João C. Ferreira, V. Monteiro, João L. Afonso, "Real-Time Information Extraction of an Electric Vehicle," Annual Seminar on Automation, Industrial Electronics and Instrumentation 2012 - SAAEI'12, pp.488-493, Guimarães, Portugal, Jul. 2012, ISBN: 978-972-98603-5-5.

Real-Time Information Extraction of an Electric Vehicle

João C. Ferreira[1], Vítor Monteiro[2], João L. Afonso[2]

1 – Centro Algoritmi – ADEETC – ISEL – Lisbon, Portugal
2 – Centro Algoritmi – University of Minho – Guimarães, Portugal

jferreira@deetc.isel.ipl.pt, vmonteiro@dei.uminho.pt, jla@dei.uminho.pt

***Abstract* — In this paper is presented the development of a project to extract, in real-time, information's related with an Electric Vehicle (EV). This project was elaborated to extract data from an EV battery charging device developed at the University of Minho, and from an EV prototype, the VEECO (*Veículo Eléctrico ECOlógico* – Ecologic Electric Vehicle), developed in a cooperation project of ISEL (Lisbon Superior Institute of Engineering) and the Portuguese company VE. The main goal of this project consists in collecting and transmitting the extracted data to inform the EV driver about the performance and the real behavior of the EV. Thereby, it is created an open interface to manage, in real-time, the main data related with the EV, as the batteries SoC (State-of-Charge), the EV speed, and internal temperatures (like the temperatures of the batteries, motor and power electronics inverter), as well as to control the start and stop of the batteries charging process, and to optimize the charging program (to define the best algorithm to preserve the batteries lifespan). This interface also controls the discharging process of the batteries, in order to make possible to deliver back to the electrical power grid part of the stored energy in the batteries, which is defined by the concept Vehicle-to-Grid (V2G). In the paper are presented and described the two main parts of this work: the real-time information extraction system and the charging device.**

Keywords - Electric Vehicles; On-Board Battery Charging Device; V2G - Vehicle-to-Gri; Real-Time; CAN-bus; Information Extraction.

I. Introduction

Automobile plays an important role in the transportation sector, and consequently into the transport in cities. Thereby, new challenges appear nowadays with the recent introduction of mass production commercial Electric Vehicles (EVs), the so-called city electric vehicles. In [1] is presented a concrete case of a technology to integrate EV with Smart Grids, and in [2] is approached the future of the electrical grid focusing the impact of the PHEV (Plug-in Hybrid Electric Vehicle) in a Smart Grid. The integration of EVs in the electrical power grid is approached in several papers in the literature [3]-[5], as the integration of PHEVs in different part around the world, like in Canada [6], Sweden [7], and China [8].

Typically, EVs have autonomy of less than 160 km, and the drivers will have to cope with a new problem: EV charging takes time and drivers need to deal with limited electric range, this problem is designated in the literature as the "EV range anxiety" [9][10].

The success of EV in part is going to depend on how comfortable people are in going where they want to go, without running out of charge, and without having to go through some recharging process that will take them a long time, and impact into their ability to use the vehicle. So, taking out EV information in real time, such as SoC (State-of-Charge), energy transactions, and other Electric Vehicle related events, plays an important role in the success of EVs.

In order to maximize the EV batteries lifetime, the batteries charging process should be performed taking into account the experience of the drivers and their behavior while driving (aiming adjust the batteries charging algorithm to not degrade the batteries). On the other hand, if the EV allows the functioning as V2G (Vehicle-to-Grid), the charging and discharging processes, besides the parameters associated with the drivers, should be correlated aiming the new paradigm of Electrical Markets (EM), with a smart EV charging system [11].

At the ISEL (*Instituto Superior de Engenharia de Lisboa* – Lisbon Superior Institute of Engineering), the project Electric Vehicle - Management System Information (SGI-EV - *Sistema para Gestão de Informação num Veículo Elétrico*) had as main objective the design and the implementation of a computational system to collect information, and to communicate useful data to monitoring the EV behavior. The system performs the data acquisition through the existing physical interface in the EV, or through some device installed in the vehicle for this purpose. To minimize the dependencies of the proprietary technologies was considered a multi-channel architecture which is mainly focused in the development of a Bluetooth adapter for CAN-bus, in particular to the Electric Vehicle VEECO (*Veículo Elétrico ECOlógico*), developed in a cooperation project of ISEL and the Portuguese company VE.

At the University of Minho was developed a power electronics device to charge and discharge the batteries of an EV. With this device, that allows functioning in bidirectional way as V2G, it is possible to control the batteries charging and discharging processes, using appropriate algorithms in order to preserve the batteries lifespan, and at the same time to maintain the power quality of the electrical grid [12].

II. INFORMATION EXTRACTION FROM THE EV AND FROM THE CHARGING DEVICE

From the EV can be obtained relevant information to be used by the driver to determine and analyze the EV performance, mainly, the variables associated with the batteries as State-of-Charge (SoC), State-of-Health (SoH) and range of temperatures, and on the other hand the variables associated with the dynamic behavior of the EV as the speed, accelerations, temperatures or the energy that can be recovered from the regenerative braking. All of these information's are sent to the control unity, which associates the driver requests and the charging log file, and determine the better use of the vehicle and the charging device, enabling an intelligent and opportunistic EV charging management for each driver according to her/his needs.

The analysis of the data transactions can also be useful for future processes of charging (G2V – Grid-to-Vehicle operation) or discharging (V2G – Vehicle-to-Grid operation), taking into account a smart charging strategy in order to combine the electrical grid limitations and requirements [13][14], and the best prices for buying and selling electrical energy, in a established environment of Smart Grids.

As illustrated in Fig. 1, the main information to be extracted from the EV is grouped in three areas: the Charging Log File; the Information from EV; and the Charging Device Control. These data can be stored on central server, and through an internet communication, the driver can check remotely the charging process, and interact with it.

A. *Information Repository*

The Central Information Repository stores the information related with the EVs, namely: (1) the profile of the drivers; (2) the electricity transactions between the EV and the electrical grid; (3) the electricity prices; (4) the weather information; (5) the driving parameters; and (6) other information related with EVs. This Central Information Repository is based on XML files and on a MySQL database. Since, in a first stage of this project, our goal was the creation of a prototype, not a commercial application, this subject was not tuned for best system performance, but only for testing purposes.

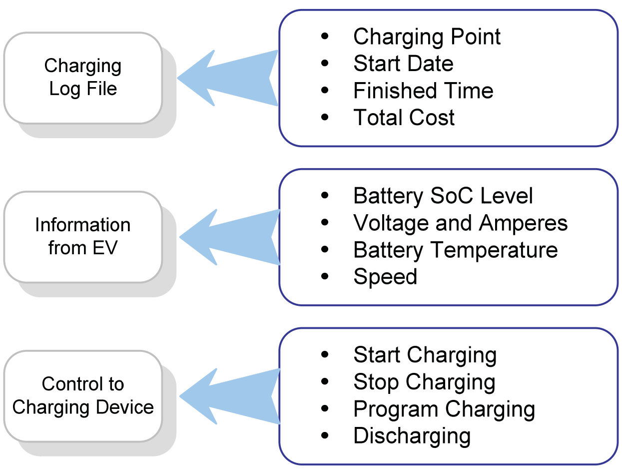

Figure 1. Main information to be extracted and the commands to be performed on the charging device.

B. *Development of the OBU (On Board Unit)*

This task consists in the development of a working prototype of an On-Board Unit (OBU), including hardware and software, to be installed on-board in the vehicle in order to provide locally (in the EV) and remotely (to mobile devices applications) relevant data communications. The OBU device is based on a microcontroller that integrates CAN-bus communication, Bluetooth, GSM/GPRS (Global System for Mobile communications / General Packet Radio Service) and GPS (Global Positioning System).

In this way, the implementation of the CAN-bus protocol allows requesting and receiving data from the vehicle, and with the available OBU wireless communication interface, it is possible to report both locally and remotely the data through Bluetooth and/or GSM/GPRS technologies, respectively. Moreover, Bluetooth allows the OBU integration with mobile equipment, such as a mobile/smart phones. Additionally, and having knowledge of the current coordinates (GPS receiver) of the EV, the OBU is able to make the best decisions to the EV. (to indicate the public charging points) GPRS allows the development and implementation of the OBU update without wired communication, increasing the easiness with which software updates are made.

C. *Process for Information Extraction*

In this item is described the proposed architecture to collect and communicate the attribute values associated to the EV. Thus, in order to extract the required information from the EV, it is proposed the architecture shown in Fig. 2.

a) Attributes Layer Access

The Attributes Layer Access, as shown by the architecture in Fig. 3, includes the logic to read the attributes related with the EV (e.g., the speed). This architecture receives as input a reference to the attribute to be read (the source of attributes is the EV that has an available interface for this propose), and it is sent as output the requested attribute value. Besides, this architecture can be implemented to read attributes from other sources, both internal (e.g., the timer value used in the system) or external (e.g., the GPS sensor).

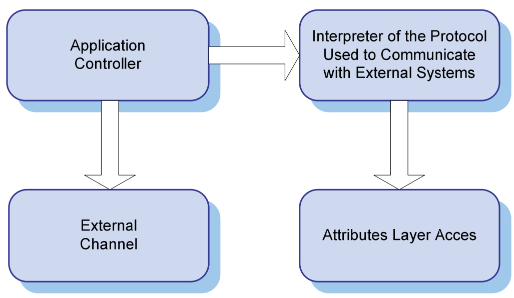

Figure 2. General block diagram of the proposed architecture to extract information's from the EV.

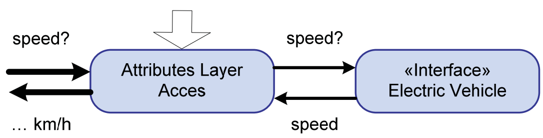

Figure 3. Architecture of the Attributes Layer Access to read attributes from the EV.

b) External Channel

The external channel works as a communication bridge between the client and the server. In this way, it is used to exchange AQP (Attribute Query Protocol) messages between the client and the server.

In this project, was used the RN-42 module to the Bluetooth communications, and it works in two modes, abstracting the programmer of the Bluetooth protocol: commands (the module is configured through ASCI commands provided by the manufacturer), or data (the module works as a connection to the AQP client). Independently of the mode, the information is shared between the client and the server using UART (Universal Asynchronous Receiver Transmitter) as shown in Fig. 4.

c) Interpreter of the Protocol used to Communicate with External Systems

The Interpreter of the Protocol used to communicate with external systems receives as input the request of values of attributes, which is sent by an external system, and sends as output the value of the attribute, in a pre-defined format.

d) Application Controller

This block is sensitive to the messages received by the channel to the outside, and it is responsible to deliver them to the interpreter block, and sends the response generated for the outside channel where the request was originated.

In this project is proposed a protocol focused on obtaining attribute values, which it is abbreviated by Attribute Query Protocol (AQP). The protocol was designed in order to access a source of abstract attributes, which makes it independent of the source. To read values of attributes it is necessary exist an AQP client (entity that needs the attributes), a server (interface to the source of attributes), and a context (list of attributes descriptions - each attribute is characterized by an identifier, a name and its type). The context for each server is fixed and previously known by the client. Thus, an AQP client can communicate with multiple servers, only changing the context of the attributes. As shown in Fig. 5, the communication is done through messages composed by a header (mandatory) and the content (optional). The header is composed by the code (request or response), by the identifier (type of operation) and by the dimension, where the first two fields identify the type of message.

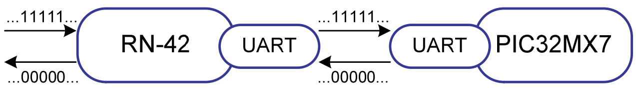

Figure 4. Shared information using UART.

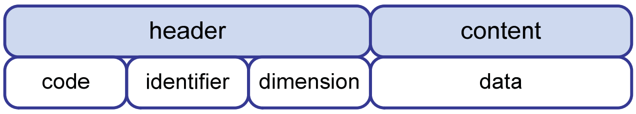

Figure 5. Structure of an AQP message.

III. AQP Server Implementation

In this section are described the details of the implementation of the AQP (Attribute Query Protocol) server, with access to the values of the sensors present in the EV. The AQP server realizes the following four tasks:

i. Receive messages from an AQP client;
ii. Interpret the messages and generates the response;
iii. Access the EV interface in order to ask the sensor values;
iv. Respond to the client request.

Taking into account that the referred tasks above do not require much computing power, aiming to reduce the cost and the physical size, the AQP server was implemented in a microcontroller of the architecture PIC32MX7 from Microchip. The development tools include the prototyping board Explorer 16, also from Microchip, and the tool chain provided by the same manufacturer, to make development in this architecture using the C language.

On the other hand, the chosen protocol for exchanging AQP messages was the Bluetooth module (with UART interface), through the RN-42 module, of the Roving Networks manufacturer. The access to the EV interface is done through the communication controller UART (integrated in the microcontroller), complemented with an UART converter to RS-232 (because the microcontroller do not support RS-232 communications).

In Fig. 6 is shown the flux diagram of the information of this system. In addition to the standard C libraries, and the abstractions to the peripherals, available by the used compiler, were also created libraries to description the variables.

A. Interpretation of AQP Requests

After receiving an AQP request, the server interprets this request and generates a response. Thus, aiming the Interpretation of the AQP Requests, was developed a stack for an AQP server, which takes as input a message received by the client and sends as outputs the respective responses. In Fig. 7 is shown the interaction of the AQP server with the access layers to attributes sources, in this case, to request the speed value of the EV (with the attribute identifier 1).

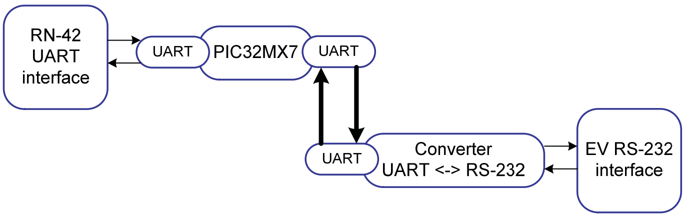

Figure 6. Flux diagram of the information between the UART RN-42 interface and the RS-232 EV interface.

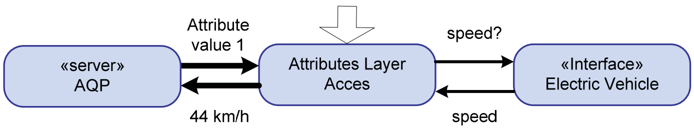

Figure 7. Interaction of the AQP server with the access layers to attributes sources.

The AQP server is the responsible to validate the received messages and, when the messages are valid, it generates the correct answers. In order to know how to interact with the attributes sources of the AQP context, each attribute has one reference to its layer of access to an attribute source (in this project only exist the EV). The layers of access to the attributes sources has as input the identifier (in the AQP context) of the attribute to be read and as output its value.

B. *Access Layer to the Attributes Source*

Due to commitments of confidentiality it is not possible to disclose the protocol used to interface with the EV. Only can be referred that it was used the RS-232, characterized by having a low rate, where the latency of a request for an attribute is approximately 500ms. The interface is synchronous, i.e., to answer it waits the request of any module about the value of any attribute, as illustrated in Fig. 8.

C. *Library to Collect Values*

In this item is described the AttributeSourceAPI library, which was developed in Java, and that allows collecting values from sources of attributes. This library contains the implementation of the access layers to the sources of attributes, aiming to collect its values. In order to know the involved attributes during the communication with the source, it is necessary to define a domain of attributes, which contains a combination of attributes to identifiers. In the context of this project, were developed the necessary layers in order to allow communicating and collecting values from an AQP server through Bluetooth. Taking into account the communication with the AQP server through Bluetooth, it was developed the class BluetoothAQPAttributeSource that uses the library BlueCove [15]. This class derives from AttributeSourceAQP and contains the implementation methods of the interface AttributeSource, which are not implemented in the abstract class AttributeSourceAQP. In Fig. 9 is shown the class hierarchy referred above.

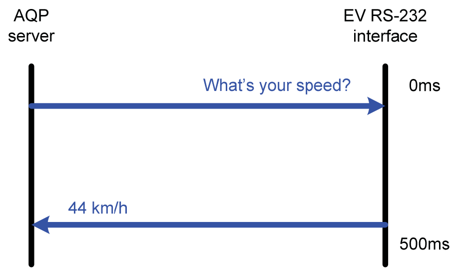

Figure 8. Functioning of the RS-232 interface for the EV interaction.

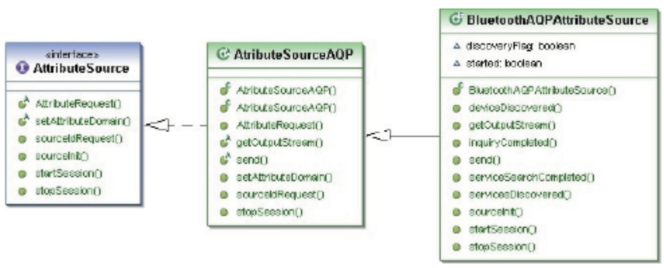

Figure 9. Class hierarchy of the attributes source.

IV. Information Extraction From the EV Charging Device

The presented charging device was developed aiming to charge the batteries of the EV with the correct algorithm, optimizing their lifespan, and consequently ensuring more reliability to the driver. With the developed charging device, the information related with the EV during the charging (G2V) or discharging (V2G) processes (as currents and voltages in the batteries along time) can be provided to the OBU (On-Board Unity) of the EV, through a wired communication.

A. *Charging Device*

The batteries charging device that was developed at the University of Minho is a power electronics equipment that converts the AC voltage from the electrical grid into a controlled DC voltage to charge the batteries. Besides the process of the batteries charging (G2V operation), this equipment allows delivering part of the stored energy in the batteries back to the electrical grid (V2G operation). In both modes of operation the waveform of the AC current is sinusoidal, and the power factor is unitary in G2V operation, contributing to preserve power quality. In Fig. 10 is shown the developed charging device. As referred before, this equipment charges the batteries with the correct algorithm to optimize their lifespan. During the tests of this charging device were used lead-acid batteries. The charging algorithm used consists in constant current followed by constant voltage. However, the charging device allows working with other algorithms, as constant voltage, constant current, or any other, taking into account the chemical technology of the batteries.

The developed charging device is composed by two main parts: the power electronics converter and the digital control system. The power converter consists is a bidirectional converter that uses four power semiconductors (IGBTs FGA25N120ANTD 25 A – 1200 V), two snubber capacitors (1 uF – 400 V), an inductance (5 mH – 10 A) to interface with the electrical grid (AC side), and a capacitor (4.7 mF – 450 V) in the batteries side (DC side). The developed digital control system is composed by a microcontroller (PIC32MX360F512L), a Digital Analog Converter (DAC – DAC712P), voltage and current sensors (LEM – Hall Effect Sensors), a command circuit, drivers to the IGBTs, and circuits for signal conditioning and errors detection.

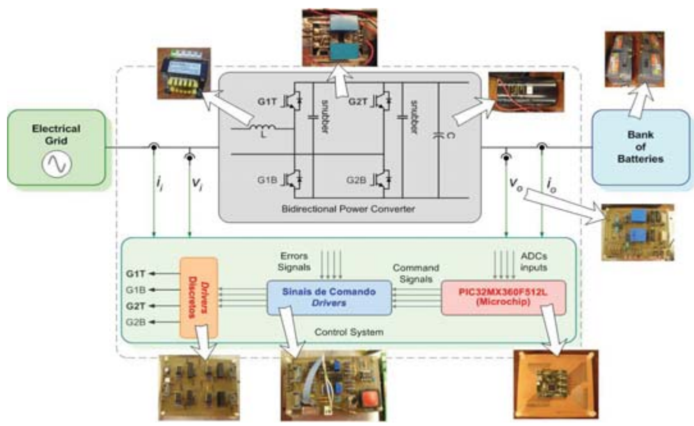

Figure 10. Conceptual diagram of the developed charging device.

B. Development of the RSU (Road Side Unit)

The developed RSU (Road Side Unit) is a prototype that combines hardware and software to be connected to the batteries charging device, aiming to provide useful data (as the batteries SoC, the charging time, the charging process efficiency, as well other data) to the EV driver. The RSU device integrates Bluetooth and GSM/GPRS communications and the stored information's about the charging system. The available wireless communications interfaces on the RSU allow report locally and remotely the data related with the charging system. As for the OBU, the Bluetooth communication allows the integration of the RSU with mobile devices, such as a mobile/smart phone, and the GPRS allows the development and implementation of the RSU update without wire, increasing the easiness of software updates.

The microcontroller, implemented in the digital control system of the charging system, has as main function control the charging and discharging processes. Thereby, the data associated with these processes, as the evolution of the voltage, the current and the temperature are managed by the microcontroller, and they can be provided to the RSU in order to deliver more information to the EV driver. Nevertheless, in this project, the interface between the developed digital control system of the charging device and the RSU was not yet developed; however, the goal is develop the interface through RS-232 communication.

Besides the data communication, the RSU can also communicates with the microcontroller in order to define the start and stop of the charging process, the charging program (to define the charging algorithm taking into account the chemical technology of the batteries), and to control the energy delivered back to the electrical grid during the discharging process.

V. APPLICATION TO EXTRACT EV INFORMATION

To handle with the extracted information was developed an application in Java language, which uses the library AttributeSourceAPI. Taking into account that the values of the attributes in the EV have different variations (e.g. the velocity varies more than the temperature), were assigned different frequencies to update the attributes, in order to make possible the setting of the refresh time of each attribute. These frequencies, as well the identifiers of each attribute can be found in the configuration files (.properties) external to the application.

To show the collected data to the user, was developed a graphical interface that contains a virtual instrument panel and some graphics related with the behavior of the instantaneous consumption and the batteries voltage. This interface was developed using the IDE NetBeans, due to the facility to create new graphical components. To the creation of some specific components was used the library JFreecharts [16].

In order to inform the user about the domain of attributes to do the interaction with the vehicle, is presented in Fig. 11 the information's table of the attributes domain. All of the collected attributes are presented in Fig. 12. As presented in Fig. 13, it is also possible to obtain the instantaneous consumption and gets the historical data.

Session Attributes

Attribute ID	Attribute	Current Value
1	Speed	
2	Handbrake	
3	BatteryCharge	
4	CurrentRMS	
5	BatteryVoltage	
6	Temperature	
7	MotorTemperature	
8	ForwardSwitch	
9	BackwardSwitch	

Figure 11. Attributes domain used to extact the informations from the EV.

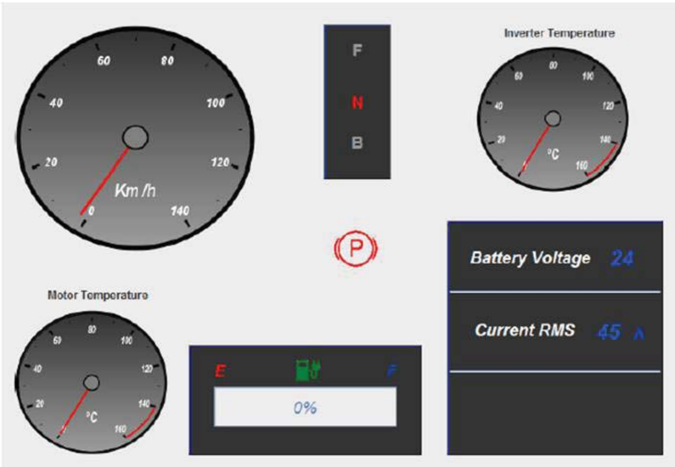

Figure 12. Visualization of EV extracted information in an external laptop.

Figure 13. Evolution of the current along the time.

VI. CONCLUSION

In this paper was presented a developed project with the objective of extracting data from an Electric Vehicle (EV) battery charging device, developed at the University of Minho, and from an EV prototype, the VEECO (*Veículo Eléctrico ECOlógico* – Ecologic Electric Vehicle), developed in a cooperation project of ISEL (Lisbon Superior Institute of Engineering) and the Portuguese company VE.

The main goal of this project is to inform the EV driver about the performance and the real behavior of the EV. In this way, relevant data related with the EV are extracted and transmitted to the driver, and it is provided an interface to manage, in real-time, the data associated with the EV, as batteries SoC (State-of-Charge), vehicle speed, and EV operating temperatures. On the other hand, the driver can also deliver control commands to the EV, mainly, during the charging (G2V – Grid-to-Vehicle operation) and discharging (V2G – Vehicle-to-Grid operation) processes. For instance, the driver can start and stop the charging process, and even modify the charging program (to define the correct algorithm aiming to reduce the charging time and to preserve the batteries lifespan). The driver can also set commands to control the amount of the stored energy in the batteries that will be delivered back to the electrical grid during V2G operation.

In practice, it would possible to collect the EV data with a non-intrusive mode, via Bluetooth and using equipment for personal use. However, the majority of the manufacturers do not allow this information interpretation. This work can be used for different EV architectures taking into account different information interpretation (the interpretation of the information depends on each manufacturer). This work can be used for different EV architectures taking into account different information interpretation (the interpretation of the information depends on each manufacturer).

In the paper were presented and described the two main parts of this developed project: the real-time information extraction system and the charging device for EVs.

As future work it is suggested the development of an AQP (Attribute Query Protocol) client to mobile platforms (e.g., PDA, Smartphone), in order to increase the comfort of utilization of the vehicle to the EV driver. Other possibility is the evolution of the AQP protocol, in order to be adapted to a wider range of channels to acquire more attribute values.

Acknowledgments

This work is financed by FEDER Funds, through the Operational Programme for Competitiveness Factors – COMPETE, and by National Funds through FCT – Foundation for Science and Technology of Portugal, under the project PTDC/EEA-EEL/104569/2008 and the project MIT-PT/EDAM-SMS/0030/2008.

References

[1] S.S.Hernandez, P.P.Galindo, A.Q.Lopez, "EPV PROJECT. Technology to integrate EV inside smart grids," IEEE 7th International Conference on the European Energy Market (EEM), 2010.

[2] A.Ipakchi, F.Albuyeh, "Grid of the Future," IEEE Power & Energy Magazine, March/April 2009.

[3] S.Shao, M.Pipattanasomporn, S.Rahman, "Challenges of PHEV Penetration to the Residential Distribution network,"IEEE PES Power & Energy Society General Meeting, pp.1-8, 2009.

[4] J.A.P.Lopes, F.J.Soares, P.M.R.Almeida, "Integration of Electric Vehicles in the Electric Power System," Proceedings of the IEEE, Volume:99, Issue:1, pp.168-183, 2011.

[5] L.Zhao, S.Prousch, M.Hubner, A.Moser, "Simulation Methods for Assessing Electric Vehicle Impact on Distribution Grids," IEEE PES Transmission and Distribution Conference and Exposition, pp.1-7, 2010.

[6] G.Joos, M.R.Dubois, "Integration of PHEVs and EVs Experience from Canada," IEEE PES Power and Energy Society General Meeting, pp.1-5, 2010.

[7] L.Bertling, O.Carlson, S.Lundmark, D.Steen, "Integration of plug in hybrid electric vehicles and electric vehicles - experience from Sweden," IEEE PES Power and Energy Society General Meeting, pp.1-3, 2010.

[8] Y.Song, X.Yang, Z.Lu, "Integration of Plug-in Hybrid and Electric Vehicles Experience from China," IEEE PES Power and Energy Society General Meeting, pp.1-6, 2010.

[9] Maria Nilsson, "Electric Vehicles: The Phenomenon of Range Anxiety," ELVIRE - ELectric Vehicle communication to Infrastructure, Road services and Electricity supply, June 2011.

[10] A.Wiederer, R.Philip, "Policy options for electric vehicle charging infrastructure in C40 cities," Transportation, Clinton Climate Initiative, 2010.

[11] J.C.Ferreira, V.Monteiro, J.L.Afonso, A.Silva, "Smart electric vehicle charging system," IV IEEE Intelligent Vehicles Symposium, pp.758-763, June, 2011.

[12] V.Monteiro, J.C.ferrreira, J.G.Pinto, D.Pedrosa, J.L.Afonso, "iV2G Charging Platform," 13th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp.409-414, 2010.

[13] J.C.Ferreira, V.Monteiro, J.L.Afonso, "V2G smart system: An intelligent system to the electric vehicle charging problem," IEEE CISTI 6th Iberian Conference on Information Systems and Technologies, pp.1-4, June 2011.

[14] V.Monteiro, J.L.Afonso, J.C.Ferreira, "An Agent Model for the Simulation of Electrical Vehicle Charging Management," IEEE CISTI 6th Iberian Conference on Information Systems and Technologies, pp.1-4, June 2011.

[15] Bluecove JSR-82 Implementation. Avaivable (18 September 2011) [online]. Available: http://snapshot.bluecove.org/

[16] JFreeCharts (18 September 2011) [online]. Available: http://www.jfree.org/jfreechart/

[17] Geada M., Nunes R., "SGI-VE: Sistema para Gestão de Informação num Veículo Eléctrico," Final Year Project at ADEETC-ISEL, Lisbon, 2011.