Artemisa: Using an Android device as an Eco-Driving assistant

V. Corcoba Magaña and M. Muñoz Organero

***Abstract*—The eco-driving concept consists on applying a set of rules while driving to save fuel. Eco-driving has acquired great importance in recent years because it is a way to reduce energy consumption that can be applied to any type of vehicle.**

However, for these rules to be applied requires a process of continuous learning and motivation. For this reason, many eco-driving assistants have emerged. The problem of these assistants is that they are dependent on the model of vehicle, expensive and imprecise.

In this context, this paper presents a novel efficient driving assistant that uses the features of the Smartphone to accurately model the driver's driving style from the point of view of energy consumption and generate eco-driving tips to correct the bad driver's driving habits.

***Index Terms*—Eco-Driving, Android OS, Driving Assistant, Expert System, OBD, Data Acquisition System**

I. Introduction

The vehicles are a major cause of death worldwide. We might think that most of those deaths are due to traffic accidents but this is not true. The main cause of death is a consequence of the gaseous pollutants emitted by vehicles [1] [2]. On the other hand, energy resources are scarce and expensive, so its use should be minimized. For this reasons, the reduction of energy consumption is a priority for governments and vehicle manufacturers.

Manufacturers have taken actions to save energy (engines with lower consumption, reduction of weight and improvement in the aerodynamics of the vehicle). However, in order to further reduce energy consumption the cooperation of the driver is needed. If the driver takes a style of efficient driving, he could save from 10 to 25% of fuel [3] [4] [5].

Artemisa´s eco-driving assistant presented in this paper tries to help the driver to take an efficient driving style. We will evaluate the driver's driving style and according to the result show eco-driving tips.

There are many proposals with the same purpose as Artemisa, but they have several drawbacks:

- The assessment of the driver driving style is not accurate because it does not consider the environmental variables such as the state of the road or weather conditions.
- They require expensive additional hardware installation.
- They are not standard, can only be applied to a particular model of vehicle.

The solution we propose is based on the use of mobile devices running the Android OS where the eco-driving assistant is executed. We will also use a Bluetooth module that connects to the vehicle´s diagnostic port. This Bluetooth module allows sending the vehicle telemetry to the smartphone. The eco-driving assistant uses the information obtained through the diagnostic port with the smartphone´s information (Sensors, GPS and Internet connection presented) to accurately model the driver´s driving style from the point of view of energy consumption.

This approach can be used on any model of vehicle and does not require any special device installed in the vehicle. Moreover, its cost is reduced, the Bluetooth module costs about 50 $ and Android Smartphone 100 $.

II. Art of State

The concept of eco-driving has become very important in recent years due to the increased number of vehicles on the roads, higher fuel costs and emissions control regulations to mitigate climate change.

The objective of the eco-driving is to reduce energy consumption by applying a set of rules based on physics that seek to reduce the demand for power. These rules do not require a technological support but the cooperation of the driver. The problem is that driving is a very complex task in which there are multiple objectives such as safety, speed, etc. Sometimes, the objectives could come in conflict [6]. For example, if the driver wants to arrive early at its destination, he will increase speed coming into conflict with the aim of

Manuscript submitted May 23, 2011. The research leading to these results has received funding by the ARTEMISA project TIN2009-14378-C02-02 within the Spanish "Plan Nacional de I+D+I", and the Madrid regional community projects S2009/TIC-1650 and CCG10-UC3M/TIC-4992.

V.Corcoba Magaña. Author is with Dpto. de Ingeniería Telemática. Universidad Carlos III de Madrid Leganés, Madrid, Spain (e-mail: vcorcoba@ it.uc3m.es).

M.Muñoz Organero. Author is with Dpto. de Ingeniería Telemática. Universidad Carlos III de Madrid Leganés, Madrid, Spain (e-mail: munozm@ it.uc3m.es).

reducing energy consumption. On the other hand, many drivers are unaware that by applying a set of rules on driving can reduce energy consumption, so that, the motivation and learning are fundamental in the eco-driving process. For the driver to learn the efficient driving techniques we can use an eco-driving assistant. There are several studies like Boriboonsomsin Kanok et al. [7] that value the suitability of eco-driving assistants to make the user acquire a more efficient driving style.

One of the main problems of eco-driving systems is to identify what factors influence energy consumption. Ericsson [8] suggests that, to save fuel, heavy acceleration and high speed driving should be avoided. Johansson et al. [9] suggested maintaining low deceleration levels, minimizing the use of 1st and 2nd gears, increasing the use of 5th gear, and block changing gears where possible. Kuhler et al. [10] identified a set of ten variables that influence energy consumption and the emission of polluting gases. The drawback of these proposals is that they do not take into account the environment where the vehicle circulates that often has a significant influence on energy consumption.

Otherwise, there are in the market several commercial solutions that attempt that the driver acquires habits of efficient driving. Nissan has designed a system called Eco-Pedal [11] that suited the acceleration depending on the circumstances. If the driver exerts on a gas pressure exceeding the recommended one, he is alerted by a warning light that was accelerating incorrectly.

Garmin has developed a program called EcoRoute [12] for their GPS devices to get the most economical route in terms of fuel economy. This software also has a scoring system that rewards good driving style from the point of view of energy saving.

In addition to these proposals, Ford, Audi, Fiat and Honda also have eco-driving assistants. These solutions use data from vehicle sensors to assess the driving style from the energy efficiency point of view and then deliver efficient eco-driving tips. The problem is that they are dependent on vehicle model and are usually offered as an extra.

III. ARTEMISA ARCHITECTURE

Artemisa´s eco-driving assistant consists on three main components:

- Data acquisition system module: It gets the value of all the variables that influence in energy consumption. The data are obtained through the smartphone and vehicle´s diagnostic port.
- Expert System Module: Responsible for evaluating the driving style using data obtained from the data acquisition system and the knowledge base, and as a result of the evaluation, provides eco-driving tips
- User Interface: Responsible to show eco-driving tips through mobile device screen and at the same time converting eco-driving tips with high trust factor to voice. It will also be responsible for rendering the graphical interface that the user employs to configure the assistant.

Then, we will describe each of the modules of Artemisa eco-driving assistant.

IV. DATA ACQUISITION SYSTEM

As previously mentioned, the Artemisa´s project has three main objectives:

- *Accurately modeling the driving style of the driver to give useful advice:* For this, the data acquisition system will get the value of all variables that influence the energy consumption associated with both the car and the environment where it circulates.
- *Economical system:* Eco-driving assistant as cheap as possible with the aim of which is widely used by the population. Other proposals are aimed at the high-end car market.
- *Standard:* Eco-driving assistant may be used in any vehicle. Many proposals are dependent on a particular model of vehicle.

To fulfill the three objectives we propose to use an Android Smartphone with a Bluetooth Adapter as data acquisition system. This solution is able to get lots of information affecting energy consumption without needing to install additional hardware to the vehicle.

In Figure 1, we can see the hardware elements of the data acquisition system. Following are described:

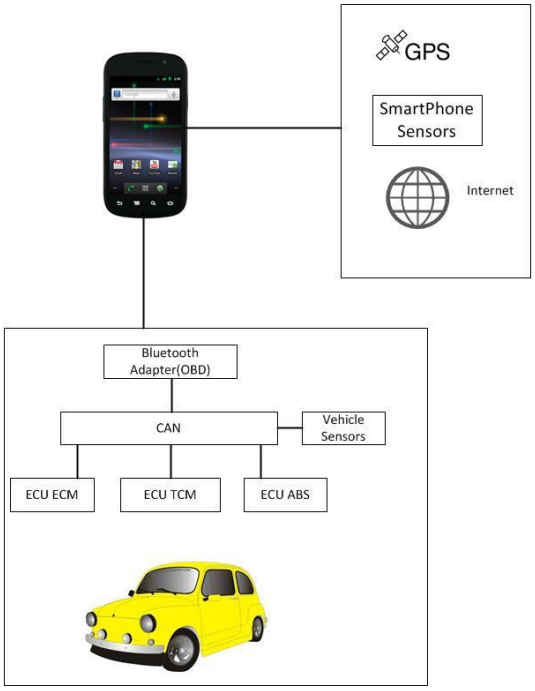

Fig. 1. Hardware elements of Data Acquisition System

A. *Hardware elements*

Diagnostic Port (OBD2): All modern vehicles have a port [17] through which data on vehicle emissions and diagnostic faults in the vehicle components can be obtained. The interface of the port is almost all OBD2. The OBD idea was proposed in 1984 and is known as the standard OBDI. OBDI is strong mind focused on the assessment of the emission of gaseous pollutants from the vehicle. In 1988, the standard OBD2 was proposed. OBD2 provides much more information than OBDI because its aim is not only to evaluate the emission of gas pollutants, but also to be able to do in-depth diagnostic about the operation of vehicle.

Bluetooth Adapter: Acts as interpreter converting diagnostic OBD2 port signals to serial data. Diagnostic port provides numerous data about engine control unit and other elements of the vehicle as the brakes, TCM, ABS, etc.

To access this information, we will use an identifier called PID (hexadecimal digits). There are standard basic lists of PIDs, and how to convert raw OBD-II diagnostic output obtained for each PID to meaningful units, but manufacturers are not required to implement all the PIDS and also include many non standard.

The process to obtain the values of the vehicle´s sensors is as follows:

- The mobile device sends a PID to Bluetooth adapter
- The Bluetooth adapter sends the PID to the vehicle's bus
- A device on the bus recognizes the PID and sends the value for that PID to bus
- The Bluetooth adapter reads the response, and sends it to mobile device

In Figure 2 we can see the process of communication between Smartphone and OBD port.

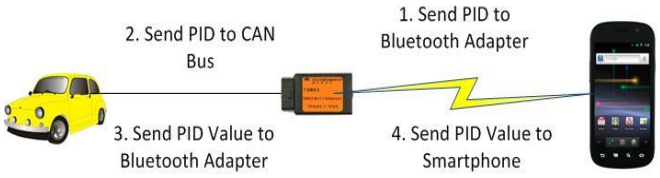

Fig. 2. Communication between Smartphone and OBD port.

The Artemisa´s eco-driving assistant uses OBDLink Bluetooth Adapter from ScanTool.Net [18] because contains the chip STN1110. This chip has better features than other competitors allowing us to sample more frequently.

Android device: Executes the Artemisa´s eco-driving assistant (data acquisition system and expert system). Mobile device must have minimum characteristics to execute the assistant but this is not a problem since most of the devices on the market have these requirements. The features are:

- GPS: It is used to determine the vehicle´s position. If GPS signal is not available, we will use the network location. Knowing the location of the vehicle is required to get the weather of the environment and state road.
- GPRS/3G/LTE Connectivity: Allows us to connect to the Internet to find out the state and weather conditions of the road.
- Bluetooth connectivity: Allows connecting mobile device Bluetooth/OBD2 adapter to get the data supplied by the diagnostic port. Android's APIs from version 1.5 allows use socket communications through the Bluetooth interface from the mobile device, so it must have this version or higher.
- Light, Orientation and Accelerometer sensors: Allow to know the value of variables that influence in the consumed energy; in particular the slope road, the ambient light and the vehicle acceleration.

B. *Software Modules*

The software architecture of the Artemisa´s data acquisition system consists of five modules: Sensors, Location, Weather, OBDII and Manager Database. Below, we will describe each of the modules.

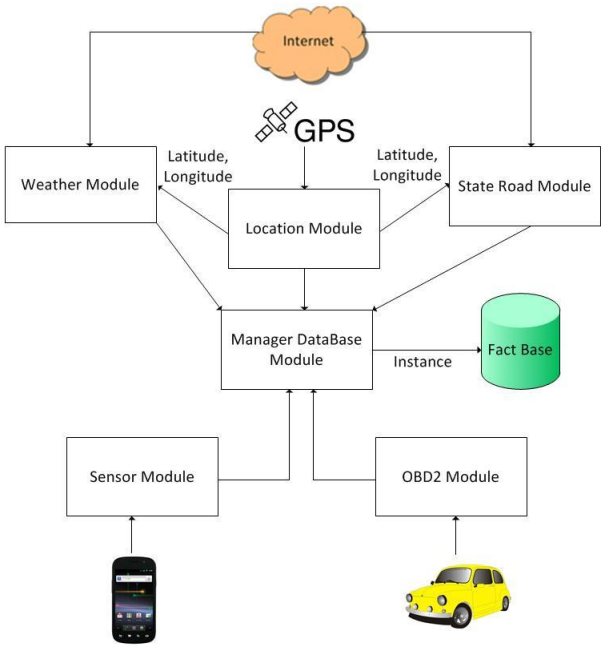

Fig. 3. Artemisa´s Architecture Software.

Sensor Module

This module is responsible for obtaining the data provided by the sensors of the Smartphone. The smartphones have a large number of sensors that allow us to obtain information about the environment. Some of this information can also be

obtained through the car sensors, but using the sensors of the smartphone in combination with the vehicle´s sensors is a good idea for the following reasons:

• It may happen that we can not access sensor data provided by the vehicle because they use non-standard PIDs. In this case we could use the information provided by the Smartphone.

• The vehicle sensors are subjected to extreme situations, so they can be easily damaged. If the difference between the data supplied by the vehicle sensor and the sensor is high, we will discard the data supplied by the vehicle sensor.

Android OS [24] allows us to access the sensor values of the mobile device through APIs. For this, the Android OS provides the classes:

• SensorManager: Allows access to the device's sensors.

• Sensor: Class representing a sensor.

• SensorEvent: This class represents a Sensor event and holds information such as the sensor's type, the time-stamp and the accuracy.

Android APIs support the following type of sensors: Accelerometer, Gravity, Gyroscope, Light, Orientation, Magnetic, Pressure, Proximity and Temperature.
Artemisa uses light, orientation and acceleration sensors. Thanks to these sensors we will get lighting conditions of the environment, the slope of the road and the acceleration of the vehicle.

Location Module

This module is responsible for obtaining the latitude, longitude, altitude and speed. Most mobile devices with Android have GPS, so we can get the geographical coordinates and vehicle´s speed. Furthermore, if the device does not have a GPS hardware or GPS signal we can get the location from mobile or Wi-Fi networks.

The Android´s APIs provide classes Location (representing a Geographic location at a particular time) and LocationManager (which provides access to the system location services)

Weather Module

This module is responsible for obtaining the temperature, speed and direction of the wind, and atmospheric pressure through the Smartphone´s Internet connection. We use a weather web service to obtain the weather information that returns a XML file. For parsing XML Android provides three methods:

- SAX: The parser reads the XML document sequentially and will generate different events with the information of each element read.
- DOM: The document is read completely before performing any action. The DOM parser turns everything into a structure of type tree, where the various elements of the XML are represented in the form of nodes and their parent-child hierarchy is defined by relations between these nodes.
- XMLPull: It is based on defining the actions carried out for each of the events generated during the sequential reading of the XML document, but unlike SAX, it explicitly realizes the reading of the next element and we can define the actions that we are going to be executed when you read an element.

In our data acquisition system, we have decided to use XMLPull because it allows more control over the reading of the XML file that the SAX method, and require less memory than DOM because XMLPull does not have to build a tree.

State road module

The module is designed to detect whether there are traffic incidents on the road. This will use the data supplied by the location module and the incident data web service from the General Direction of Traffic (DGT) in Spain. However, we could use a different Web Service that provides state road information due to the modular nature of the proposal and the facilities provided by the Android OS for the recovery and processing information obtained through the Internet.

DGT uses multiple cameras installed on Spanish roads to get the data about its state. Data provided from DGT web service includes coordinates, that will be used along with the coordinates supplied by GPS/Network from the device to see if there is any incident in the area where the vehicle travels.

OBDII Module

This module is responsible for obtaining data (speed, gear, distance traveled, etc.) provided by the vehicle's OBDII port.

For this, we use a plugin for Torque (an Android Application developed by IAN Hawkins). Toque [19] obtains the data provided by the OBD port using the Bluetooth adapter previously described. The main drawback of Torque is that only implements a basic list of PIDs. However, we can add custom PID for more information.

Manager Database Module

This module manages the database where to store the data from the remaining modules of the data acquisition system.

Android uses SqlLite3 as database management system. It also provides high-level functions to manage the database through the ContentProviders.

In the eco-driving assistant, the database will be accessed simultaneously by the data acquisition system and expert system. Concurrent access is managed transparently by the ContentProvider.

V. EXPERT SYSTEM

Analyzing the driving style from the point of view of energy consumption for then giving eco-driving advice is a very

complex problem due to the large number of variables involved. We propose a solution based on the use of an expert system that was executed on an Android device.

Current mobile devices have processors at 1 GHz or even there are already devices with dual-core processors (TEGRA, Samsung Exynos and OMAP4) on the market. Also tend to have 512 MB of RAM, so they are powerful enough to execute complex tasks.

Expert system must comply with three requirements: Fast execution, low memory usage and the ability to handle instances with missing attribute values.

Expert system should be run faster because eco-driving tips should be related to the actions taken by the driver in the last 10 minutes. Also keep in mind that even though the mobile device processors are very powerful can not be compared to a desktop computer.

Moreover, it is also important to note that expert systems require a great amount of memory and mobile devices do not usually have more than 1 GB.

We must also bear in mind that we will not always have the value of all the attributes of the expert system. Internet connection could fail, and we could not know data such as the state of the road or weather conditions. In addition, the data provided by the diagnostic port depends on the model of vehicle. However, the system must be able to continue to evaluate the driver and give advice with the information available. For this reason, expert system has to handle instances with missing attribute values.

Expert system used by the eco-driving assistant consists of the following elements:

- Facts Base: consists in a SQLite database that contains the data collected by the data acquisition system during the last ten minutes.
- Preprocessing module: Responsible for generating a single instance from stored instances from facts base.
- Knowledge Base: Contains the training set that the classifier used to determine the efficient driving tips. Training set has been obtained using several efficient driving manuals.
- Classifier: Responsible for inferring the most appropriate advice, taking into account the driver's driving style over the last 10 minutes. It uses the knowledge base and the instance generated by the preprocessing module.

A. *Facts Base*

It is a SQLite database that stores the data obtained by the data acquisition system. Its format is:

ValueAttribute_1, ValueAttribute_2, …, ValueAttribute_i

The attributes are the set of variables (speed, gear, weather conditions, ect…) whose values are obtained by the data acquisition system. When the value of the attribute is not known we insert a question mark instead.

Facts base only store data in the last ten minutes, so we have to progressively remove the oldest entries.

B. *Preprocessing Module*

This module is executed every 10 minutes. It is responsible for generating a single instance from the data stored on the base of facts. To obtain the instance we will calculate the arithmetic mean for each attribute of the facts base, if the attribute is a numeric type. In case that the attribute is nominal, we will choose the most frequent value.

C. *Knowledge Base*

It is a SQLite database that contains the knowledge (also called training set) extracted from the manuals about efficient driving [3] [4]. In addition, the knowledge base will increase with each new classified instance, in order to improve the response of the system in each new interaction.

However, the Expert System runs on a mobile platform with limited memory resources, so that Knowledge Base can not grow without control. Otherwise, if the knowledge base is too big, the construction of the classifier will consume too much time and the eco-driving tips may be issued too late.

To resolve this problem, we include new instances in the knowledge base while the build time of the classification model does not exceed a threshold. In our case the limit is nine minutes.

In future work, we will explore a smart way to replace the instances introduced during the classification process for new ones, once the threshold has been exceeded. The aim is that the knowledge base contains instances that produce the best results.

The format of the instances stored in the knowledge base is as follows:

ValueAttribute_1,ValueAttribute_2 ,....., Class

Attributes are the set of variables (speed, gear, weather conditions, etc….) whose values are obtained by the data acquisition system. When the value of the attribute is not known, we insert a question mark instead. Class represents an eco-driving advice. Below, we will show an example of the instance of the knowledge base.

?,?,?,?,?,?,?,?,?,?,?,?,?,150,?,?,?,?,?,1

This example expresses that if a vehicle circulates at 150 km/h must be displayed the driving tip one. The advice one indicates that the user have to reduce speed because circulate at high speeds increases fuel consumption since is needed more power engine.

D. *Classifier*

Classifier algorithms used by the expert system should consume few resources and at the same time to produce good results. The reason is that although today's mobile devices are

powerful, its memory resources and battery remain limited. Classification algorithms make intensive use of CPU with the consequent energy expenditure, moreover require lots of memory. We have to notice that currently Smartphone with more memory have only 1 GB.

To check the feasibility of implementing an expert system on Android device and select a good initial classification algorithm, we have analyzed three classifiers (Random Forest, J48 and NaïveBayes) often used in the area of machine learning.

In the future, when we have a lot of real data, we will examine whether there another classifier algorithm with better results.

Random Forest

It is an ensemble classifier algorithm [20] that combines Random Decision Trees with Bagging to achieve very high classification accuracy. It was proposed by Leo Breiman in 1999. The main features of this algorithm are:

- Easy to implement
- It has good generalization properties
- Algorithm outputs more information than just class label
- It runs efficiently on large data bases.
- It can handle thousands of input variables without variable deletion.
- It gives estimates of what variables are important in the classification.

J48

J48 is an open source Java implementation [21] of the C4.5 algorithm. C4.5 was proposed by Ross Quinlan and derives from the ID3 algorithm. It builds decision trees from a set of training data using the concept of information gain. Currently, it is one of the most commonly used in the machine learning area. The main features of this algorithm are:

- Permits both continuous and discrete attributes
- Is robust in the presence of noise
- Handling training data with missing attribute values
- Pruning trees after creation in order to save resources
- Provides good results with most of the datasets

Naïve Bayes

A Naive Bayes classifier [22] is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong independence assumption. It assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. For example, P (Pass | Study, Clever) = P (Pass| Study).

Assumption may not hold true in some occasions but has shown good results. This algorithmic is called Naïve Bayes due to this strong assumption.

Election of classification algorithm

The Artemisa´s eco-driving assistant uses Random Forest classification algorithm. The choice of this algorithm was made after a comparative study between the algorithms J48, Naive Bayes and Random Forest.

The algorithms were executed in a Nexus S with ArmV7 processor at 1 GHz, 512 MB of RAM and Android 2.3.1. For this study were used several datasets extracted from the UCI website and the Weka library adapted to Android.

It is important to note that the implementation of Random Forest in Weka does not include many powerful features of the algorithm as variable importance, interactions, scaling, etc. It only allows select the number of trees and controlling the number of random attributes to be chosen for each node. Other implementations of the algorithm could produce better results.

For this study, we used several datasets extracted from the UCI website [23] and the Weka Machine Learning software [24] using 10 fold cross validation. **A** Algorithms were executed with Weka´s default parameters. In table 1, we can see the characteristics of the datasets.

	Attributes	**Instances**
Sonar	61	208
Segment	20	1500
SoyBean	36	683
Spambase	58	4601

Table 1. Characteristics of the datasets.

	Time (seconds)	**Correctly Classified Instances (%)**
Random Forest	8,22	83,1731
J48	7,156	71,1538
Naïve Bayes	3,457	66,8269

Table 2. Result of the execution of the algorithms with Sonar dataset.

	Time (seconds)	**Correctly Classified Instances (%)**
Random Forest	41,519	97,2
J48	20,608	81,0667

Naïve Bayes	14,070	95,2

Table 3. Result of the execution of the algorithms with Segment dataset.

	Time (seconds)	**Correctly Classified Instances (%)**
Random Forest	24,3531	92,9722
J48	78,84	90,776
Naïve Bayes	3,672	92,6794

Table 4. Result of the execution of the algorithms with SoyBean dataset.

	Time (seconds)	**Correctly Classified Instances (%)**
Random Forest	257	94,9359
J48	299	92,6103
Naïve Bayes	48,4111	79,6131

Table 5. Result of the execution of the algorithms with SpamBase dataset.

Execution results are shown in tables from 2 to 5. We have considered the time it takes to build the classifier model and classify the instance because the knowledge base will increase with each new classified instance, in order to improve the response of the system in each new interaction.

It can be seen that J48 presents good precision classifying with most of the datasets. On the other hand, their execution time is higher than Naïve Bayes, especially when the number of attributes or the training set is large.

Naive Bayes, despite the strong independence assumption, obtained reasonably good results and it is the fastest of the three classifications algorithmic.

Random Forest execution time is similar to J48. However, it performs better than J48 and Naïve Bayes in terms of prediction accuracy. For this reason, we have chosen Random Forest as the expert system classifier algorithm.

VI. USER INTERFACE

Distraction is one of the major causes of traffic accidents. The use of non-driving devices such as mobile phone, GPS or eco-driving assistant has many negative effects if handled while driving or we divert too much attention to them. Some of these adverse effects are:

- Interference in the management of the vehicle controls
- Increased reaction time
- Loss of perception of traffic signs
- Alteration of speed and safety distance
- Realization of disallowed maneuvers

To avoid these adverse effects, Graphical interface module will display with a clear typographic the eco-driving tips. In addition, we will convert the advice with more probability to voice using TTS API provided by Android since version 1.6.

Furthermore, we need have a TTS Engine installed on the smartphone. Android 2.1 brings by default Pico TTS engine but by storage problems could not be installed. In any case, we can download it from the market.

VII. FUNCTIONAL DESCRIPTION OF THE ARTEMISA´S ECO-DRIVING ASSISTANT

In the figure 3, we can see a functional description of the Artemisa´s eco-driving assistant. Below, we will describe the process that is performed to obtain the advice of efficient driving.

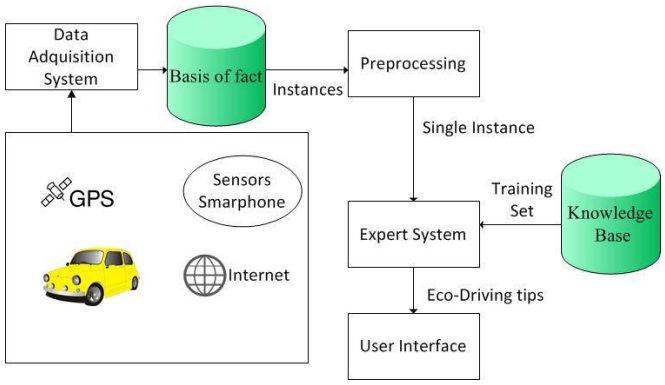

Fig. 3. Functional description of the Artemisa´s eco-driving assistant.

The first step to be performed by the eco-driving assistant is to get enough information to model the driver's driving style from the point of view of energy consumption. The information is gathered by the module data acquisition system through the mobile device, Internet and vehicle´s diagnostic port. The data collected are stored in the database of facts. Samples are taken every second.

In parallel, every 10 minutes is run the expert system module. The first component to run expert system is the preprocessing module. This module generates a single instance from the data collected in the last 10 minutes that are stored in the basis of facts. The resulting instance will be classified by the expert system obtaining eco-driving tips. Each eco-driving tip has a probability that is assigned by the classifier. Higher probability indicates more possibilities that the driver is not applying the eco-driving advice. Expert system module does

not run continuously because occasional actions that negatively influencing energy consumption must not be penalized. For example, if a man crossing the road incorrectly we have to stop suddenly even if it is a mistake from the point of view of efficient driving.

Finally, user interface module will convert to voice the eco-driving tip with more probability and will show all tips on screen. All the eco-driving advices are not converted to voice because we want prevent that the driver swerve attention from the road.

VIII. CONCLUSION

In this paper, the Artemisa´s eco driving assistant has been presented. Artemisa accurately assesses the driver's driving style from the standpoint of power consumption and based on the result of assessment issues tips to improve his driving style.

Unlike other solutions Artemisa takes into account environmental variables (weather conditions, road state, etc.) that affect energy consumption. The advantage is that the evaluation of the driving style is much more accurate. Moreover, the solution is inexpensive and can be installed in any vehicle.

Future work contemplates improving the system analyzing the actions taken by the driver in certain situations such as stopping at a crosswalk or traffic signal, take a curve, incorporation to the road, etc. To detect the situation, Smartphone camera can be very helpful.

ACKNOWLEDGMENT

The research leading to these results has received funding by the ARTEMISA project TIN2009-14378-C02-02 within the Spanish "Plan Nacional de I+D+I", and the Madrid regional community projects S2009/TIC-1650 and CCG10-UC3M/TIC-4992.

REFERENCES

[1] Spanish Society of Pneumology and Thoracic Surgery. URL: http://www.separ.es. January 2011.

[2] "Traffic: civilization or barbarism". The risk Observatory. Institute for security studies (IDES). URL: http://www.seguretat.org. 2006.

[3] IDEA (Institute for Energy Diversification and Saving of Energy). Manual of eco-driving for industrials vehicles. November 2005.

[4] IDEA (Institute for Energy Diversification and Saving of Energy). Manual of eco-driving for cars. January 2007.

[5] Jack N. Barkenbus, Eco-driving: An overlooked climate change initiative, Energy Policy, Volume 38, Issue 2, February 2010, Pages 762-769, ISSN 0301-4215, DOI: 10.1016/j.enpol.2009.10.021. (http://www.sciencedirect.com/science/article/B6V2W-4XJN4X4-5/2/6ffebae2ed6ed89c7f0c083fa7b37e6d).

[6] Mark S. Young, Stewart A. Birrell, Neville A. Stanton, Safe driving in a green world: A review of driver performance benchmarks and technologies to support `smart' driving, Applied Ergonomics, Volume 42, Issue 4, Applied Ergonomics and Transportation Safety, May 2011, Pages 533-539, ISSN 0003-6870, DOI: 10.1016/j.apergo.2010.08.012.

[7] Kanok Boriboonsomsin, Alexander Vu, and Matthew Bart. "Co Eco-Driving: Pilot Evaluation of Driving Behavior Changes among U.S. Drivers". University of California Transportation Center. 2009.

[8] E.Erikcsson. "Independent driving pattern factors and their influence on fuel-use and exhaust emission factors" Transportation Research Part D: Transport, 2001 – Elsevier, 325-345J.

[9] Johansson, H., Gustafsson, P., Henke, M., Rosengren, M., 2003. Impact of EcoDriving on emissions. International Scientific Symposium on Transport and Air Pollution, Avignon, France.

[10] Kuhler, M., Kartens, D., "Improved driving cycle for testing automotive exhaust emissions". SAE Technical Paper Series 780650. 1978.

[11] EcoPedal.Nissan.URL:http://www.nissanglobal.com/EN/NEWS/2008/_STORY/080804-02-e.html. May 2011.

[12] EcoRoute.Garmin.URL:http://www.garmin.com/garmin/cms/us/services/ecoRoute. May 2011.

[13] FordmyTouch.URL:http://www.ford.com/technology/sync/myfordtouch/ May 2011.

[14] Audi EcoTraining. URL: http://uk.cars.yahoo.com/18072008/36/audi-plans-new-green-technology-0.html. May 2011.

[15] Fiat Eco-Drive. URL: http://www.fiat.com/ecodrive/. May 2011.

[16] HondaEcoAssist.URL:http://www.greencarcongress.com/2010/02/honda-study-finds-insights-eco-assist-system-results-in-average-10-improvement-in-fuel-economy-after.html. May 2011.

[17] Godavarty, S.; Broyles, S.; Parten, M.; , "Interfacing to the on-board diagnostic system," Vehicular Technology Conference, 2000. IEEE VTS-Fall VTC 2000. 52nd, vol.4, no., pp.2000-2004 vol.4, 2000 doi: 10.1109/VETECF.2000.886162 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=886162&isnumber=19144.

[18] OBDLink. URL: http://www.scantool.net. May 2011.

[19] Torque. URL: http://torque-bhp.com/software/torque-android-obd2-adapters/. May 2011

[20] Breiman, Leo. "Random Forests". Machine Learning. 2001-10-01. Springer Netherlands. Issn: 0885-6125. Doi: 10.1023/A:1010933404324.

[21] Weka Official Website. URL: http://www.cs.waikato.ac.nz/~ml/weka/. May 2011.

[22] I. Rish. "An empirical study of the naive Bayes classifier". IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence.

[23] UCI. Url: http://archive.ics.uci.edu/ml/. May 2011.

[24] Weka for Android OS. Url: https://github.com/rjmarsan/Weka-for-Android. May 2011.

[25] Android Os. Url: http://developer.android.com/index.html. May 2011.