehicles and it will take (several) years before they can be made available to family car models. Given such situation, this project aims to address such constraints. Indeed, this project aims to: (i) Create an open environment for the design of innovative applications for smartphone, promoting the emergence of informative driving assistance functions; (ii) Offer an affordable solution by offsetting the processing capacity on the smartphone, optimized by exploiting data of the vehicle and its environment; (iii) Specify a safe and effective interaction between the driver, the mobile software and the synthesis of lives on board the vehicle; and (iv) Reinforce road safety by familiarizing users with ADAS-type functions and optimizing future on-board functions through feedback.

The driving assistance system for connected vehicle is a technological solution centred on the development of a smartphone driving assistance applications based on: (i) The sensors and resources of the smartphone to ensure the perception of the driving environment and the state of health of the driver as well as the treatment of the driving assistance functions; (ii) A communication channel between the smartphone and the vehicle to use vehicle data and activate vehicle functions (Car Easy Apps solution) [22]; and (iii) Establishment of a test platform to simulate the vehicle and its environment in order to develop, validate and demonstrate the applications developed.

To achieve the link with the vehicle data, this project has chosen to use the SDK (software development kit) of the project CEA (Car Easy Apps) [23]; CEA is a project from the Franco-Spanish consortium collaboration led by PSA Peugeot-Citroen[1], in which Continental[2] also participated. This consortium offers a framework under Android that allows wireless (Bluetooth or WiFi) or wired (USB) connection to interface with the multimedia system of the vehicle and thus access a wide variety of signals present on the various CAN (controller area network) bus of the vehicle. The CEA being scalable, one can easily add new signals as needed. See Fig. 1(a).

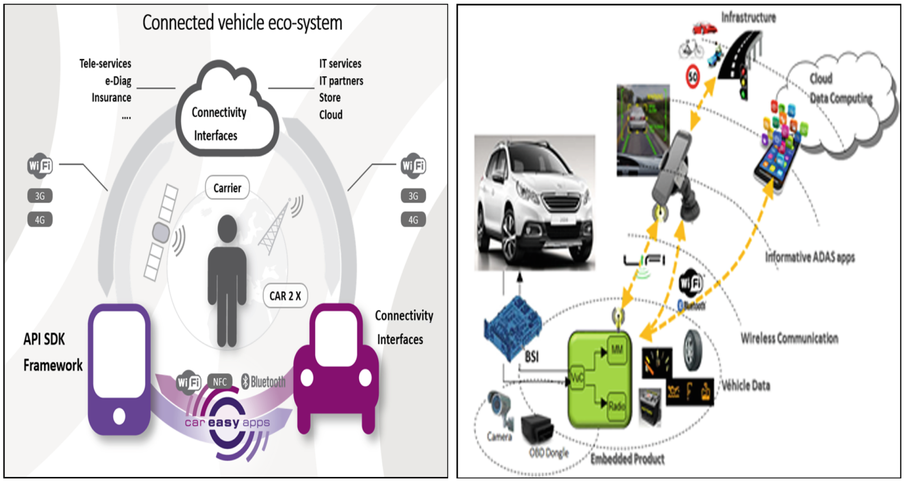
---

[1] PSA Peugeot-Citroën: https://www.groupe-psa.com/en/.

[2] Continental France: https://www.continental-corporation.com/fr-fr.

3.2 Connected Vehicle Architecture

The architecture of our intelligent vehicle project is shown in Fig. 1(b). The architecture is layered and layers are as follows: (i) the material layer (i.e. this contains pertinent sensors), (ii) the input services layer or pilots that apply the appropriate algorithms related to vision, identification of objects (e.g. road, road sign, weather, vehicles) and the transformation of the data into a low-level event, (iii) the server layer for storing and broadcasting events, (iv) the multimodal fusion layer of events that generate higher-level events, (v) the human-vehicle interface (an exit service) which provides the necessary information to the user in the form of web pages and dashboard utilities of the vehicle, and (vi) Multiple layers of communication to the cloud [24, 25] and road infrastructure.



(a) Car Easy App – The car as a connected object

(b) CASA Architecture

Fig. 1. (a) CEA – The car as a connected object, (b) the CASA architecture

Likewise, as shown in the diagram, the system components are as follows: (i) A vehicle – A PSA vehicle (Peugeot or Citroen) cockpit used as vehicle in the laboratory and the actual road tests; (ii) Various tools inside the vehicle – Given that we are interested in obtaining the driving context, we make use of various tools to realize such objective: (a) BSI – is a back-illuminated, digital-image sensor that uses a novel arrangement of imaging elements to increase the amount of light captured and thereby improve low-light performance; (b) OBD – is an onboard diagnostic gadget; it is a vehicle's self-diagnostic and reporting capability, providing end users with access to the status of various vehicle subsystems; (c) OVIP – is an infotainment platform developed for PSA vehicles; (d) Camera – for driver monitoring, and (e) Bluetooth and LiFi – for wireless communication; (iii) Smartphone – used as HMI (human-machine interaction) tool and to access from IoT and connected objects; and (iv) The CASA App – an Android application developed for this project, installed on the smartphone for driver assistance for safe and green driving.

4 Vehicular Data Communication, Processing and Control

4.1 Publish-Subscribe Architecture

Figure 2(a) shows the data communication structure in an intelligent connected vehicle. Such communication architecture is based on the "publish-subscribe" [26] mechanism: the publication and subscription of messages. In this structure, a publisher publishes messages without knowing a priori who the subscribers to these messages are. The subscriber is the recipient of the message or data. It too does not know who the producer/publisher of the message is. The structure is simple: a message class is associated with the sent messages without knowing if there are recipients or not. In the same way, the recipients only subscribe to the classes that interest them, and receive only the corresponding messages without knowing if there are senders. For interoperability with existing platforms and web services, the content of XML messages is in Javascript object notation (JSON) format [27]. The message transfer protocol is REST over TCP-IP [28]. SOAP [29] can also be used. For the protection of messages, the publication or subscription services authenticate using specific encryption keys.

4.2 Services in an Intelligent Vehicle

All vehicular services (input, intermediate or output) are independent in terms of architecture. For example, if the sensor information does not arrive, the system will not warn of hazard. In this system, the orchestration server (also called the "broker") receives messages, store them and distribute them at once to each subscriber upon receipt. In this way, the transmission delays are minimized. All services can publish as well as subscribe; there is no particular constraint. The server can also control the quality of service by measuring the response times of various services. There are many services that can be provided in an intelligent vehicle. This paper, however, is not going to deal with an exhaustive list of services; it is sufficient that some representative samples of services in an intelligent vehicle are presented. All others would follow the same pattern and principle. As shown in Fig. 2(b), our intelligent vehicle has the following services:

- *Vision Sensor Publisher*: this service collects data related to obstacle (e.g. pedestrian, vehicle, bicycle, etc.) detection and publishes such info whenever it finds one.
- *GPS Location Publisher*: this service obtains the position of the vehicle and publishes it whenever the position coordinates of the vehicle change.
- *Car Sensor Publisher*: this pertains to a group of sensor that monitors the driver status (i.e. fatigued, stressed, etc.) and that of the vehicle (i.e. its distance from vehicle/obstacle in front, presence of stop sign, etc.) and publishes such information.

- *Weather Service Publisher*: this service monitors if there is a fog, rain or snow; if the visibility is poor, and if the road is hazardous due to the weather condition.
- *Event Multimodal Fusion Publisher and Subscriber (EMFPS)*: this is the brain of the system. It subscribes to many signals to be able to fusion such information and deduce the driving situation. It determines if there is a driving assistance to be done for the given driving situation. If so, then it publishes the corresponding signals.
- *Tablet HMI Publisher and Subscriber*: This service is both a publisher and a subscriber. As a publisher, it sends signal the setting configuration for driving assistance preferred by the driver (example: the language of the driving assistance = English | French, etc.). As a subscriber, it receives driving assistance messages published by the EMFPS and it broadcasts a message directed towards the driver, the vehicle or both to assist in the driving situation.
- *Car Dashboard Subscriber*: this service obtains message from the broker to determine if it is going to put the air-conditioner on/off, the music on/off, the heater on/off, activate/deactivate the dashboard wiper, etc.
- *Car Broker*: this is like the maestro of an orchestra. It receives data or messages from the publishers and sends same to its subscribers.

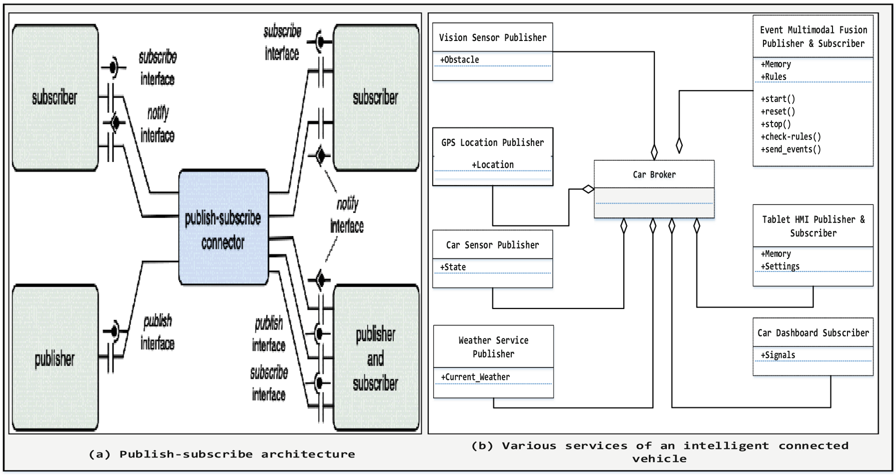


Fig. 2. (a) The publish-subscribe architecture, (b) the various services of an intelligent connected vehicle.

5 The Multimodal Event Fusion Service

The task of Event Multimodal Fusion Publisher and Subscriber (EMFPS) service is to perform the multimodal fusion of various events coming from other services, and to provide them messages and information in higher-level of abstraction in order to inform the user (the driver) on how to react on the new case of interaction. Briefly, it means informing the driver of the situation or of driving assistance message suited for the situation. The EMFPS service is both publisher and subscriber. The detection of context and the adaptation of exit events to the context are realized using a set of rules that are integrated in the service. These rules make use of all input information.

Following receipt of new data or message, the multimodal fusion involves applying a set of rules each time the values of subscribed variables gets updated. Figure 3(a) shows the EMFPS service. As shown, the solid arrows represent data flows between the functions (light blue rectangles) and the memory (cylinder), while the dotted-line arrows correspond to activations of broker server for publication and subscription purposes.

At start-up, the service initializes the array of variables "Memory" to their default values and then subscribes to the various events that it needs to receive. Subscribing means that the program gives the broker server the address of a service function. The call by the server of the call-back function (myCB) which manages the reception of events will be made by the broker server. Upon receiving data or messages related to one or more events, the service retrieves the values of the JSON and updates the variable data in memory and then performs an evaluation of all the rules that are affected by the updated variables. If one or more rules have been triggered, the output events are immediately published to the server.

5.1 Signals for the Cognition of Driving Context

In order to optimize and speed up the code, it is decided that there is a need to put all the variables in a table. The table of variables in the memory is implemented as shown in Fig. 3(b). The items in the table are as follows: (i) *`uuid`*: a unique identifier of each variable given by the server, (ii) *`name`*: name of the variable, (iii) *`unit`*: unit of value if numeric, (iv) *`value`*: value of the variable, (v) *`oldvalue`*: previous value needed to detect an update, (vi) *`received date`*: date of receipt of the variable in order to calculate its age, (vii) *`used in rules #`*: number of the rule affected by an update of this variable, and (viii) *`datatype`*: data type of the value of the variable (text or numeric).

As Fig. 3(b) shows, the set of variables are related to the detection of the following: (i) *context of the driver*: driver disturbance status; (ii) *context of the vehicle*: beam status, intersection direction, lane number, vehicle speed, vehicle engine status, vehicle turn indicator status; (iii) *context of the environment*: speed limit, intersection type, intersection signal, intersection distance, visibility, obstacle distance, obstacle position, obstacle speed, obstacle type, time to collision; and (iv) *green driving*: CO_2 driver advise, CO_2 explicit speed limit.

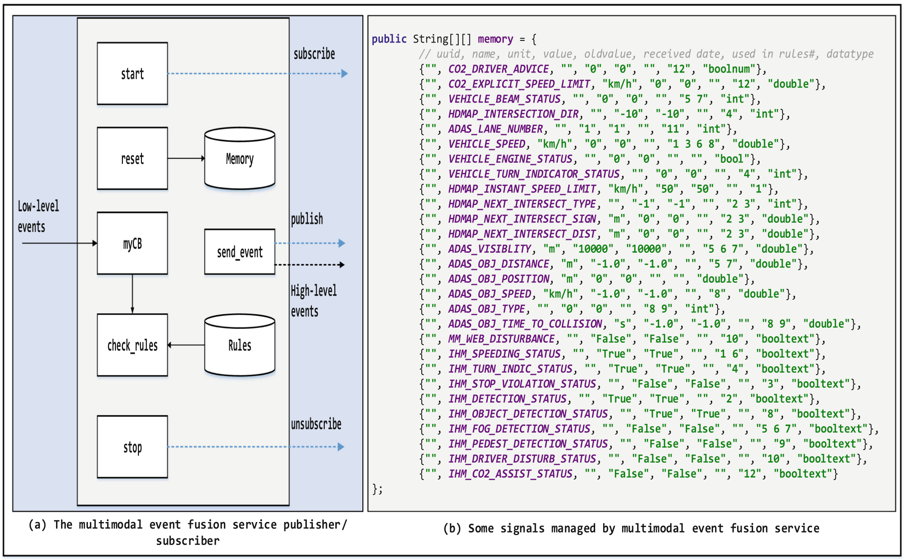
The service also provides driving assistance by sending messages to the server. Such messages are picked up by the Tablet HMI Subscriber and broadcasted to the driver, the vehicle or both. The driving assistance is of three types: (i) *assistance directed to the driver*: speeding status, turn indicator status, stop violation status, obstacle detection status, pedestrian detection status, driver disturbance status; (ii) *assistance directed to the vehicle*: activation of fog light (implemented by the Car Dashboard Subscriber service); and (iii) *assistance for green driving*: CO_2 assist status.

5.2 Human-Vehicle Interaction Interface

The human-vehicle interaction interface [30, 31] adopted for this work is shown in Fig. 4. The images shown in the diagram are mere depiction of the real ones which cannot be shown in this paper due to proprietary restrictions. In this work, there are two types of messages intended for the driver: (i) *Notification* – a message to inform the driver, and (ii) *Alert* – a type of message that attempts to get the driver's attention.

A notification or alert is sent according to the category of driving situation: (i) *Behaviour* – this refers to the driver's conduct of driving. For example, the driver's failure to stop in the Stop light merits an Alert message concerning the driver's behaviour; (ii) *Danger* – a potential risk to the driver or people on the road exists. Entering a foggy zone merits a notification message on danger; and (iii) *Ability* – this concerns about the person's ability to drive a vehicle. For example, if the system detects that a driver is going to fall asleep, a danger message concerning his ability to drive is sent.

When two or more messages need to be sent to the driver then system adopts a priority scheme. Only one message will be sent to the driver at any given time. Here is the priority scheme of this work: (i) *Alert* has a higher priority than *Notification*, and (ii) *Ability* has the highest priority in the message category. Next priority belongs to *Danger* and the last priority belongs to *Behaviour*.



```
public String[][] memory = {
        // uuid, name, unit, value, oldvalue, received date, used in rules#, datatype
        {"", CO2_DRIVER_ADVICE, "", "0", "0", "", "12", "boolnum"},
        {"", CO2_EXPLICIT_SPEED_LIMIT, "km/h", "0", "0", "", "12", "double"},
        {"", VEHICLE_BEAM_STATUS, "", "0", "0", "", "5 7", "int"},
        {"", HDMAP_INTERSECTION_DIR, "", "-10", "-10", "", "4", "int"},
        {"", ADAS_LANE_NUMBER, "", "1", "1", "", "11", "int"},
        {"", VEHICLE_SPEED, "km/h", "0", "0", "", "1 3 6 8", "double"},
        {"", VEHICLE_ENGINE_STATUS, "", "0", "0", "", "", "bool"},
        {"", VEHICLE_TURN_INDICATOR_STATUS, "", "0", "0", "", "4", "int"},
        {"", HDMAP_INSTANT_SPEED_LIMIT, "km/h", "50", "50", "", "1"},
        {"", HDMAP_NEXT_INTERSECT_TYPE, "", "-1", "-1", "", "2 3", "int"},
        {"", HDMAP_NEXT_INTERSECT_SIGN, "m", "0", "0", "", "2 3", "double"},
        {"", HDMAP_NEXT_INTERSECT_DIST, "m", "0", "0", "", "2 3", "double"},
        {"", ADAS_VISIBLITY, "m", "10000", "10000", "", "5 6 7", "double"},
        {"", ADAS_OBJ_DISTANCE, "m", "-1.0", "-1.0", "", "5 7", "double"},
        {"", ADAS_OBJ_POSITION, "m", "0", "0", "", "", "double"},
        {"", ADAS_OBJ_SPEED, "km/h", "-1.0", "-1.0", "", "8", "double"},
        {"", ADAS_OBJ_TYPE, "", "0", "0", "", "8 9", "int"},
        {"", ADAS_OBJ_TIME_TO_COLLISION, "s", "-1.0", "-1.0", "", "8 9", "double"},
        {"", MM_WEB_DISTURBANCE, "", "False", "False", "", "10", "booltext"},
        {"", IHM_SPEEDING_STATUS, "", "True", "True", "", "1 6", "booltext"},
        {"", IHM_TURN_INDIC_STATUS, "", "True", "True", "", "4", "booltext"},
        {"", IHM_STOP_VIOLATION_STATUS, "", "False", "False", "", "3", "booltext"},
        {"", IHM_DETECTION_STATUS, "", "True", "True", "", "2", "booltext"},
        {"", IHM_OBJECT_DETECTION_STATUS, "", "True", "True", "", "8", "booltext"},
        {"", IHM_FOG_DETECTION_STATUS, "", "False", "False", "", "5 6 7", "booltext"},
        {"", IHM_PEDEST_DETECTION_STATUS, "", "False", "False", "", "9", "booltext"},
        {"", IHM_DRIVER_DISTURB_STATUS, "", "False", "False", "", "10", "booltext"},
        {"", IHM_CO2_ASSIST_STATUS, "", "False", "False", "", "12", "booltext"}
};
```

(a) The multimodal event fusion service publisher/ subscriber

(b) Some signals managed by multimodal event fusion service

Fig. 3. (a) The multimodal event fusion service structure, (b) Some signals managed by the multimodal event fusion service provider. (Color figure online)

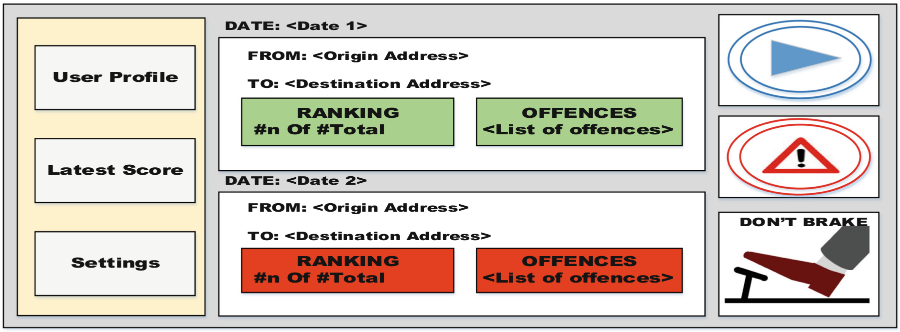


Fig. 4. Human-vehicle interaction interface of project CASA

5.3 Case Study: Scenarios Simulation

There is a need to simulate the multimodal fusion, fission and cognition of driving context that were presented in the earlier sections of this paper. To do this, a sample case scenario is developed and is shown in Fig. 5. Laboratory and road tests reveal that the CASA system is found to be operational and fit for purpose.

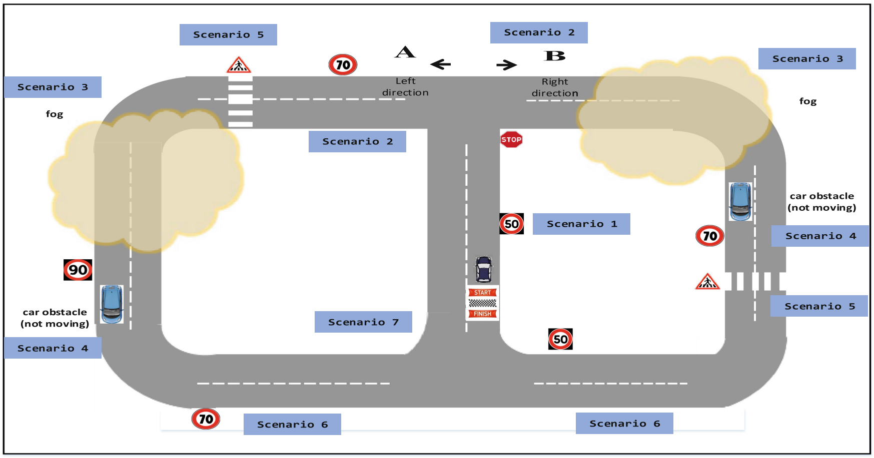


Fig. 5. The case scenario for the cognition of driving context and driving assistance

5.4 Sample Driving Actions

Consider for example the implementation of action "`ChangeLane`". There are two cases by which this action is invoked, as given below:

Case 1: Obstacle Moving Very Slow. (i) Let X be an instance of class "`Vehicle`" == `Vehicle (?X)`. (ii) Let there be road R == `Road(?R)`. (iii) Let there be lane

L1 in road R == `hasLane(?R, ?L1)`. (iv) Let there be lane L2 in road R == `hasLane(?R,?L2)`. (v) Let lane L1 be different from lane L2 == `differentFrom(?L1, ?L2)`. (vi) Let there be obstacle W (from previous subsection) == `Obstacle(?W)`. (vii) Let the speed of obstacle W be extra slow == . (viii) Let there be action Z for this driving situation == `Action(?Z)`. If we do fusion of these parameters, we will end up with the following: *Vehicle(?X) ∧ Road (?R) ∧ hasLane(?R,?L1) ∧ hasLane(?R, ?L2) ∧ differentFrom(?L1, ?L2) ∧ Obstacle(?W) ∧ hasSpeed(?W,ExtraSlowSpeed) ∧ Action(?Z) → ChangeLane (?Z).*

Case 2: Obstacle Has No Speed. **(i)** Let the description in subsection Case 1 holds true. (ii) Let the obstacle W have no speed (if might be a tree, a rock, etc.) == . If we fusion these parameters, we will end up with: *Vehicle(?X) ∧ Road (?R) ∧ hasLane(?R, ?L1) ∧ hasLane(?R, ?L2) ∧ differentFrom(?L1, ?L2) ∧Obstacle(?W) ∧ hasSpeed(?W, NoSpeed) ∧ Action(?Z) → ChangeLane (?Z).*

5.5 Contributions

In this paper, the functionalities of an intelligent connected vehicle is presented. The contributions of this work are as follows:

- *A generic, low-cost ADAS*: ADAS in general are reserved for expensive vehicles and closed to proprietary constraints. This work addresses those constraints.
- *A system adaptable for both autonomous and semi-autonomous vehicle*: The driving assistance can be set-up to be directed towards the driver, in which case the system is for a semi-autonomous vehicle. The assistance may be directed directly to the vehicle, in which case we will come up with an autonomous, self-driving vehicle.
- *An efficient publish-subscribe data communication architecture*: In this paper, the publish/subscribe architecture was presented. Publishers send their data or messages to the repository (the broker) without having idea who will consume it. Subscribers obtain the data and messages they need from the broker without having idea of who produced it. The subscribers are only informed that they need to pick-up their data/messages only when they are new. Hence, the data communication is efficient.
- *Assistance towards safe and green driving*: This work also contributes towards the reduction of traffic accident and on lesser fuel consumption while doing the same work. This means lower CO_2 emission, and also reduced cost of voyage.

The first phase of Project CASA was presented to the public during the 2015 ITS European Congress[3], 5–9 October 2015, Bordeaux, France. For five days, the project was presented to the public to get its opinion about the features CASA is offering. We collected data based on the public perception using questionnaires given to some 150 random participants. The results are as follows: (i) Consumers are willing to share their personal data with some applications. Consumers are most willing to grant access to applications that are directly related to navigation and mobility; and (ii) Consumers find that CASA is helpful for road navigation and mobility.

---

[3] http://itsworldcongress.com/.

6 Conclusion and Future Work

Project CASA is conceived with the vision that today's ADAS systems are closed to proprietary constraints and are usually accorded to expensive vehicles. CASA is a generic ADAS and can be used by any vehicle. This is an Android app and is based on smartphone. This project is a collaboration of five industrial and two academic partners, namely PSA-Peugeot-Citroen, Continental, Nexyad, Oktal, DPS and UVSQ. UVSQ, for its part, collaborates with ECE Paris on other intelligent system tasks.

The driving context is the fusion of all parameters related to the context of the driver, the vehicle and the environment. The driving context yields a situation that may or may not need assistance. In the case of it needs an assistance, the system determines the types of assistance using the rules that are stored in the system's knowledge base. The assistance may be directed towards the driver, the vehicle or both (although in this paper, the assistance is almost all directed towards the driver, a case of semi-autonomous vehicle). Taken into account that only one message can be sent to the driver at any given time, priorities are taken into consideration when two or more driving events occur at the same time. Tests and validations are done in the laboratory and on the road. Here, it showed that the use of CASA reduces traffic rules infractions and road accidents. Project CASA was also presented to the public during ITS European Congress in Bordeaux, France in 2015. It was well-received by the public. Future works include further improvement via machine learning component, in particular, the cognitive user interface design [32] and the cognitive component [33] that learns new driving situation, reason with purpose and interact with humans naturally. The component learns from its interaction with system users and from its experiences with the environment.

References

1. Rajkumar, R., Lee, I., Sha, L., Stankovic, J.: Cyber-physical systems: the next computing revolution. In: Design Automation Conference, Anaheim, CA, USA (2010)
2. Wang, S.: Develop Vehicle Control Systems as CPS for Next-Generation Automobiles, March 2015
3. Estl, H.: Paving the way to self-driving cars with advanced driver assistance systems, August 2015
4. Hina, M.D., Thierry, C., Soukane, A., Ramdane-Cherif, A.: Cognition of driving context for driving assistance. In: Presented at the ICAIA 2018: 20th International Conference on Artificial Intelligence and Applications, Kuala Lumpur, Malaysia (2018)
5. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of Things (IoT): a vision, architectural elements and future directions. Future Gener. Comput. Syst. **29**, 1645–1660 (2013)
6. Hina, M.D., Guan, H., Deng, N., Ramdane-Cherif, A.: CASA: safe and green driving assistance system for real-time driving events. In: Bi, Y., Kapoor, S., Bhatia, R. (eds.) IntelliSys 2016. LNNS, vol. 15, pp. 987–1002. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-56994-9_67
7. Hina, M.D., Guan, H., Ramdane-Cherif, A., Deng, N.: Secured data processing, notification and transmission in a human-vehicle interaction system. In: 19th IEEE International Conference on Intelligent Transportation Systems, ITSC 2016, Rio de Janeiro, Brazil (2016)

8. Sentinel, T.M.: Phantom Auto' will tour city. In: Google News Archive, ed. (1926)
9. Carnegine Mellon University - The Robotics Institute, NavLab: The Carnegie Mellon University Navigation Laboratory. www.cs.cmu.edu/afs/cs/project/alv/www. Accessed Feb 2018
10. Kanade, T.: Autonomous land vehicle project at CMU. In: Presented at the 14th ACM Annual Conference on Computer Science (1986)
11. University of Parma: VisLab, Italy - Public Road Urban Driverless-Car Test 2013 - World premiere of BRAiVE, ed. (2013)
12. Parkinson, S., Ward, P., Wilson, K., Miller, J.: Cyber threats facing autonomous and connected vehicles: future challenges. IEEE Trans. Intell. Transp. Syst. **18**, 2898–2915 (2017)
13. Geng, H.: Connected vehicle. In: Internet of Things and Data Analytics Handbook, ed. Wiley Telecom (2017)
14. RAC Australia: Autonomous vehicle survey (2016). https://rac.com.au/-/media/files/rac-website/pdfs/about-rac/publications/reports/2016/autonomous-vehicles-survey.pdf
15. Ho, J.Y., Koh, W.Y., Veeravalli, B., Wong, J.W., Guo, H.: Secure sensing inputs for autonomous vehicles. In: Presented at the TENCON 2017, Penang, Malaysia (2017)
16. Foley & Lardner LLP: 2017 Connected Cars & Autonomous Vehicles Survey (2017)
17. Louridas, P., Ebert, C.: Machine learning. IEEE Softw. **33**, 110–115 (2016)
18. Tchankue, P., Wesson, J., Vogts, D.: Using machine learning to predict driving context whilst driving. In: Presented at the SAICSIT 2013, South African Institute for Computer Scientists and Information Technologists, East London, South Africa (2013)
19. Witten, I., Frank, E., Hall, M.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Burlington (2011)
20. Li, L., Wen, D., Zheng, N.-N., Shen, L.-C.: Cognitive cars: a new frontier for ADAS research. IEEE Trans. Intell. Transp. Syst. **13**, 395–407 (2012)
21. Ithape, A.A.: Artificial intelligence and machine learning in ADAS. In: Presented at the Vector India Conference, Pune, India (2017)
22. PSA Group: Car Easy Apps: Co-designing the connected car of the future (2016). https://www.groupe-psa.com/en/newsroom/automotive-innovation/car-easy-apps/
23. PSA Group: Car Easy Apps: PSA Peugeot Citroën's Application programming interface (2014). https://www.youtube.com/watch?v=3cTsNeKZDTU
24. Oracle: Architectural Strategies for Cloud Computing. Oracle White Paper in Enterprise Architecture, August 2009
25. Yousif, M.: The state of the cloud. IEEE Cloud Comput. **4**, 4–5 (2017)
26. Microsoft: Publish/Subscribe (2018). https://msdn.microsoft.com/en-us/library/ff649664.aspx
27. w3schools.com: JSON – Introduction (2018). https://www.w3schools.com/js/js_json_intro.asp
28. Flanders, J.: Service station - more on REST. MSDN Mag. **27**(7) (2009). http://msdn.microsoft.com/en-us/magazine/dd942839.aspx
29. W3C: Latest SOAP versions (2018). https://www.w3.org/TR/soap/
30. Dumas, B., Lalanne, D., Oviatt, S.: Multimodal interfaces: a survey of principles, models and frameworks. In: Lalanne, D., Kohlas, J. (eds.) Human Machine Interaction. LNCS, vol. 5440, pp. 3–26. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00437-7_1
31. Oviatt, S.L.: Multimodal interfaces. In: Jacko, A.S.J. (ed.) The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, pp. 286–304, 2nd edn. CRC Press (2008)
32. Peschl, M.F., Stary, C.: The role of cognitive modeling for user interface design representations. Mind Mach. **8**, 203–236 (1998)
33. Kelly III., J.E.: Computing, cognition and the future of knowing: how humans and machines are forging a new age of understanding (2015)