CarCoach: A Generalized Layered Architecture for Educational Car Systems

Taly Sharon, Ted Selker
MIT Media Laboratory
20 Ames st.
Cambridge, MA 02139 USA
{taly, selker}@media.mit.edu

Lars Wagner
RWTH Aachen
Wuellnerstrasse 1
52066 Aachen, Germany
lwagner@media.mit.edu

Ariel J. Frank
Dept. of Computer Science
Bar-Ilan University
Ramat-Gan, Israel 52900
ariel@cs.biu.ac.il

Abstract

Car accidents are a major concern. Consequently, a lot of research is carried out on car user interfaces. For each such research, usually a special simulator or car is developed, algorithms and tools are redeveloped, and similar issues arise. We propose CarCoach, an educational car system, based on a generalized layered architecture. We present the system design, the intelligent modular architecture, its layers, including details of some of its relevant modules. Using the Chrysler 300M IT-Edition car as a platform, a prototype was implemented and initial experimentation was carried out and is reported. We demonstrate that CarCoach provides a flexible environment for car research and support of varied car applications.

1. Introduction

Motor vehicle crashes are the leading killer of Americans between the ages of one and 29. For example, in 1999, an average of 112 people were killed in motor vehicle crashes every day – one every 13 minutes. While the greatest cost is incalculable human suffering and loss, motor vehicle crashes also cost Americans an estimated $192.2 billion in 1999 [19].

Moreover, the US Department of Transportation safety programs were unable to meet their own fatalities reduction targets. For example, in 2002, the target rate of reduction of highway fatalities was 1.4 fatalities per 100 million vehicle-miles traveled. However, this target was not met, and the actual estimated rate was 1.5 [2].

One of the causes for this is the fact that many licensed drivers are not good drivers – some have forgotten the rules over time; others have developed bad habits along the way; and they often drive in an automatic-unconscious manner [16].

Consequently, to cope with this severe problem, driver support and warning systems, are being built (for example, see [15]). Significant research is being carried out in developing such driver support systems. These systems have many functions. Janssen et al. [12] describes nine types of basic driver support functions (examples in parenthesis):

(1) Enhancing information (increasing visibility by retroflection).
(2) Augmentation (special information about icy patches).
(3) Warning (against speeding or other violations).
(4) Advice (to take a less congested route).
(5) Explanation (reason for delay, e.g., accident ahead).
(6) Instruction (feedback about incorrect action).
(7) Intervention (speed delimiter).
(8) Substitute or secondary control (cooperative driving).
(9) Autonomous or primary control (robot driving).

As in many other areas, education can help. So educational car systems that warn and instruct the driver on mistakes are being researched and developed. It has been suggested that such technological solutions can provide feedback on driving ability, warn about dangers, and ultimately improve driving performance [9].

However, the platforms for such research are usually either a special simulator or proprietary cars that provide specific support for the developed application. We propose herein the Chrysler 300M IT-Edition (*300M for short*) as a platform for these research systems [21]. On top of it, we have designed CarCoach, a generalized layered architecture, to provide full support for research and development of applications for educational car systems.

The contribution of this paper is in presenting a design and architecture for an educational car system, CarCoach, and demonstrating that it provides a flexible environment for car research and support of varied car applications.

The structure of this paper is as follows. In the next section, the design of an educational car system is outlined. In section 3, a supporting generalized layered architecture is proposed. Then, an implementation platform, the 300M IT-Edition is introduced. In section 5, a prototype CarCoach is described. Then, initial experimentations are reported. In section 6, additional applications are suggested. Lastly, conclusions and discussion are presented.

2. Educational Car System

As has been suggested, in addition to safety systems, educational systems are required. Educational car systems are meant to continuously monitor and train drivers to drive at their best. The assumption is that these systems can improve the overall skill of a driver in all cases and that by this overall improvement, they could also facilitate either a better response from the driver at the time of an emergency or reduce cognitive load during the emergency.

2.1. System Design

The design for an educational car system is outlined here using the following approach. The system is meant to improve performance – not to teach unlicensed drivers how to drive, nor to warn drivers about their driving (at least not as the main task). Its role is to provide feedback, usually **after** the driving mistake has been made and to try and educate drivers to drive at their best. An important feature of the system is the use of calculated feedback, so as to separate the input from the output and make it more versatile and human like, for instance, by giving both criticism and affirmation (positive feedback on improvements).

This design is based on previous ideas and works in the area, especially PSALM [6], and recommendations regarding driving-education systems, while also taking into account known considerations of human factors.

The design addresses three main aspects of driving: illegal, unsafe, and inefficient driving behavior, as follows:

1) Illegal driving – such as turning or changing lanes without signaling.
2) Unsafe driving – such as using excessive force on the brake that might increase the risk of being hit from behind.
3) Inefficient driving – such as using excessive force on the throttle, which reduces gas mileage as well as the engine's lifetime.

Following the identification of driver behavior, the design aims to balance between the following guidelines:

- Personalized interaction – based on learning each driver's profile, or history of behavior.
- Multimodal interaction – use different channels of feedback rather than audio, such as tactile and some visual ones.
- Humanly interaction – provide versatile feedback (not the same reaction or verbal comments on the same mistake all the time) and positive reinforcements in addition to criticism.
- Quick reaction – give the feedback as soon as possible after the mistake, to prevent confusion.
- No information overload – give the feedback when the driver is not overloaded.
- Post driving information – provide statistics and information for after drive analysis.

2.2. Software Design

The proposed software design for the educational car system is presented in Figure 1. It includes the Car Interface (1), States & Behavior Identifier (2), Feedback Generator (3), Control Panel (4) and supporting data repositories.

The modules' functionalities are as follows:

(1) Car Interface

The Car Interface module interacts with the car and reads the sensor inputs into a representation of the car in memory. A secondary role of that interface is to activate output devices in the car.

(2) States & Behavior Identifier

The States & Behavior Identifier (SBI) module tests the state of the car and identifies states and driver behaviors (such as "did not look in the rear view mirror when pressed the brake"). It obtains extra knowledge from a task characteristics knowledge repository. As a result, SBI updates the driver history to reflect the new state.

SBI manages the driver history repository. It keeps a log of every criticism scenario as counters of successes and failures per mileage. The scenarios may be organized in groups of mistakes of the same type; for instance, all signaling mistakes (when changing lanes, when turning, when pulling over, etc.) can be grouped together.

(3) Feedback Generator

The Feedback Generator (FG) module is the core of the system. FG is activated when a new state has occurred. Its task is to react to the new state. It takes into account the driver's history, the feedback history, and executes a series of rules to generate feedback. Its role is to decide when it is a good time to interrupt (e.g., not in the middle of a turn), how much feedback should the driver get (e.g., based on knobs setup), and using which modality. For example, previous results from lab environments have indicated that tactile feedback is effective in conveying messages to drivers [4, 26].

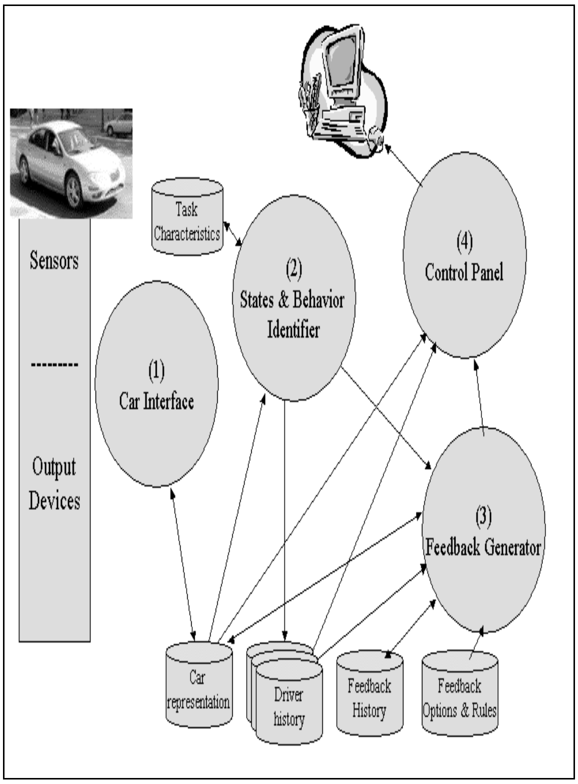

Figure 1 – Software design

When using several channels – effectivity increases [10]. Therefore, the design includes a combination of feedback methods, such as tactile, visual, and audio feedback channels. Tactile feedback is provided as controlled vibrations of the steering wheel, accelerator, brake, and the seat. Guidelines followed here regarding tactile feedback are [10]:

- It should be given right after the task or it will not be understood.
- It should be given with the relevant device or it may not be understood. For example, steering vibration for steering mistakes, throttle vibration for mistakes related to throttle/speed, brake vibration for mistakes related to braking, etc.

Other relevant guidelines for audio feedback considered here are [24]:

- Praise should sound sincere.
- Critique should be gentle, and given sparingly.
- Novices prefer more flattery while for experts the compliments should be subtler by picking up more intricate material and by noticing detail.

FG uses the following information to make the feedback decision:

- Setup knobs (Car Representation) – depending on the switches setting, the amount of feedback is increased or reduced. For instance, when the criticism switch is all the way down – no criticism feedback is provided, and vice versa. Because people tend to switch off things and forget to switch them on again, the setup knobs should be digital, to be reset by the system as needed.
- Stress/distraction level – stress/distraction level gauges that take into account several factors such as weather, number of passengers in the car, speed and driving patterns – they affect the feedback amount and timing decision.
- Driver history – including the repetition pattern of a mistake and a group of mistakes, enabling prioritization of feedback messages.
- Feedback history – including what feedback was already given to the driver, when, and how effective it was, enabling further provision of variable (non-repetitive), effective feedback, using the right modality.
- Priorities – are established on the driving mistakes to address. The priorities are based on the ratio of failures vs. successes for each mistake and group of mistakes, the frequency per mileage, and on the overall severity of the mistake. The focus is given to higher priority mistakes; while lower priority mistakes are not related to until the higher priority mistakes are overcome.
- Feedback options and rules – to select the feedback from. This repository stores an absolute priority of severity of driving mistakes (e.g., changing lanes without signaling is more dangerous than over-exerting the car). In addition,

it stores all the feedback options for each mistake. Generally, each mistake has a few associated audio messages and often tactile or visual feedback as well. Also, it has affirmation feedback options to be used when the driver performed well and did not make the mistake. Finally, each feedback option has a rating of expertise level, from novice to expert.

(4) Control Panel

The Control Panel module monitors the states of the car and the driver, and displays that information. It enables selecting drivers, giving setup parameters for all the driver support systems in the car, and eventually downloading information to be further analyzed and reported.

3. Generalized Layered Architecture

This section describes a generalized layered architecture that can benefit application developers. The purpose of this architecture is to avoid the need for each researcher or developer to program the low-level sensors and to develop new modules for similar components, such as stress detectors. This can be achieved by adding to cars a computer that includes support for applications using generic modules.

This intelligent architecture includes five layers (bottom up): Sensors, Interfaces, Car Facilitator, Intelligent Mediators, and Application (see Figure 2). On the bottom end, the Sensors and Interfaces layers are hardware-dependent and proprietary to each car; they will be described in the next section as part of our research platform. On the other side, the top, the Application layer, the aforesaid educational system resides, but any other car application can be implemented there. The conceptually new layers are the Intelligent Mediators and the Car Facilitator.

3.1. Application Layer

This architecture supports experimentation with varied user applications involving sensors in the car. Many applications could benefit from this architecture, including warning systems such as Collision Avoidance Systems (CAS).

Even though the educational car system is our exemplary application, we are already using it to develop other applications. In chapter 7, we describe here several of those that we have implemented.

As for the educational car system, from the software design, the modules (3) Feedback Generator and (4) Control Panel are part of the Application Layer. In fact, these are the modules that make the system what it is, and also interact with the user.

Application: coaching, controlled warnings….
Intelligent Mediators: stress, direction, behavior …
Car Facilitator
Interfaces: boards, DataPump, FaceLab …
Sensors: camera, J1815, IR, pressure sensors …

Figure 2 – Generalized layered architecture

3.2 Intelligent Mediators Layer

The Intelligent Mediators layer includes modules that serve the applications. As an example, three generic modules that serve many applications are described here. First, we describe the system's module (1) States & Behavior Identification. Afterwards, we describe other modules that serve the application, such as Stress Identification and Distraction Identification. These two modules serve module (3) Feedback Generator in the Application layer, by providing it the information needed to decide whether to issue feedback, postpone it, or avoid it completely, depending on the level of stress and distraction the driver is experiencing.

It is important to note that this layer's components are often not mature enough to be integrated into real working systems. However, considering the vast amount of resources devoted to research in this area, the assumption is that with time there will be more classifiers robust enough for implementation. Also, the separation of these components from the application layer makes it easier to handle these components separately so as to focus on the applications that use them.

In the following subsections, we analyze these components and give some examples for possible car and driver related factors and sensors needed to detect and rate the aforesaid relevant states (some based on experiments reported in section 6).

3.2.1 Driving Behavior Identification

Some driving maneuvers can be identified and even predicted [13,14,20]. The Driving Behavior Identification module can identify driving maneuvers (e.g., lane keeping, lane changing, turning etc.), as well as driving mistakes (e.g., unsteady steering, lane changing or turning without signaling, etc.) to be used for a variety of applications.

3.2.2 Stress

Previous works have attempted to identify stress in driving [8]. Using their and others' conclusions, as well as common sense, a stress model can be developed. This model can include different stress inducers and calculate a measure of stress to be used by upper-level applications.

A few examples for stress inducers are described in Table 1, with possible sensors to identify their existence. For instance, bad conditions of weather and environment can increase the difficulty and load on the driver: wetness, ice, fog, dark, etc. These can be detected by using telemetric sensors in the car, such as the external temperature and humidity, or even the activity of the windshield wipers, as well as by using external information such as weather forecasts and reports.

A different example for stress can be based on the driving activity, such as driving in reverse, or performing maneuvers such as changing lanes, turning, etc.

Factor	Detection
Bad conditions – wetness, icy roads, fog, darkness	Humidity, temperature, darkness sensors, wipers on, lights on, external information
Reverse	Car gear state
Changing lanes	Specific classifier
Intersections, rotaries	GPS
Merging into highway	GPS
Certain risky locations	GPS with reports from other drivers or local police (accident leading areas)
General stress (some aspects)	Grip force on the steering wheel

Table 1 – Stress factors and sensors

Another example, based on location, is stressful locations that can be identified based on a compiled database of stressful areas coordinates combined with use of a GPS. This database can include, for example, drivers' subjective reports, all ramps merging into highways, or reports from the police about risky driving areas. A different approach, based on the subjective behavior of drivers may be potentially detected by using pressure sensors on the steering wheel, with the assumption that the amount of pressure applied on the steering wheel often increases when the driver is in stress (similar to pressure applied on a computer mouse [23]).

3.2.3 Distraction

To answer a different problem, of driver distraction, many factors can be taken into account, as presented in the examples of Table 2. Such factors can be driver in active conversation (can be detected by using a microphone, the cell phone activity), driver drinking, handling the radio, or even just not looking at the road.

Factor	Detection
Driver conversing	Microphone, cell phone in use
Driver drinking	Cup holder is active
Driver handling radio/AC	IR sensor around the HVAC
Driver does not look at the road	Cameras/Facelab

Table 2 – Distraction factors and sensors

3.3. Car Facilitator Layer

This layer facilitates hardware transparency. It includes the module (1) Car Interface. This module is responsible of reading the different proprietary interfaces to the sensors and providing the higher layers a single, standard, portable, and well defined data interface, regardless of the proprietary hardware in a specific car.

3.4. Interfaces Layer

This layer includes the low level interfaces to the car sensors.

3.5. Sensor Layer

This layer includes the sensors in the car. It is proprietary for each type of car and its specific sensors in use.

4. The 300M IT-Edition

Since simulators do not provide as real an experience as driving cars [3], the research platform is a real car, the 300M IT-Edition (see Figure 3 for a picture of the car from the outside and Figure 4 for the interior). The 300M is a regular model offered by Chrysler, while the additional sensors and devices have provided us with this special model called the "IT-Edition". The 300M IT-Edition is a highly instrumented research vehicle equipped with many sensors and devices [21], as presented in Figure 5.

Figure 3 – The 300M IT-Edition

Figure 4 – The 300M IT-Edition interior

This section describes together the two bottom layers of the architecture: Interfaces and Sensors, as implemented in the 300M. At the end of the end of this section, the components are classified to each layer.

4.1 Infrastructure

The computation center and the interfaces are housed in the car's trunk. It includes an Ethernet/802.11 communication network with a wireless access point. The infrastructure is flexible; its core is a set of NetBurners, which are programmable interface boards connecting sensors and serial devices to the local Ethernet [17]. They are configured here with either UDP or Telnet network protocol, as suitable for each device. At any time, one application computer can read the data from the devices by connecting to the network and setting up the boards to send UDP information to it through HTTP.

4.2 Car devices

Each relevant device or sensor in the car is described herein:

- Engine data: speed, throttle position, brake pressure, RPM, etc. The access protocol is based on the J1850 protocol [11].
- A set of sensors accessible via a data acquisition board (nicknamed DataPump): steering angle, pressure sensors in the seats, cup holders, arm rests and all the car pedals, cellular phone activity (and disabling) sensor, and infrared sensors for legs position on the pedals (including the special dead-pedal to rest the left foot).

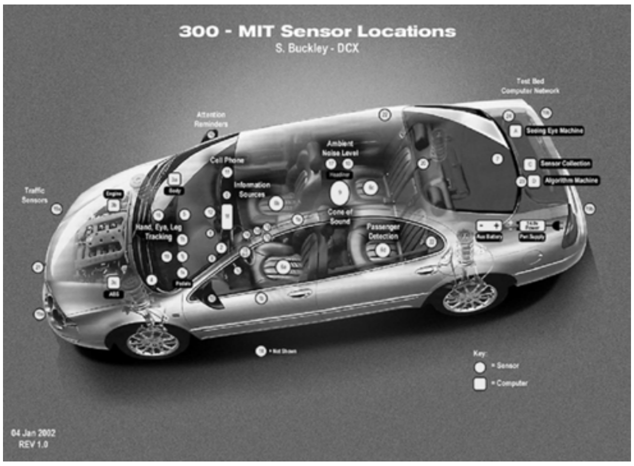

Figure 5 – The 300M IT-Edition sensors

- Controllable lights on the mirrors and car sides (see Figures 6 and 7 respectively), accessed via the DataPump.
- A Busy and a Warning (two colors – yellow and red) combination of lights and buttons, all accessible via the DataPump (see Figure 7).
- Vibrators in the steering wheel, driver seat, gas and brake pedals, and setup knobs, allowing affirmation and criticism intensity setup. They are built using a modified iRX [22].
- Controllable standard lights in the instrument panel, such as signal, brake, fuel, and warnings.
- Infrared sensor built around the Heating Ventilation and A/C (HVAC) and the radio controller (see the frame around the radio area in Figure 3). This sensor is capable of sensing movement and location around the HVAC.

Figure 6 – Light on the right mirror

Figure 7 – Light on the car side

- Pressure sensors in the steering wheel and in the gearshift. The sensors are meant to detect the location of the hands and the amount of pressure applied on the devices (In the process of interfacing to the car network).
- Global Positioning System (GPS) [5] to detect the location of the car, available via Telnet with GAR NMEA protocol [18].
- Other sensors in work such as BlueEyes camera [1], special bike warning lights, etc.

Figure 8 – Busy and warning lights/buttons

From the above components, those that belong to the Interface layer are: iRXs, FaceLab, NetBurners, DataPump. The rest are the sensors that belong to the Sensors layer.

5. Prototype: CarCoach

To demonstrate the model and architecture of the educational system, a prototype, called CarCoach, was designed and implemented based on the 300M that includes some scenarios of user warnings and feedback, and also some stress and distraction considerations.

The prototype uses basic car sensors, ones that exist in any standard car, or ones very cheap to install. A detailed summary of them is provided in Table 3.

Device	Sensors and effectors used
CarCoach kit	Vibrators + setup knobs.
Amplifier	
J1850 interface	RPM, turn signals, speed, gear state.
Extra sensors and devices (Datapump)	Brake pressure, steering angle, cell phone sensor, warning and busy lights and buttons.

Table 3 – CarCoach sensors and effectors

CarCoach has five scenarios implemented, which are summarized in Table 4:

1) Over-exerting the car
2) Strong braking
3) Low gear
4) Turn without signaling
5) Turn with signaling.

CarCoach provides both criticism (scenarios 1-4) and affirmation (scenario 5). It provides audio and tactile feedback. The tactile feedback is immediate and uses the most appropriate device: steering wheel for mistakes related to steering/turns, throttle and brake vibration for mistakes related to gas and brake. For the affirmation, it uses the seat vibration massage as a device that gives pleasure/reward for good actions.

In addition, the driver has full control over the feedback by using the setup knobs and may switch off the criticism and/or the affirmation at any time. In order to demonstrate the load and stress prevention, whenever the car is in reverse or there is a cell phone activity, the feedback is either switched off completely (reverse) or does not use the audio channel (cell phone activity).

Action	Feedback	Type
Over exerting the car (RPM>3000)	Throttle vibrates, Audio: “Easy on gas”	Criticism
Strong braking (Brake pressure>2100)	Brake vibrates, Audio: “Brake gently”	Criticism
Low gear (instead of Drive)	Audio: ”Gear is low”	Criticism
Turn without signaling	Steering wheel vibrates, Audio: “Please signal”	Criticism
Turn with signaling	Seat vibrates, Audio: “Thanks for signaling”	Affirma-tion

Table 4 – CarCoach scenarios

To demonstrate this “Busy” state, the “Busy” light is switched on as long as the system is in “Busy” mode and does not generate feedback. Due to the lack of a suitable display in the 300M, to demonstrate the informing of the drivers on the level of their driving, use is made of the warning lights. When the driver has made three mistakes, the amber warning light is turned on. After five mistakes, the amber is turned off and the red is turned on. Once the driver acknowledges getting the information, by pressing the button in the middle of the warning device, the light switches off. Note that the warning light is located on the left side of the driver (see Figure 8), and is rather private to the driver. Table 5 presents this CarCoach effectors model.

Sensor	Effect
Setup knobs – criticism off	Cancels all criticism feedback
Setup knobs – affirmation off	Cancels all affirmation feedback
Cell phone is in active call	Eliminates audio messages
Reverse gear	Busy light turns on, cancels all feedback
3rd mistake this drive	Yellow warning turns on
5th mistake this drive	Red warning turns on

Table 5 – CarCoach effectors model

6.1 CarCoach Trials

An initial experience with CarCoach [25], mostly in demonstrations, has shown that CarCoach is appealing to drivers. Those who tried CarCoach have shown strong reaction and excitement from it, especially from the tactile feedback. In one case, a driver used CarCoach for a period of 1/2 hour and then switched to different software in the car. The driver and the passengers noticed that also during this period, when CarCoach was not active, he improved his driving and made fewer mistakes, especially signaling mistakes.

This has shown us that CarCoach has good potential to improve driving performance, as well as to be appealing to the drivers. The main question about CarCoach is how drivers will accept it and use it on a long-term basis. Many concepts in CarCoach could be further tested. The possibility of delaying feedback when a driver is in a complex maneuver is generally important element that would pertain to other scenarios as well.

6.2 Postponing Educational Messages

Many beneficial driver support systems pose the risk of overloading drivers when issuing non-urgent messages. One solution to this problem is to slightly delay these messages when the driver is overloaded [27]. There has been some evidence from the field of education that delaying feedback can even lead to better performance [7]. For an educational car system like CarCoach, it is an important issue to explore. Therefore, an experiment in the context of CarCoach was conducted.

28 subjects (14-M, 16-F) performed a driving task, fast acceleration, 10 times. During the task, the experiment system issued instructional messages guiding them to a certain acceleration pace. Half of the subjects got a delayed messages, i.e., at the end of the acceleration maneuver. The delayed feedback group performed significantly better than the immediate feedback group (graded 148.43 vs. 81.88 $t(20.287)=1.748$, $p=0.048$) [25].

6.3 Stress Detection

Based on the study of 25 subjects that reported the level of stress while driving, the following scenarios have indicated high stress possibility: backing up, wide turns and certain locations such as intersections, rotaries, parking lots or merging lanes.

6.4 Driving Behavior/Mistakes Identification

In addition to well-known algorithms that can identify driving maneuvers, we tried to identify driving mistakes. Using 4 drivers, of which one is a professional driver/teacher, we have found that many of these mistakes are surprisingly easy to recognize, while of course, many are very difficult.

For example (recognition accuracy is in parenthesis): braking without looking at the rear-view mirror (100%), fast/unsafe turns (90%), unsteady steering (95%), turning without signaling (100% in low-medium speed, ramps 0%).

7. Additional Applications

The generalized architecture for the car can benefit many applications, even those that are more than educational. Two such applications were developed using the 300M, and described herein: Cellular Phone Control, and Controlled Warnings.

7.1 Cellular Phone Control

This application is meant to suppress cellular phone rings when the driver is under stress. It identifies that the driver is under stress and suppresses it as long as the driver is focusing on the traffic situation (usually around 10-15 sec.). The system uses the stress module in the Intelligent Mediator layer, and deploys the "Busy" button (see Figure 8) to show that state.

7.2 Controlled Warnings

This application suppresses low priority warnings in the instrument panel when the driver is under stress. Whenever a warning or maintenance message occurs, the warning priority is tested (for instance, low priority – low washing fluid or fuel; high priority – engine heating or low oil pressure). When the stressful situation is over, the warnings appear with a special Warning light (see Figure 8), enabling the driver to acknowledge the warning and view details on a separate display.

Furthermore, to prevent further distraction, by pressing the Warning Button, the dashboard's warning lights are turned off for the rest of the drive, since the driver is already aware of the problem. The system uses the stress module in the Intelligent Mediator layer, and deploys the "Warning" light/button (see Figure 8) to show that state.

8. Conclusions and Discussion

In this paper we proposed CarCoach, a model and architecture for educational car systems. We described a platform and a prototype, as well as initial experimentation with its architecture layers, a prototype, and some other applications. The experimentations taught us the following:

- Delays in educational messages can be used when needed to prevent driver overload and stress.
- Some stress situations can be identified based on the use of simple measures.
- Similarly, some driving behaviors and mistakes can be easily identified.
- Additional applications, such as Controlled Warning and Cellular Phone Control, can efficiently deploy the generalized architecture and its intelligent mediators (e.g., stress, distraction).

As a result of this undertaking, we realized some interesting benefits, as follows.

The modularized architecture enables us to keep the platform (two bottom layers) transparent to the top level layers via the Car Facilitator layer. This alleviates the necessity to develop a different application for each type of car.

In addition, the Car Facilitator layer can serve many applications at the same time. This alleviates the problem of the 300M lower levels resource contention (where any interface board is only capable of serving one entity at a time). This also enables a simplified implementation of these lower levels for any car platform.

The same considerations apply to each module in the Intelligent Mediators layer (see section 3.2). Considering the current immaturity of this layer's components, separating them from the application, and using a modular structure, enables their replacement as research advances.

In summary, CarCoach, as a multi-modal, multi-sensor, educational car system, provides a flexible environment for car research and support of varied car applications.

9. Acknowledgements

We thank DaimlerChrysler and Motorola for their support of this project, especially Steve Buckley and Jim Kemp.

10. References

[1] BlueEyes, http://www.almaden.ibm.com/cs/BlueEyes/index.html, 2003.

[2] Performance and Accountability Report, FY2002 U.S. Department of Transportation, Bureau of Transportation Statistics, http://www.dot.gov/perfacc2002/Entire report.htm, 2002.

[3] Denn, P.R.M., Motion Effects on Driver Training, White paper, England, http://www.q3000.com/pdf/sim5.pdf, 1994.

[4] Enriquez M., Afonin O., Yager B., Maclean K., A Pneumatic System for the Driving Environments, PUI 2001, 2001.

[5] What is GPS, http://www.garmin.com/aboutGPS, 2003.

[6] Groeger J.A., Degrees of Freedom and the Limits on Learning: Support Needs of Inexperienced Drivers, in: Parkes M. and Franzen S. (Ed.), Driving Future Vehicles, Taylor & Francis, pp. 77-88, 1993.

[7] Groeger J.A., Understanding Driving: Applying cognitive psychology to a complex everyday task, Psychology Press, UK, 2000.

[8] Healey J. and Picard R.W., SmartCar: Detecting Driver Stress, in *Proceedings of ICPR'00*, Barcelona, Spain, 2000.

[9] Hutton K.A., Sibley C.G., Harper D.N. and Hunt M., Modifying driver behaviour with passenger feedback, in *Transportation Research Part F: Traffic Psychology and Behaviour* 4(4), pp 257-269, 2001.

[10] MMI of Warning Systems in Vehicles, draft, ISO/TC22/SC13/WG8 N341, 2002.

[11] J1850: Class B Data Communications Network Interface, Vehicle Architecture for Data Communications Standards, *SAE*, 2001.

[12] Janssen W.H., Alm H., Michon J.A., Smiley A., Driver Support, in Michon J.A. (Ed.), Generic Intelligent Driver Support, Taylor & Francis, pp. 53-66, 1993.

[13] Kuge N., Yamamura T., and Shimoyama O., A driver behavior recognition method based on a driver model framework, *Society of Automotive Engineers Publication*, 1998.

[14] Liu A. and Pentland A., Towards real-time recognition of driver intentions, *In Proceedings of the 1997 IEEE Intelligent Transportation Systems Conference,* 1997.

[15] Michon J.A. (Ed.), Generic Intelligent Driver Support, Taylor & Francis, 1993.

[16] Nardi B.A., Context and Consciousness Activity Theory and Human-Computer Interaction, MIT Press, 1997.

[17] http://www.netburner.com/, 2003.

[18] NMEA 0183 v2.0 Protocol, http://www.scnt01426.pwp.blueyonder.co.uk/Articles/GPS/NMEA.htm, 2000.

[19] US National Safety Council, Safety Agenda for the Nation – On the Road, http://www.nsc.org/safetyagenda/transp.htm, 2000.

[20] Oliver N., and Pentland A., Graphical Models for Driver Behavior Recognition in a SmartCar, *Proceedings of IEEE Intl. Conference on Intelligent Vehicles 2000,* Detroit, Michigan, 2000.

[21] Pompei F.J., Sharon T., Buckley S, Kemp J., An Automobile-Integrated System for Assessing and Reacting to Driver Cognitive Load, *Proceedings of IEEE/SAE Convergence 2002*, pp. 411-416, 2002.

[22] Poor R.D., The iRX 2.2 – where Atoms Meet Bits, http://web.media.mit.edu/~ayb/irx/irx2/, 1999.

[23] Qi Y., Reynolds C., and Picard R.W., "The Bayes Point Machine for Computer-User Frustration via PressureMouse", *The 2001 Workshop on Perceptive User Interfaces (PUI 01),* Orlando, Florida, 2001.

[24] Reeves B. and Nass C., The Media Equation, CLSI publications, 1996.

[25] Sharon T., An Advanced Driver Warning Framework Incorporating Educational Warnings, Master Thesis, MIT Media Laboratory, http://xenia.media.mit.edu/~taly/publications/, 2003

[26] Tijerina L., Johnston S., Parmer E., Pham H. A., and Winterbottom M. D., Preliminary Studies in Haptic Displays for Rear-end Collision Avoidance System and Adaptive Cruise Control System Applications, U.S. Department of Transportation technical report, Sep 2000.

[27] Verwey W.B., How can we Prevent Overload of the Driver? In: Parkes M. and Franzen S. (Eds.), *Driving Future Vehicles*, Taylor & Francis, pp. 235-244, 1993.