IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

VEHICLE SHARING SYSTEM WITH FLEXBLE ARCHITECTURE

S.R.Rohile*, D.S.Bhosale

* Department of Electonices and Telecommunication, Bhivarabai Sawant Institute Of Technology and Research, Pune, India

ABSTRACT

Vehicle sharing systems are the important in todays lifestyle. They should meet the features to answer the different user needs, and must be automated.As automatic system is used it avoid intermediaries between users and system. Finally, they can used for a wide variety of vehicles and on an open ownership model to become a viable alternative to private vehicles. This letter presents the solution devised in the to tackle this challenge. In particular, it focuses on the vehicle on-board units, for which a dedicated middleware is being developed. This system controll the various parameter of vehicle.The middleware allows developers to create applications that can be dynamically loaded/unloaded while preserving the needed safety levels of the vehicle motion functions.

KEYWORDS: Embedded architecture, middleware,Electrical vehicle

INTRODUCTION

Now days there are integration of vehicles that differ in type like cars, scooters, bicycles,with different technology of electric, hybrid, or with classical combustion engines and different ownership they can be publicly or privately owned, fully or partially shared within the same system. They must offer end-users common functions, services and interfaces [1]. To realize such systems, we must design suitable, innovative solutions must be found to design a hardware/software architecture that manages the on-board vehicle embedded control units and connects them to a back-end system that provides the mobility services to the final users.Mostly through personal mobile devices, such as smart phones, tablets, etc. This calls for solutions that meet the transparency and availability requirements within a distributed environment. This presents the technological solutions adopted to address the above issues.

Ths system aims at realizing an innovative vehicle sharing system based on zero-emissions electric vehicles. The system is innovative for many aspects: the heterogeneity of the vehicles involved, the openness of the ownershipmodel and the interaction protocols between the users and the system, which eliminate all intermediaries. In [3], readers can find an overview of the project and its relation with existing vehicle sharing systems. The project focuses on electric vehicles (EVs) as a means to achieve a sutaible model of mobility, in particular in the urban context.In vehicle sharing, possibly including different kinds of vehicles owned by different entities, is crucial to mitigate the obstacles to the distribution of EVs, e.g., their high price compared to that of similar combustion engine vehicles entails that EVs become economically viable only if they cover long distances during their lifetime, a target that is difficult to achieve outside of vehicle sharing solutions.

The technology developed within the system is not specific for EVs, aside from some low-level details concerning the monitored information (e.g., the battery state of charge), and can be adapted to traditional vehicles, too. It answers specific needs that arise in vehicle sharing systems with heterogeneous ownership models such as those that can maximize the utility of EVs. A key aspect of the system is that it does not involve intermediaries between users and vehicles: reserving, acquiring and releasing a vehicle are all done automatically through software running on the users' smartphones. Even keys for accessing vehicles are not required as they are substituted by software keys. This is reflected in Fig. 1, which shows the core of the architecture. It includes the the *green e-boxes (GEBs)*, and the users' smartphones on which the *GM client app* is installed. Communication and coordination among these

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

components are based on a novel middleware, which also allows applications to be dynamically loaded/unloaded on app.

Fig.1.Architecture of vehicle sharing system.

The *GM middleware* uses bluetooth channels (channel Fig. 1) to communicate with GEBs to manage the feet e.g., current GPS position, speed, state of charge,speed,door status) to the user. Finally, the uses of channel to dynamically add services to each GEB, by uploading new software modules that provide additional functions. This is a crucial aspect for a vehicle sharing system with different forms of ownership: as the fleet itself is reconfigurable, so must be the vehicle sharing functions available on vehicles. For example, a vehicle can be made available for rental through the system by its owner when she does not need it, but it reverts to being a private car when used by its owner in predefined time intervals. Notice that by using a direct (bluetooth or NFC) link between the user's smartphone and the GEB, this communication can occur at any time.

The same Bluetooth connection can be used to access, from the client app, information that is available on the GEB, such as trip statistics. To manage this complex system and meet the desired requirements of transparency and availability, abstraction mechanisms have been implemented, which allow the seamless use of technologically different vehicles, characterized by different available signals, different on-board networks, a different split between digital and analog signals, and so on. Furthermore, as we mentioned above, the GM middleware allows administrators to manage (add/remove) the features and services of GEBs while preserving the integrity of the low-level safety-related routines of the vehicles. The rest of the letter discusses these aspects in detail. Before presenting the solutions adopted in the project, we point out that systems that offer functions similar to those of GEBs are available on the market, such as OnStar, My-Chevroletand Viper. Among these, the Viper system is designed to be integrated into any type of vehicle, hence it is the most similar to ours. Nevertheless, the goal of these systems is to allow a vehicle owner to remotely control her vehicle by means of a smartphone, while the GEB has been developed to allow any vehicle to be inserted into a smartphone-based vehicle sharing system.

MATERIALS AND METHODS:

This system include main 3 parts user unit,on-board embedded unit and communication unit. The user unit means the user smartphone in which the app is install.The Bluetooth is used for communication.The android app displays the nessesory parameter like battery,door status,obstacle detected,authendicated person and also we can display other importand parameters.

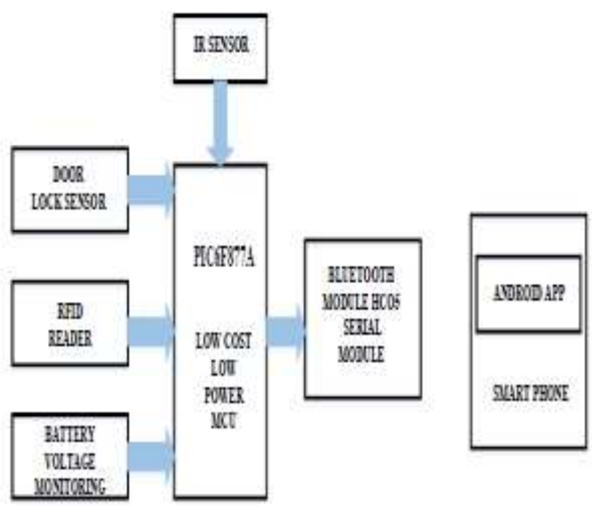

Fig.2. Block Diagram of vehicle sharing syste

On-board vehicle embedded control unit include the microcontroller with controles the system.The PIC6F877A microcontroller is used because it uses the low power and monitor the status of the all sensor connected to the input. A sensor is a device that measures a physical quantity and converts it into a signal which can be read by an observer or by an instrument. For example, a mercury thermometer converts the measured temperature into expansion and contraction of a liquid which can be read on a calibrated glass tube. Thr door lock sensor cheaks the status of all door of vehicle that can provide to the microcontroller.For authendication we uses RFID in which RFID tag is keep with the user and rfid reader is kept in vehicle.The vehicle user when come across the car then RFID reader reads the tag id and then only the door is open. So this can provide the safety.

Battery status cheaking is also done in this system. The battery votage is read by system and this can prevent the accidental battery discharge.The abstacle detection sensor is used which IR obstacle detection sensor from Robot Base is designed for robust operation. Using a modulated LED light source of 40Khz, and a sensitive tuned IR receiver, the module effectively discriminates against most indoor light sources. The board supports an adjustable gain potentiometer, as well as an adjustable threshold potentiometer. The sensor module outputs a logic high if no obstacle is detected, and outputs a logic low if an obstacle is detected. A convenient LED is supported which provides an immediate status of the sensor. The detection range of the IR sensor is between 2-40cm, with an operating voltage of 3.5-5V. The IR detector is compatible with most Arduino Boards as well as several other microcontrollers. See datasheet for more details of this exciting sensor product.

All outputs of the sensor network provide to the signal conditioning which convert the output signal into the format that can process by the controller. Controller process the signal and provide the require signal pins high. This status is transfer to user smartphone'. We use the Bluetooth for wireless communication .

RESULTS AND DISCUSSION

The vehicle sharing system is used in wide range as there is need of safe and smart vehicle system. EMBEDDED GREEN E-BOX MIDDLEWARE is man part osf system. The functions realized by the GEB can be divided in two categories: core operations of the GMVSS (*e.g.*, user authentication, vehicle monitoring, etc.), and optional functions that, though not essential, provide added value to GM users (*e.g.*, commercial and traffic information). Whereas core operations are known from the design phase of the GMVSS and change infrequently, optional functions could be added or removed after the system deployment (e.g., new commercial agreements between GM and its partners). Consider a scenario such that, after the GMVSS has been deployed, a new commercial deal is struck with and provider who developed an application to notify users in real time of discounts available in shops close to their current location. In this case, the application should be installed on all GEBs from the beginning of the commercial agreement, until its end. In addition, the application should be able to read the current location of the vehicle from the GEB, to provide users with accurate information. This suggests that the GEB should have the possibility of dynamically loading and unloading applications which have suitably controlled access to vehicle information retrieved by the GEB itself.

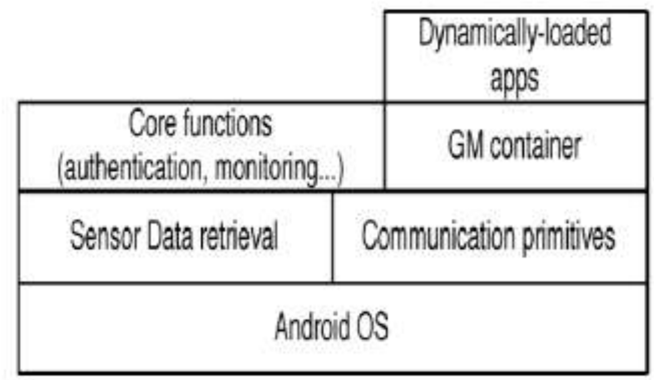

Fig.3. Layered structure of the functions of the GEB.

Thus, the GEB software should be designed to enable the kind of scenarios outlined above. In particular, it should have a modular structure, where each module oversees a cohesive set of GEB functions (*e.g.*, vehicle data retriev); it should also provide a set of primitives and mechanisms that make it possible to build and manage applications in a dynamic way. In the rest of this section we discuss the choices made and the mechanisms realized to achieve these goals. TheGEB software is built on theAndroidOS according to the layered structure. At its core are the modules providing the basic mechanisms for the retrieval of data from the vehicle sensors and for the communication with theuser. They are used to realize the core functions of the GEB, such as, for example, vehicle monitoring, and they are made available to third-party applications through a component called GMContainer. The GMContainer is part of the GM middleware (GMM) infrastructure that allows system administrators to load and remove applications from GEBs dynamically. The main components of the GMM are the Code Server, which resides on the GMC, and the GMContainer part of the GEBs.

1) Code Server: The code server (CS) is a module of the GMC that allows trusted parties to upload their applications, to verify them and to distribute them to GEBs through the GMM. It also allows administrators to get the current list of devices running a certain application, to stop any running instance and uninstall it, or to deploy it a second time. Applications are uploaded as signed JAR files, which are verified for authenticity before being made distributable.

2) GMContainer: The main responsibility of the GMContainer is to allow green move applications (GMAs) to be added and removed at run time. GMAs are downloaded when requested and their code is linked to the GEB code at run-time.The application entry point is then executed and a reference to the latter is kept by the GEB for future unloading. As depicted in Fig. 4, GMAs are provided appropriate APIs for interacting with the GMecosystem. They allow applications to:

• display text or images to the end user (ads, messages);

• communicate with other GM components (including the GMC) by sending and receiving events according to a publish- subscribe paradigm;

• read sensors data from the GEB embedded board, e.g., the GPS position and the vehicle speed. GMAs are independent of one another, although no restriction is enforced on their interaction. They are provided to GEBs via

the GMC. Their code is downloaded and executed, and then disposed of when the application no longer needs to be active, or system resources need to be freed.

3) Implementation: GMAs are Android components which adhere to the single entry point convention enforced by the GMContainer. This means that applications could be coded in any language which can be run on top of the Android Java Virtual Machine (Dalvik), such as Ruby or Python. The current implementation assumes that the application code is sealed in a standard JAR file. CS allows a publisher to upload JAR files, to verify their integrity according to the publisher's certificate and to select the specific GEBs that have to receive the application, e.g., all of them or only those installed on a specific vehicle type, etc. The CS leverages the T-Rex publish-subscribemiddleware, see [14], to communicate with GEBs and send them the "load code" message, which contains the references to the file to load and the entry point of the application.

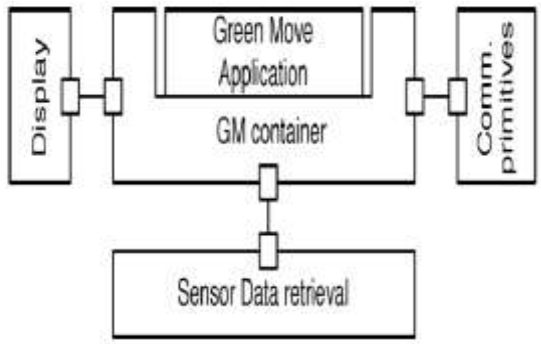

Fig.4. Interfaces of the GMContainer.

The GMContainer is implemented as an Android Service. It provides a primitive that allows GMAs to access the API through which sensor data can be retrieved. Similar primitives allow GMAs to access the communication infrastructure used by the GMContainer to communicate with the GMC, and the abstraction layer to access the display to show text and images. Every GMAimplements an interface offering a primitive to stop the application.

Table 1.Parmeter on android app

Parameters	Use
Battery voltage	Voltage is less or not.and if low need tto charge battery
Authendication	Person allowed to open door or not
Obstacle detection	Any obstacle present at back, left and right side of vehicle at the the of parking.
Door status	Door are properly close or not. It indicate if any door is open.

CONCLUSION

The architecture of the system was designed at bothat the hardware and at the software level, to provide the flexibility that is necessary to manage vehcle sharing system to manag different applications.It avoid the intermediate between user and system.. As such, it is a blueprint for a wide class of systems that include, but are not limited to, next generation vehicle sharing systems. This letter focused in particular on the

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

structure of the on-board unit, communication between user and system, which acts as the interface between the system and the vehicle, the middleware infrastructure that has been designed to simplify architecture to send information to user, also we can send it to other vehicles through the event-distribution component. As such, it provides capabilities that could be used to develop applications that are based on interactions among vehicles.

REFERENCES

1. Kalogeras, P. Foundas, M. Georgoudakis, K. Charatsis, and P. Konstantinopoulos, “Integrated system for smart transport services,” in *Proc. IEEE Conf. Emerg. Technol. Factory Autom.*, 2009, pp. 1–6.
2. A. Osório, H. Afsarmanesh, and L. Camarinha-Matos, “Towards a reference architecture for a collaborative intelligent transport system infrastructure,” *Collab. Netw. a Sustainable World*, vol. 336/2010, pp. 469–477, 2010.
3. G.Alli *et al.*, “Greenmove: Towards next generation sustainable smartphone- based vehicle sharing,” in *Proc. SustainIT*, 2012.
4. Chevrolet 2013 [Online]. Available: http://www.chevrolet.com
5. Directed Electronics 2013 [Online]. Available: http://www.viper.com/
6. Tazzari GL s.p.a. 2012 [Online]. Available: http://www.tazzarizero. Com
7. W. J. Fleming, “New automotive sensorsa review,” in IEEE Sensors Journal, vol. 8, no. 11, pp. 1900–1921, 2008.
8. U. Lee, E. Magistretti, M. Gerla, P. Bellavista, and A. Corradi, “Dissemination and harvesting of urban data using vehicular sensing platforms,” in IEEE Transactions on Vehicular Technology, vol. 58, no. 2, pp. 882–901, 2009.
9. C. Palazzi, F. Pezzoni, and P. Ruiz, “Delay-bounded data gathering in urban vehicular sensor networks,” in Pervasive and Mobile Computing, vol. 8, no. 2, pp. 180–193, 2012.
10. A. Mednis, A. Elsts, and L. Selavo, “Embedded solution for road condition monitoring using vehicular sensor networks,” in the 6th IEEE International Conference on Application of Information and Communication Technologies (AICT’12), 2012, pp. 1–5.
11. ESTRIMA SRL, 2012 [Online]. Available: http://www.estrima.com
12. Piaggio & c s.p.a. 2012 [Online]. Available: http://www.piaggio.com
13. S. Di Martino, C. Giorio, and R. Galiero, “A rich cloud application to improve sustainable mobility,” *Web Wireless Geograph. Inf. Syst.*, vol. 6574/2011, pp. 109–123, 2011.
14. M. Conti, D. Fedeli, and M. Virgulti, “Bluetooth for electric vehicle to smart grid connection,” in *Proc. IEEE Intell. Solutions Embed. Syst.*, vol. 1, pp. 1318–.
15. A. Dardanelli, M. Tanelli, B. Picasso, S. Savaresi, O. di Tanna, and M. Santucci, “A smartphone-in-the-loop active state-of-charge manager for electric vehicles,” in *Proc. IEEE ASME Trans. Mechatron.*, vol. 17, no. 3, pp. 454–463.
16. G. Cugola and A. Margara, “Complex event processing with T-REX,” *JSS*, vol. 85, no. 8, pp. 1709–1728, 2012

AUTHOR BIBLIOGRAPHY

	**Miss.Suvarna R. Rohile** Received the bachelor’s degree in E&TC engineering from University of pune India. Now pursuing the Master degree in the VLSI and embedded system at pune University,India.Research interests are in Embedded system.
	**Prof.Dilip. S. Bhosale** Received mster degree in E&tc in Mumbai university,India. And work as a professor at collage BSIOTR,Pune,India. Area of intrest is embedded system and image processing.