A Mobile Knowledge-based System for On-Board Diagnostics and Car Driving Assistance

Michele Ruta, Floriano Scioscia, Filippo Gramegna, Eugenio Di Sciascio
Politecnico di Bari
via Re David 200
I-70125 Bari,Italy
m.ruta@poliba.it, f.scioscia@poliba.it, gramegna@deemail.poliba.it, disciascio@poliba.it

***Abstract*—*In-vehicle* electronic equipment aims to increase safety, by detecting risk factors and taking/suggesting corrective actions. This paper presents a knowledge-based framework for assisting a driver via her PDA. Car data extracted under On Board Diagnostics (OBD-II) protocol, data acquired from PDA embedded micro-devices and information retrieved from the Web are properly combined: a simple data fusion algorithm has been devised to collect and semantically annotate relevant safety events. Finally, a logic-based matchmaking allows to infer potential risk factors, enabling the system to issue accurate and timely warnings. The proposed approach has been implemented in a prototypical application for the Apple iPhone platform, in order to provide experimental evaluation in real-world test drives for corroborating the approach.**

***Keywords*-Semantic Web; On Board Diagnostics; Ubiquitous Computing; Data Fusion; Intelligent Transportation Systems**

I. INTRODUCTION

The social and economic costs of road accidents are widely acknowledged. Three main factors able to influence their incidence have been identified: to educate drivers to a more careful behavior; to improve road conditions; to enhance features and capabilities of protection devices on vehicles. Evidence shows that investing resources in any of these fields can lead to a decrease in the frequency and severity of car crashes [1].

Modern vehicles are equipped with several Electronic Control Units (ECUs) coordinating and monitoring internal components and devices, communicating over one or more car network buses, such as for example *CAN-Bus* [2]. International standards require new vehicles support the *On Board Diagnostics, version 2* (OBD-II) protocol (http://www.arb.ca.gov/msprog/obdprog/obdprog.htm - last accessed on July 19th, 2010) and be equipped with an OBD-compliant interface to provide direct access to data in the vehicle network. The OBD-II port can be found under the dashboard in the majority of current automobiles. It provides real-time access to a large number of vehicle status parameters. Furthermore, in case of malfunctions, Diagnostic Trouble Code (DTC) values are stored in the car ECU and can be later retrieved by maintenance technicians using proper hardware and software kits. In latest years, access has been granted also to the general public of car enthusiasts by developing OBD-II *Scan Tools*, *i.e.*, cheap electronic devices that bridge the OBD-II port with standard wired (RS-232, USB) or wireless (Bluetooth, IEEE 802.11) computer communication interfaces.

This paper presents a knowledge-based framework for assisting drivers, able to monitor vehicle data extracted via OBD-II and integrating environmental information gathered from external sources in order to detect potential risk factors and to provide warnings and suggestions in real-time. The mobile system we propose allows to process:

- vehicle status data collected from an OBD-II Scan Tool;
- data acquired from embedded smartphone micro-devices, such as GPS (Global Positioning System) and accelerometer;
- optionally, information retrieved from external Web-based data sources, *e.g.*, weather conditions.

Data are collected within short observation intervals and, by means of proper processing and fusion algorithms, the system is able to identify specific high-level events and conditions, based on low-level data streams. Detectable conditions include: vehicle health and safety equipments status; environmental factors (road surface, traffic); driving style. Furthermore, exploiting common Semantic Web techniques and technologies, got events are semantically annotated w.r.t. an ontology modeling factors influencing driving safety. Annotated descriptions undergo a matchmaking process – exploiting non-standard reasoning services [3]– which is able to discover all possible risks referred to current state of the "driver+vehicle+environment" system. The matchmaking outcome is used to suggest the driver actions and behaviors she can adopt in order to minimize perils.

The proposed framework has been implemented in a prototypical mobile software system, using the Apple iPhone smartphone (iPhone Specifications, http://www.apple.com/iphone/specs.html - last accessed on July 19th, 2010) as reference platform. The experimental evaluation has been carried out taking into account several real-world test drives under different conditions. Obtained results prove both feasibility and usefulness of the presented approach.

In the remaining of the paper, after a survey on most relevant related work in Section II, the proposed framework is described in detail in Section III. Experiments corroborating the approach are presented in Section IV, and finally, conclusion and future work close the paper.

II. Related Work

The basic design scheme for systems using OBD for automobile fault diagnostics is reported in [4]. It consists of three main elements: (i) on-board sensors and fault indicators, built in the vehicle and communicating with the ECU through a bus; (ii) VCI (Vehicle Communication Interface) that bridges the ECU and the computer diagnosis system through a wired or wireless interface leveraging either OBD or CAN-Bus protocols; (iii) diagnostic software, which provides both user interface and connection capabilities toward a remote maintenance center.

Available literature about OBD-based systems for real-time vehicle monitoring and signaling refers to *remote* and *on-board* solutions, respectively. The former follow the basic architectural model introduced in [5], where a system for both on-line vehicle diagnosis and real-time early warning is presented. It acquires GPS coordinates and vehicle OBD DTCs sending them to a Maintenance Center server via GPRS for immediate actions. All the collected data are stored into a database which is scanned by a diagnostics expert system that classifies vehicle status into either *critical* or *non-critical* and generates a rough suggestion advising the maintenance engineer for taking next action. A similar approach can be found in [6].

Our proposal differs from such works because it does not require expert technicians to understand system outputs. Furthermore, in our solution all processing happens in a smartphone application and then it better reflects an on-board framework.

Consider that, though useful for managing vehicle fleets, remote monitoring do not allow a direct driver assistance. To this aim, on-board monitoring prototypes and reporting systems have been developed [7]. They allow the car driver to be informed about relevant vehicle status conditions during trip. Such systems use custom circuitry for the OBD-II-to-computer interface and include several independent devices, communicating through both wired and wireless technologies. Nowadays freeware and commercial software packages are available, allowing to monitor OBD-II vehicle data by using just a smartphone and off-the-shelf Scan Tools. Nevertheless, all existing on-board monitoring systems directly display the acquired low-level data, and they do not provide more user-friendly information. Particularly, no solutions exploiting logic-based techniques for on-line monitoring of driving risks able to meaningfully assist drivers have been yet presented, to the best of our knowledge.

More recently, researchers acknowledged the possibility to exploit the wealth of real-time vehicle data available trough OBD in order to analyze driver behavior [8]. Current efforts aim at using multi-source information fusion to better understand the relationships between driving habits and vehicle performance, as well as to detect risk situations. Nevertheless, in current approaches, the analysis is performed off-line after data gathering, so they are not able to provide driver support in real time.

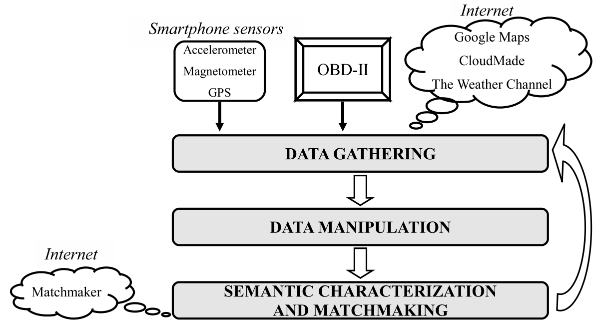


Figure 1. Workflow of the proposed framework

III. Framework

The framework we present includes both architecture and algorithms of a knowledge-based system leveraging the OBD-II car diagnosis and Apple iPhone to monitor environmental conditions, vehicle features, enabled protection equipments and driving style. Through a semantic-based matchmaking the system will be able to evaluate the driving risk level and to suggest how to reduce or even eliminate danger.

A *Kiwi Wifi* PLX wireless adapter (PLX Devices, Kiwi Wifi, http://www.plxkiwi.com/kiwiwifi/hardware.html - last accessed on July 19th, 2010) is exploited for interacting with OBD-II. When turned-on, it builds an ad-hoc network exposing a static IP address allowing an application to communicate with the OBD interface via socket in read/write mode.

As sketched in Figure 1, the proposed approach works along three subsequent stages: (i) data gathering; (ii) data manipulation; (iii) semantic characterization and matchmaking. They are repeatedly executed, within a fixed observation interval (in our case study, a period of 60 seconds was selected). At the end of each data gathering cycle, the data manipulation processing and semantic matchmaking steps are executed and outcomes are displayed to the user on the iPhone screen. In what follows framework details are reported.

A. Data Gathering

At this stage low level data are collected, such as kinetic and vehicle parameters useful to determine driving style, status of safety car equipments, weather conditions, road and traffic information.

The OBD-II interface is used to get data about vehicle performance. OBD-II specifications only comprise the *Physical*

HEADERS			DATA							CRC
H1 TYPE	H2 TARGET	H3 SOURCE	D1 MODE	D2 PID	D3	D4	D5	D6	D7	CHECKSUM

Figure 2. OBD frame common structure

Signal Layer (PSL) and *OBD-II Data Communication Layer* (DCL) w.r.t. the ISO/OSI model. Particularly, PSL outlines hardware characteristics, standard connector (SAE J1962 [9]) conformation and exploited protocols. DCL defines the structure of diagnosis messages exchanged with the ECU, as described in SAE J1979 standard [10]. Request and reply messages have the same conformation, reported in Figure 2. Header bytes H1, H2 and H3 denote priority or message type, destination and sender addresses, respectively. The first data byte D1, namely *mode byte*, indicates the modality to access OBD information. The standard supports 10 modes for diagnostic requests. In particular, *mode 1* is used to obtain current diagnosis data, and it is arguably the most useful mode for our purposes. The second data byte comprises the so-called PID (Parameter IDentification): a value indicating what data is required. The PID also fills the second byte in the corresponding reply packet coming from the vehicle. The remaining data bytes, when used, are reserved for further specification about required data; in a reply message, they are the actual data returned from the vehicle ECU. The last byte is exploited for message error control.

Though the proposed system is able to retrieve all possible vehicle parameters via the OBD-II interface, our case study focuses on vehicle speed (PID $\texttt{0D}_\texttt{h}$) and RPM (PID $\texttt{0C}_\texttt{h}$), which contribute to characterize driving style as well as road traffic. The Apple iPhone (like many currently available high-end smartphones) integrates several micro-devices and offers wireless Internet connectivity through the cellular network. Such capabilities are exploited to collect information about the environment. The GPS receiver provides latitude and longitude coordinates of vehicle current position, which are used for a *reverse geocoding* query using the Google Maps API (Google Maps API Family, http://code.google.com/intl/it-IT/apis/maps/ - last accessed on July 19th, 2010) to get the corresponding location address. Location is further exploited to get weather conditions, using a free service offered by TWC (The Weather Channel: weather XML Data Feed, http://www.weather.com/services/xmloap.html - last accessed on July 19th, 2010) website. For our purposes data concerning weather description (rain, snow, fog, cloudy) and wind speed are exploited. Finally, in order to access roads information, an additional reverse-geocoding operation is implemented using the CloudMade service (http://cloudmade.com/ - last accessed on July 19th, 2010). The reply message, in JSON (JavaScript Object Notation data interchange format, http://www.json.org/ - last accessed on July 19th, 2010) format, contains an indication of road type and speed limit. For what concerns the safety equipments available on the vehicle, the user can enable/disable/check their status exploiting the *Settings* view of the implemented application leveraging the interaction with the car via OBD.

B. Data Manipulation

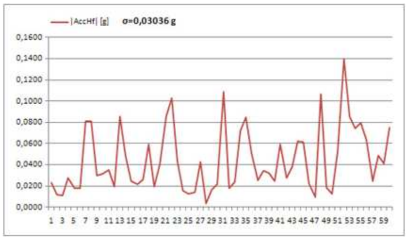
The final aim of this application phase is to process previously collected data in order to identify conditions and events, which can be annotated w.r.t. a reference ontology (described in Section III-C). Several statistical procedures and data fusion algorithms were devised and tested in order to build a set of (binary or multiple) classifiers for mapping data series to events. Solutions outlined hereafter were selected because they provide adequate sensitivity also maintaining moderate computational and memory requirements.

Road conditions. It is possible to distinguish between even and uneven road surface by computing the standard deviation of acceleration values produced by vehicle oscillations. In the previous data gathering step, the three-dimensional acceleration vector is sampled from the iPhone accelerometer at a 2 Hz frequency. Then high-pass filtering is applied, in order to discard components due to gravity and normal vehicle acceleration/deceleration: 15 Hz was found as the optimal cutoff frequency. As shown in Figure 3(a) and 3(d), acceleration values variability is significantly higher on an uneven road surface w.r.t. to an even one. Experimental tests proved an optimal threshold value of $\sigma = 0.020g$ ($g = 9.80665\, m/s^2$) for the standard deviation of acceleration to classify road surface.

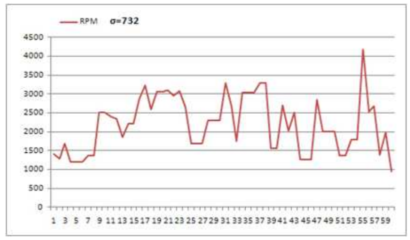
Driving style. In order to distinguish between an *imprudent* and a *regular* drive, abrupt speed and direction changes should be detected. The standard deviation of RPM (Revolutions Per Minute) of the vehicle engine –retrieved from OBD-II interface at 1 Hz frequency in the observation period– was selected as discriminatory parameter. An imprudent driving style can be distinguished from an even pace by observing the variability of RPM, as depicted in Figure 3(b) and 3(e). Our experiments proved a threshold value of $\sigma = 400$ RPM provide good reliability in distinguishing the two driving styles.

Speed. To characterize vehicle speed, it is sufficient to compute the average speed value for the observation interval. W.r.t. Italian urban speed limits, the threshold value to distinguish a high-speed driving from a low-speed one was set to 40 km/h.

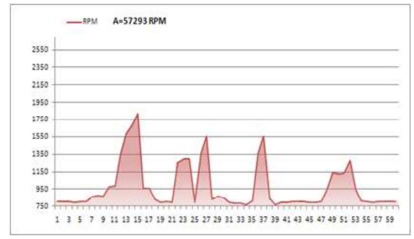
Traffic conditions. OBD parameters are also useful to characterize traffic conditions. In congested traffic situations, a driver usually alternates frequent and fast speedups/stops and downtimes. In terms of RPM, this behavior produces a typical sawtooth waveform –with sharper upward and downward slopes– that alternate with stages at a minimum



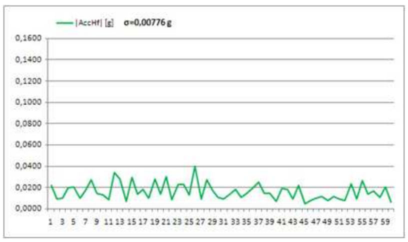
(a) Acceleration values for an uneven road



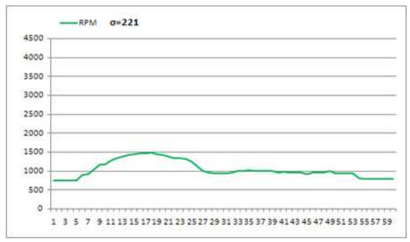
(b) RPM values for an aggressive driving style



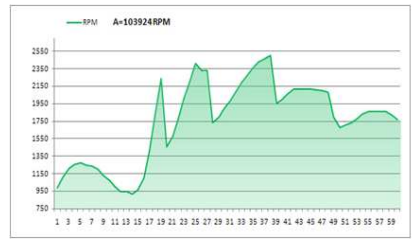
(c) RPM integral area for high density traffic



(d) Acceleration values for an even road



(e) RPM values for an even pace driving style



(f) RPM integral area for low density traffic

Figure 3. Acceleration and RPM values acquired in data gathering phase

value. As depicted in Figure 3(c) and 3(f), the integral area computed in case of traffic congestion is clearly lower than in case of lack of traffic. A threshold value of 65000 (with a data gathering phase of 60 seconds) allows to discern between these two traffic conditions.

Wind. Wind speed is a relevant driving risk factor when it exceeds a specific value. In accordance with commonly exploited *Beaufort scale*, in our framework the threshold value is set to 40 km/h.

C. Semantic Characterization and Matchmaking

The semantic annotation of environmental and driving events closes the context extraction and it prepares the subsequent matchmaking phase. A prototypical ontology modeling the domain of interest has been implemented, using OWL-DL [11] formal language, grounded on Description Logics (DL) semantics. It specifies classes and properties (a.k.a. concepts and roles, respectively) needed to characterize all the events and situations that can be detected by the data gathering and manipulation steps. As the framework will be augmented with new data sources and algorithms to detect more situations, it will be possible to extend the domain ontology accordingly. Consistency checks are performed at each stage of ontology evolution, in order to ensure that new knowledge does not conflict with previously modeled one. In greater detail, the following classes and properties have been defined.

- *Weather* describes weather conditions. It has five subclasses: *Fog*, *Snow*, *Cloudy*, *Rain*, *Clear*.
- *Wind* refers to wind strength. The corresponding subclasses are *Weak_Wind* and *Strong_Wind*.
- *Road_Surface* represents road conditions. Two different kinds were modeled, through subclasses *Uneven_Road* and *Even_Road*.
- *Road_Condition* models the kind of road. The corresponding subclasses are *High_Speed_Road* and *Low_Speed_Road*.
- *Traffic* is related to traffic conditions. The subclasses are *High_Density_Traffic* and *Low_Density_Traffic*.
- *Driving_Style* refers to user driving style, with subclasses *Even_Pace_Style* and *Imprudent_Style*.
- *Vehicle_Speed* describes the vehicle speed, whose subclasses are *High_Speed* and *Low_Speed*.
- *Safety_Equipment* represents protection devices that may be available on a vehicle. In particular, the following subclasses were modeled: *Fog_Lamp*, *ABS*, *ESP* and *Snow_Chains*.
- *Vehicle* is related to the car. It is involved in the following property relations: *hasDriving_Style* with *Driving_Style* class, *hasSpeed* with *Vehicle_Speed* class and *hasSafety_Equipment* with class *Safety_Equipment*.

The first five classes model the environment, whereas the remaining ones are used to describe both vehicle and user driving style. It is important to note that the ontology is not just a taxonomy, since definition and inclusion axioms are used. Particularly, the semantic description of environmental conditions focuses on potential risks they can cause to driver, on the required vehicle equipment and driving style able to minimize the risk. For example, let us consider the following semantic annotation in DL formalism: $Fog \equiv Weather \sqcap \forall\ hasSafety\_Equipment.(Fog\_Lamp \sqcap ABS) \sqcap \forall\ hasSpeed.Low\_Speed \sqcap \forall\ hasDriving\_Style.Even\_Pace\_Style.$ It means that, in order to avoid risks produced by *Fog*, the vehicle must be equipped with fog lamp and ABS and the user must adopt an even pace driving style with low speed.

Semantic matchmaking process exploits `MaMaS`-TNG (MatchMaking Service-The Next Generation, available as

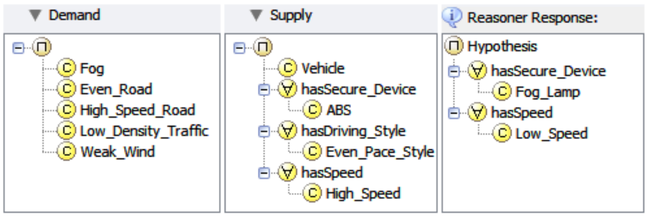


Figure 4. Abduction query example

an HTTP service at: http://dee227.poliba.it:8080/MAMAS-tng/DIG) matchmaker to infer possible risks for the user, warning her to avoid danger, given the context. The *Concept Abduction* non-standard inference service was selected to perform this task. Due to space limitations, the reader is referred to [3] for a thorough presentation of the Concept Abduction Problem (CAP). In a nutshell, given a request R and an available resource/service S, described w.r.t. a common ontology, Concept Abduction can be used to identify what is missing in S in order to completely satisfy R. In our framework, the context semantic annotation represents the request (*i.e.*, what requirements are needed to travel safely), while the semantic description of vehicle and user driving style model resources availability (*i.e.*, what is provided by the "vehicle+driver" system). In this way, the abduction process will infer the safety requirements that are not explicitly satisfied by current vehicle configuration and driver behavior, thus providing proper suggestions to the driver so that every kind of danger is prevented.

Figure 4 shows an example of abduction query and results with reference to the previous very small example. As the reasoner reply underlines, the presence of *Fog* concept in the request implies the need of both fog lamps and ABS, as well as an even pace driving style with low speed. On the other hand, as the vehicle only offers ABS and the driver has an even pace driving style with high speed, the abduction outcome suggests the driver to activate fog lamps and adopt low speed to attenuate risks.

IV. Tests and Experimental Results

Effectiveness of the devised framework and usability of the mobile application were tested in three different environmental conditions described in Table I, using a Mercedes C220 CDI registered in 2003. For each scenario, two test drives were performed with different speed, driving style and safety equipments settings, as described in Table II. Video recordings of tests are available at http://sisinflab.poliba.it/idrivesafe/. It is possible to see that, under good cellular network coverage, system performance is adequate to grant a satisfactory user experience.

The first example scenario is featured by an uneven and low speed road. In Figure 5 a screenshot of system outcome is shown. Part (a) shows information about weather, traffic and road description. Information about vehicle speed and driving style are reported in part (b). Finally, part (c) will

Table I
Context description

	Test 1	Test 2	Test 3
Location	Grumo Appula(BA)	Bari	Toritto(BA)
Road	SP. 71	C.so V. Emanuele II	SP. 1
Weather	Cloudy	Clear	Rain
Wind	Weak	Weak	Strong
Road type	Low Speed	Low Speed	Low Speed
Road surface	Uneven	Even	Uneven
Traffic	Low density	High Density	Low Density

Table II
Test settings

	Speed	Driving Style	Safety Equipments
Test 1, Setting 1	Low Speed	Even Pace	ABS, ESP, Fog Lamp
Test 1, Setting 2	High Speed	Imprudent	None
Test 2, Setting 1	Low Speed	Even Pace	ABS, ESP, Fog Lamp
Test 2, Setting 2	Low Speed	Imprudent	None
Test 3, Setting 1	Low Speed	Even Pace	ABS, ESP, Fog Lamp
Test 3, Setting 2	High Speed	Even Pace	None

contain system suggestions to the user. In the first case, the system detects a risk-free situation. Although the road is uneven and imposes a low speed, the driver is adopting a driving style suitable for these conditions and the car is equipped with the needed safety equipments. In the second case, the system detects a dangerous situation, due to an imprudent driving style, a high speed (inappropriate for given road conditions) and the lack of ABS and ESP (strictly needed on an uneven road). To reduce such risk factors the system suggests the driver to moderate her driving style, to reduce speed and to activate the required safety devices.

The second test was performed in high-density traffic conditions on a low-speed road. As depicted in Figure 6, the first configuration is not dangerous for the user. Although the traffic is intense, the driver adopts an even pace and the car is featured by the needed safety equipment. In the second case, instead, the system detects risk factors, due to an imprudent driving style (absolutely not suitable in high-density traffic) and lack of ABS. Hence, the system suggests to drive with caution and to activate ABS (if possible).

The last test refers to adverse climatic conditions with rain and high wind. Figure 7 shows system outcomes. In the first case no risk for the user is detected. Notwithstanding adverse weather conditions, the driving style is proper and

Figure 5. System outcome in Test 1

Figure 6. System outcome in Test 2

Figure 7. System outcome in Test 3

the car is safe (ABS and ESP are turned on). In the second configuration, on the contrary, the system reveals a danger, deriving from high speed (very risky in case of rain, high wind and narrow roads) and from the lack of ABS and ESP protection. So the system suggests the user to activate ABS, ESP and to reduce the speed.

V. Conclusion and Future Work

We have presented a knowledge-based framework and a prototypical system for real-time driving assistance. They refer to every OBD-based vehicle and comply with several driving context without the need of learning stages. By means of information extracted trough the accelerometer and GPS embedded in an Apple iPhone PDA and exploiting Web-based available services, a context annotation is performed. It enables semantic-based inferences which finally provide useful recommendations for driving safely. Experimental evaluations evidenced that the system is able to detect a variety of road and traffic conditions, as well as driving behavior, issuing accurate suggestions to minimize risk factors.

Future work includes enhancements to the mobile prototype, such as voice alerts and the local integration of the inference engine. The mobile matchmaker devised in [12] will be ported to the target smartphone platform to remove the dependency on centralized reasoners and to reduce bandwidth usage. An extensive experimental campaign is also under planning, to evaluate system performance in each workflow stage. As far as research is concerned, more OBD parameters and smartphone peripherals (*e.g.*, camera, microphone) could be used, in order to detect and feature a larger array of contexts.

Acknowledgments

The authors acknowledge partial support of Apulia Region Strategic Project PS_121 - Telecommunication Facilities and Wireless Sensor Networks in Emergency Management.

References

[1] L. Evans, *Traffic safety and the driver*. Van Nostrand Reinhold, 1991.

[2] BOSCH, *CAN Specification*, 2nd ed., ROBERT BOSCH GmbH, 1991.

[3] S. Colucci, T. Di Noia, A. Pinto, A. Ragone, M. Ruta, and E. Tinelli, “A non-monotonic approach to semantic matchmaking and request refinement in e-marketplaces,” *International Journal of Electronic Commerce*, vol. 12, no. 2, pp. 127–154, 2007.

[4] J. Hu, F. Yan, J. Tian, P. Wang, and K. Cao, “Developing PC-Based Automobile Diagnostic System Based on OBD System,” in *Power and Energy Engineering Conference (APPEEC), 2010 Asia-Pacific*, March 2010, pp. 1 –5.

[5] C. Lin, C. C. Li, S. H. Yang, S. H. Lin, and C. Y. Lin, “Development of On-Line Diagnostics and Real Time Early Warning System for Vehicles,” in *Sensors for Industry Conference, 2005*, 8-10 2005, pp. 45 –51.

[6] J. Lin, S. C. Chen, Y. T. Shin, and S. H. Chen, “A Study on Remote On-Line Diagnostic System for Vehicles by Integrating the Technology of OBD, GPS, and 3G,” in *World Academy of Science, Engineering and Technology, 2009*, aug. 2009, pp. 435–441.

[7] Y. Chen, Z. Xiang, W. Jian, and W. Jiang, “Design and implementation of multi-source vehicular information monitoring system in real time,” in *Automation and Logistics, 2009. ICAL ’09. IEEE International Conference on*, aug. 2009, pp. 1771 –1775.

[8] S. Choi, J. Kim, D. Kwak, P. Angkititrakul, and J. Hansen, “Analysis and Classification of Driver Behavior using In-Vehicle CAN-Bus Information,” in *Biennial Workshop on DSP for In-Vehicle and Mobile Systems*, June 2007.

[9] SAE Standard J1962, “Diagnostic Connector - Equivalent to ISO/DIS 15031-3,” 2002.

[10] SAE Standard J1979, “E/E Diagnostic Test Modes - Equivalent to ISO/DIS 15031-5,” 2002.

[11] W3C Recommendation, “OWL Web Ontology Language,” 2004, http://www.w3.org/TR/owl-features/ last accessed on July 19th, 2010.

[12] M. Ruta, F. Scioscia, and E. Di Sciascio, “Mobile Semantic-Based Matchmaking: A Fuzzy DL Approach,” in *7th Extended Semantic Web Conference (ESWC2010)*, ser. Lecture Notes in Computer Science, vol. 6088. Springer, 2010, pp. 16–30.