A Prototype for Remote Vehicle Diagnostics

Jonas Kuschel[1], Henrik Fagrell[2], Carl Johan Andersson[3]

[1]IT University of Göteborg, Box 8718, 40275 Göteborg, Sweden
jonas.kuschel@ituniv.se
[2]Newmad Technologies AB, Thorildsgatan 7, 41125 Göteborg, Sweden
henrik.fagrell@newmad.se
[3]Volvo Parts AB, Product Support Development,
Dept 31410 ARH 7, 40530 Göteborg, Sweden
carl.johan.andersson@volvo.com

Abstract. The field of Remote Vehicle Diagnostics can be described as the remote management of vehicles equipped with electronic control systems. Despite the great potential that is ascribed Remote Vehicle Diagnostics there are few practical applications that address the needs of end-users. This paper asks how service mechanics remotely can get detailed vehicle data when the driver is concerned about the vehicle's behaviour, or the vehicle's internal control system detects an error. We describe a prototype that enables service mechanics to remotely receive notifications of vehicle diagnostics trouble codes, read real-time usage parameters, and periodic log parameters according to specified rules and filters. The paper concludes with a future outlook on how the architecture can support new kinds of services.

1 Vehicle Electronics

In the automotive industry companies show an increasing interest in Remote Vehicle Diagnostics (RVD). RVD is the remote access, diagnosis and software update of vehicle systems. In 1998, Jameel *et al.* [2] predicted that new vehicles within five years would enable basic telematics services, such as sending status data and error reports via the Internet. While this has shown to be too optimistic, the interest in RVD among vehicle manufacturers, telematics service providers and end-customers is increasing [1].

There are a handful of commercial services, e.g., On Star or Volvo OnCall that all focus on consumer needs such as road assistance and guidance, but applications addressing the needs of service mechanics have not been in the spotlight.

We present an application prototype that aims to support service mechanics in identifying and solving problems remotely. The question we seek to answer is: *How can service mechanics remotely get detailed vehicle data when the driver is concerned about the vehicle's behaviour or the vehicle's internal control system has detected an error?* The application architecture underlying the prototype meets typical industrial requirements, e.g., cost per product unit, operation cost and scalability.

The model that has informed our design is primarily based on the results of an ethnographic field study reported by Kuschel & Ljungberg [3]. They conclude their work by proposing a decentralized approach to RVD. A part of the decentralized approach is to enhance after market mechanics with remote diagnostics access to the customers' vehicles. This perspective contrasts from the prevailing manufacturer-centric model of RVD where the local dealer mechanic plays a minor role or is totally removed.

The material was complemented with interviews and workshops with personnel from Volvo to further detail the requirements. The development was performed in co-operation with three master students using agile development methods. Finally, the prototype was evaluated as a proof-of-concept in a realistic environment.

2 Vehicle Electronics

We first introduce some technical concepts of vehicle diagnostics, since it is not a field normally addressed in HCI research. A modern vehicle is to a large extent controlled by computers, often called Electronic Control Units (ECUs). Figure 1 is an example of the electrical system of a Volvo truck. Several sensors are located all over the vehicle enabling the ECUs to monitor the status of onboard technology (e.g., fuel pressure) and the surrounding environment (e.g., barometric pressure). If a sensor value is outside the allowed range an ECU will signal an error code (sometimes called Diagnostic Trouble Code (DTC)). Error codes are categorized according to how serious the error is. A minor error would not be shown to the driver whereas a major one would force the vehicle to a stand-still position. Each ECU executes software and can thus be programmed to behave in different ways depending upon its application. For example, parameters such as injection ratio can be modified in order to control the performance, emission levels and fuel consumption.

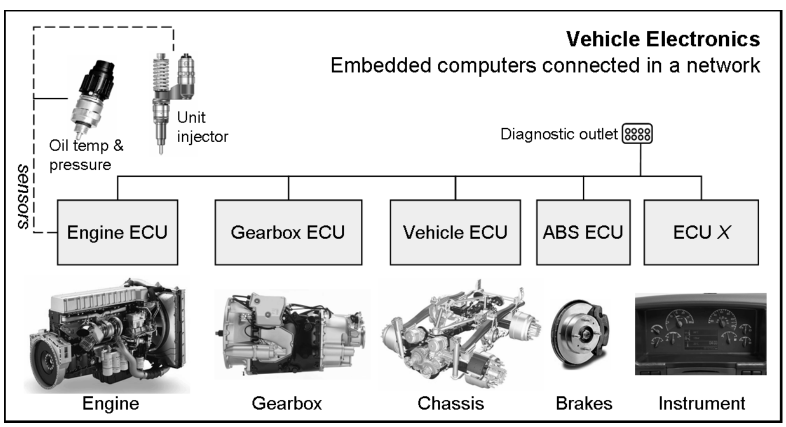

Fig. 1. The figure illustrates the electronic system of a modern vehicle.

Since a vehicle is such a complex piece of technology mechanics of today have to rely on computer programs to perform service. The diagnostic computer application used in the repair shop can be connected to the diagnostic outlet of the vehicle allowing the mechanic to read and reset error codes, read and set parameters, run scripted tests, and update the software of individual ECUs.

3 Meeting the Requirements

The prototype outlined in this paper aims to meet several requirements regarding mechanics' work practice, these are: (1) alerts when certain error codes occur; (2) reading run-time parameters; and (3) recording predefined parameters according to rules and filters.

By getting notifications about error codes the mechanic is able to start the diagnostic process and take action prior to the customer getting to the workshop. This would make the diagnostics process more efficient and thus improve customer satisfaction.

An important conclusion made by Kuschel & Ljungberg [3] is the fact that technicians define their jobs as identifying problems experienced by the customer, as opposed to technical problems as such. This requires mechanics to be able to analyze sets of run-time parameters since not all customer experienced problems are equal to error codes.

Both a problem description by the customer and diagnostic data stored in the ECUs are valuable clues that help the mechanic to define the problem. However, mechanics we have interviewed repeatedly point to the lack of relevant data in the time frame when a customer experiences a problem or an error code is set by the system. This becomes more evident regarding more difficult problems that require extended test driving and determining data defined by the mechanic. Hence, the prototype aims to enable mechanics to remotely define and record parameters according to rules and filters.

There are also several commercial and environmental constraints that have to be considered. Future telematics services must accord to the principle of low unit and maintenance costs.

4 The prototype

The prototype builds upon a uniform remote communication module. It handles network breakdowns, roaming between networks and truncation of the data stream.

A PDA was used as onboard client because the platform allows for speedy development. In a commercial application the onboard functionality would be developed for a less sophisticated device (i.e., ECU) to reduce the hardware costs. The server is PC-based with a Java platform. The mechanic interacts with the system via a web browser (see figure 2). Needless to say, the communication with the vehicles must be wireless.

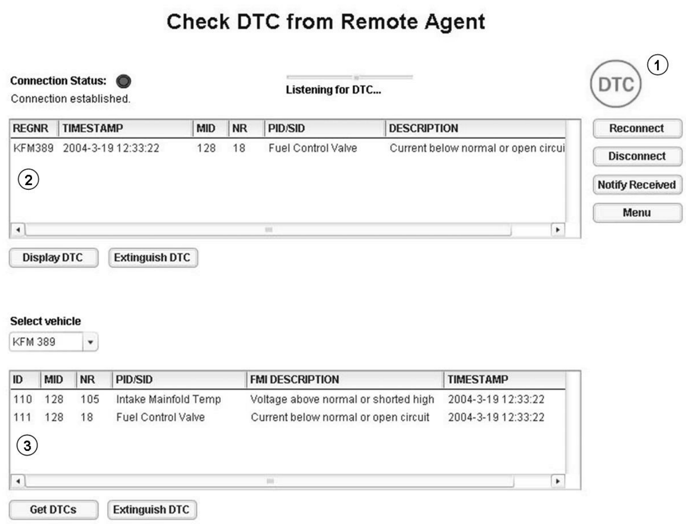

Fig. 2. A screenshot of the web interface, with (1) a notification about a recently set error code, (2) data about the error code, and (3) data about previous error codes.

Three services have been implemented in the communication module; an error code notification service, a service for run-time parameter reading, and a service for recording of predefined parameters.

The notification service informs the mechanic about error codes recently occurred. Only changes in the ECU states are sent to the server. Accordingly, the server will always keep the latest state about the vehicle and the mechanic is able to get data about a vehicle's state immediately without waiting. While connected to the server a notification is sent in order to get the mechanic's attention (see figure 2, note 1).

The service for reading run-time parameters enables the mechanic to remotely read parameters, e.g., boost pressure. This is a rather straightforward service that, by getting a request from the web client, requests a parameter from the ECU chosen and sends the data back to the mechanic.

The third service enables mechanics to log parameters according to five different settings; parameter, total time, interval, frequency, and rules. The mechanic can select different parameters and define how long each parameter should be read. An interval scheduling how often data is sent to the server and the frequency of reading the parameter from the ECU are further settings to be defined. Finally, rules can be set that define, for instance, a data range between which the parameter should be logged. The log service enables mechanics to conduct more profound analysis remotely. Up until now, so-called flight recorders, storing data in a hardware unit, have to be used for this purpose. Our prototype enables mechanics to continuously analyse the data and, most important, change the settings during operation.

Summarizing all three services they operate according to cost effectiveness and mobile computing constraints. Most important, our ambition has been to develop an architecture that enables a smooth interaction with the application. Error codes or requested data are continuously sent to the server and cached in a database, thus offering a quick access and only notifying a mechanic when any changes occur.

The proof-of-concept evaluation was conducted at a test drive track (figure 3). The setting of the evaluation was (1) a laptop connected to the Internet and running a Web browser; (2) a PDA and an interface jacket connected to the diagnostics outlet of the Volvo FH 12 (3). The PDA was connected to the Internet via Bluetooth to a GPRS phone giving us wireless access to the truck when it was on the road (4).

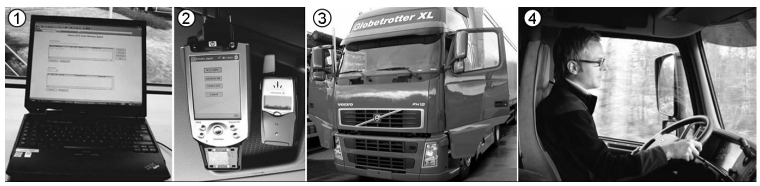

Fig. 3. The setting of the proof-of-concept evaluation.

All of the services of the prototype were successfully performed. Despite low network coverage of the test track site all data could be transferred via GPRS. This indicates the level of data efficiency we were able to achieve.

5 Discussion

In this paper we have addressed the issues of how service mechanics remotely can be provided detailed vehicle data when the driver is concerned about the vehicle's behaviour or the vehicle's internal control system has found an error. The proof-of-concept prototype has been tested under realistic conditions with promising results. As a next step a new service for remote parameter setting and ECU software updates is going to be introduced. We also plan to evaluate the complete systems on professional mechanics. In doing this, we believe that there is a great potential to find new requirements on how mechanics want to remotely interact with the vehicles and the customers in order to complement the diagnostic data.

References

[1]Gartner G2, Telematics Industry Outlook: Think Outside the Vehicle, RPT-0902-0163 Gartner G2, 2002.

[2] Jameel A., Stuempfle M., Jiang D. and Fuchs A. [1998], Web on Wheels: Toward Internet-enabled Cars, *IEEE Computer* 31(1), 69-76.

[3] Kuschel J. and Ljungberg F. [2004], Decentralized Remote Diagnostics: A Study of Diagnostics in the Marine Industry, to be published in *Proceedings of HCI 2004*, Leeds, UK.