Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Context-aware smart car: from model to prototype*

Jie SUN, Zhao-hui WU, Gang PAN[†‡]

(*College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*)

[†]E-mail: gpan@zju.edu.cn

Received Mar. 5, 2008; Revision accepted Oct. 8, 2008; Crosschecked Apr. 27, 2009

Abstract: Smart cars are promising application domain for ubiquitous computing. Context-awareness is the key feature of a smart car for safer and easier driving. Despite many industrial innovations and academic progresses have been made, we find a lack of fully context-aware smart cars. This study presents a general architecture of smart cars from the viewpoint of context-awareness. A hierarchical context model is proposed for description of the complex driving environment. A smart car prototype including software platform and hardware infrastructures is built to provide the running environment for the context model and applications. Two performance metrics were evaluated: accuracy of the context situation recognition and efficiency of the smart car. The whole response time of context situation recognition is nearly 1.4 s for one person, which is acceptable for non-time critical applications in a smart car.

Key words: Smart car, Intelligent vehicle, Context-aware, Ubiquitous computing
doi:10.1631/jzus.A0820154 **Document code:** A **CLC number:** TP39

INTRODUCTION

Cars are becoming important private places frequently used in daily life. However, they also bring many problems, such as traffic congestions and accidents. A smart car aims at assisting its driver with easier driving, less workload and less chance of getting injured (Moite, 1992). For this purpose, a smart car must be able to sense, analyze, predict and react to the road environment, which is the key feature of smart cars: context-awareness.

Lots of technologies have been developed in the past decade, such as intelligent transportation systems (ITS) (Wang *et al.*, 2006) and the advanced driver assistant system (ADAS) (Küçükay and Bergholz, 2004). However, current smart cars are not really context-aware. Only a few types of the information of road environments, which is called contexts, are utilized. Besides, most of current smart cars lack complex reasoning. These drawbacks limit the smart car's ability of assisting the driving task efficiently and safely. This research focuses on how to build a context-aware smart car.

The remainder of this paper is organized as follows. Section 2 introduces the related work on smart cars. A general description of a smart car is given in Section 3. Section 4 proposes a hierarchical context model for comprehensive definition and classification of information in a smart car environment. The smart car prototype, including the hardware infrastructure and software platform, is presented in Section 5. The performance evaluation is shown in Section 6 and the conclusions are given in Section 7.

RELATED WORK

In the past decade, many researches from academic and industrial communities have been made on smart cars. The following is a summary of the major progresses in this field.

[‡] Corresponding author

[*] Project supported by the National Hi-Tech Research and Development Program (863) of China (Nos. 2006AA01Z198, and 2008AA01Z132), the National Natural Science Foundation of China (No. 60533040), and the National Science Fund for Distinguished Young Scholars of China (No. 60525202)

(1) New manufacturing technology. MIT Media Lab presents a conceptual car, City Car (MIT, 2004), a lightweight electric vehicle. This car employs fully integrated in-wheel electric motors and suspension systems, which are self-contained, digitally controlled, and reconfigurable. With the wireless connectivity and a Google-like information grid, drivers could use the information to navigate in a very intelligent way.

(2) Driver assistant system. Automotive manufactures implement many novel ideas in their newest series of concept cars. BMW's ConnectedDrive includes BMW Assist, BMW Online and driver assistance systems, supporting lane change warning and parking assistant (Hoch *et al*., 2007). Mercedes-Benz is developing an intelligent driver assistance system that utilizes stereo cameras and radar sensors to monitor the surroundings around the car (Benz, 2007). Volvo's CoDriver is an intelligent assistant that coordinates information, studies the traffic situation and assists the driver (Volvo, 2007). Lexus provides advanced active safety technologies on its LS-series, including an advanced pre-collision system, dynamic driving, electronic brake assistance, and park-assistance systems (Lexus, 2007).

(3) Collision avoidance system. The SAVE-IT project develops a central component that monitors the roadway, the states of the vehicle and the driver, with evaluation of the potential safety benefits (Lee *et al*., 2004). The Cybercars project addresses navigation, obstacle avoidance and platooning (Parent and Fortelle, 2005). The SAFESPOT project aims at expanding the time horizon for acquiring safety relevant information and improving precision, reliability and quality of driving (Giulio, 2007). The prevent project develops preventive safety technologies and in-vehicle systems, which sense the potential danger and take the driver's state into account (Matthias, 2006).

(4) Driver-vehicle interface. The Adaptive Integrated Driver-vehicle Interface (AIDE) project tries to maximize the efficiency and safety of advanced driver assistance systems, while minimizing the workload and distraction imposed by in-vehicle information systems (Kutila *et al*., 2007). The Communication Multimedia Unit Inside Car (COMUNICAR) project aims at designing an easy-to-use on-vehicle multimedia human-machine interface. An information manager collects the feedback information and estimates the driver's workload according to the current driving and environment situation (Bellotti *et al*., 2005).

(5) Driver behavior recognition. The driver plays an important role in a smart car. Machine learning and dynamical graphical models, such as HMM (Oliver and Pentland, 2000), Gaussian Mixture Modeling (GMM) (Miyajima *et al*., 2007) and the Bayesian network (Kumagai and Akamatsu, 2006), can be applied for modeling and recognizing driver behaviors.

(6) Communication and cooperation. The CarTALK project enables information transmitting among cars in the vicinity (Reichardt *et al*., 2002). The COM2REACT Project (2006) establishes a cooperative and multi-level transport virtual sub-center by vehicle-to-vehicle communication and vehicle-to-centre communication. The COOPERS Project (2006) provides local situation information, traffic and infrastructure status information via a dedicated infrastructure to support vehicle communication link. The COVER Project (2006) develops semantic-driven cooperative systems with the main focus on communication between the infrastructure and vehicles. The I-WAY Project (Rusconi *et al*., 2007) designs an intelligent cooperative system, which provides real-time information from other vehicles in the vicinity and roadside equipments to improve driver's responses. The WATCH-OVER Project (2006) develops a cooperative system for the prevention of road accidents involving vulnerable road users, such as motorcycles, bicycles, and pedestrians. The Cooperative Vehicle-Infrastructure Systems (CVIS) Project (2006) creates a unified technical solution allowing all vehicles and infrastructure elements to communicate with each other in a continuous and transparent way.

(7) Safety in vehicles. The Secure Vehicular Communication (SEVECOM) Project provides a full definition and implementation of security requirements for vehicular and inter-vehicular communications (Panos *et al*., 2006). Vehicular Ad-hoc Network (VANET) security also partly addresses the safety in vehicles (Magda *et al*., 2002; Hubaux *et al*., 2004; Raya and Hubaux, 2005; Bryan and Adrian, 2005), which gives the problem statement and proposes the outline of a general solution for VANET.

However, we found that most of the work listed above is not fully context-aware. Current work usually focuses on special practical technologies, such as communication, sensing and driver assistance. In addition, the reasoning of contexts for further analysis is not put enough emphasis on. These will limit smart cars to be cars with certain accessories, so different requirements will result in different smart cars. There is a lack of a common consensus and comprehensive understanding of smart cars in a holistic view.

This study attempts to build a smart car from the viewpoint of context-awareness in a bottom-up manner. We want to build a general theoretical foundation and an infrastructure framework for a smart car. All the contexts that can characterize an entity of the driving environment will be collected and defined. Reasoning will play an important role in complex situation analysis. In such a smart car, we can develop different services and applications without much modification to the current architecture.

GENERAL ARCHITECTURE

A smart car is a comprehensive integration of many different sensors, control modules, actuators, and so on (Wang, 2006). A smart car can monitor the driving environment, assess the possible risks, and take appropriate actions to avoid or reduce the risk. A general architecture of a smart car is shown in Fig.1.

(1) Traffic monitoring. A variety of scanning technologies can be used to recognize the distance between the car and other road users. Active environments sensing in- and out-car will be a general capability in the near future (Tang *et al*., 2006). Lidar-, radar- or vision-based approaches can be used to provide the positioning information. The radar and lidar sensors provide information about the relative position and relative velocity of an object. Multiple cameras are able to eliminate blind spots, recognize obstacles, and record the surroundings. Besides the sensing technology described above, the car can get traffic information from the Internet or nearby cars.

(2) Driver monitoring. Drivers represent the highest safety risk. Almost 95% of the accidents are due to human factors and in almost three-quarters of the cases human behaviour is solely to blame (Rau, 1998). Smart cars present promising potentials to assist drivers in improving their situational awareness and reducing errors. With cameras monitoring the driver’s gaze and activity, smart cars attempt to keep the driver’s attention on the road ahead. Physiological sensors can detect whether the driver is in good condition.

(3) Car monitoring. The dynamics of a car can be read from the engine, the throttle and the brake. These data will be transferred by controller area networks (CAN) to analyze whether the car functions normally.

(4) Assessment module. It determines the risk of the driving task according to the situation of the traffic, driver and car. Different levels of risks will lead to different responses, including notifying the driver through the Human Machine Interface (HMI) and taking emergency actions by car actuators.

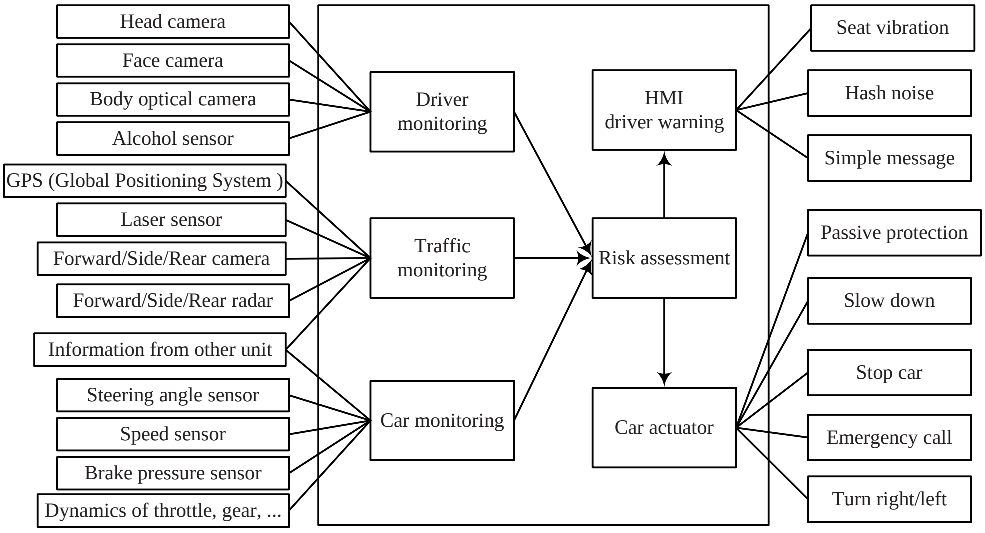


Fig.1 General architecture of a smart car

(5) HMI. It warns the driver of the potential risks in non-emergent situations. For example, a tired driver would be awakened by an acoustic alarm or vibrating seat. Visual indications should be applied in a cautious way, since a complex graph or a long text sentence will seriously impair the driver's attention and possibly cause harm.

(6) Actuators. The actuators will execute specified control on the car without the driver's commands. The smart car will adopt active measures such as stopping the car in case that the driver is unable to act properly, or applying passive protection to reduce possible harm in abrupt accidents, for example, popping up airbags.

HIERARCHICAL CONTEXT MODEL

Contexts are information collected when monitoring the roadway, the car and the driver. To implement a context-aware smart car, we must begin with the context analysis. We develop a hierarchical context model, which is the basis of representation and analysis of the smart car environment.

Hierarchical context model

We categorize context data into three layers according to the degree of abstraction and semantics: the sensor layer, the context atom layer and the context situation layer, as shown in Fig.2. The sensor layer is the source of context data, the context atom layer serves as an abstraction between the physical world and semantic world, and the context situation layer provides description of complex facts with fusion of context atoms.

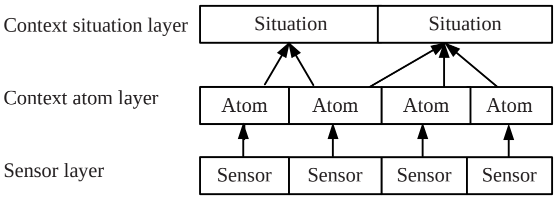


Fig.2 Three-layer context model

(1) Sensor data: we use S to denote the output set of the sensor layer.

$$S=(S_{t,1}, S_{t,2}, ..., S_{t,n}),\quad S_{t,i}=(V_t, t, q_i),$$

where $S_{t,i}$ denotes the sensor output set of sensor i at time t, V_t is the value of sensor i at time t, and q_i is the degree of credibility of the value.

(2) Context atom: we use A to denote the atom set.

$$A=(A_{t,1}, A_{t,2}, ..., A_{t,m}),\quad A_{t,i}=(\Gamma_i, t, q),$$

where $A_{t,i}$ denotes a semantic piece retrieved from sensor i at time t, called a context atom. Γ denotes an assertion retrieved from the sensor data with the assistance of ontology technology, which cannot be divided to more trivial ones.

(3) Context situation: A context situation represents the current state of an entity. We use C to denote the situation set.

$$C=(C_1, C_2, \ldots, C_s),\ C_i=(A_i, Ser_i, p_i),$$
$$A_i=(A_{t,1}, A_{t,2}, \ldots, A_{t,i}).$$

A_i denotes the set of atoms that constitute a situation. Ser_i denotes the set of services that should be performed in this situation, which may be null. p_i is the priority of a situation. We define the priority of situations for safety higher than that for entertainment.

Context atoms: ontology definition

Each sensor corresponds to a type of context atom. For each type of context atom, a descriptive name must be assigned for applications to use the contexts. We use ontology to define the name to guarantee the semantic understanding and sharing in smart cars. We use three ontologies as shown in Fig.3.

(1) Ontology for environment contexts. The environmental contexts are related to physical environments. The ontology includes the description of weather, road surface conditions, traffic information, road signs, signal lamps, network status, etc.

(2) Ontology for car contexts. The car ontology includes three parts: the power system, the security system and the comfort system. The power system concerns the engine status, the accelerograph, the power (gasoline), etc. The security system includes factors related to safety of the car and the driver, such as the status of the air bag, the safe belt, the anti-lock braking system (ABS), the reverse-aids, the navigation system, and the electronic lock. The comfort system is about entertainment devices, the air conditioner, windows, etc.

(3) Ontology for driver contexts. The driver contexts are about the driver's physiological conditions, including the heart beat, blood pressure, density of carbon dioxide, diameter of pupils, etc. The information is used to evaluate the health and mental statuses of the driver for determining whether he/she is able to continue driving.

We build the ontologies by Protégé, as shown in Fig.4.

To use the context atoms, the subscription and publication mechanisms are employed. Those applications interested in specific contexts atoms will be added to the subscriber list, along with information on how to publish context to them. Once a subscribed context changes, the new data will be delivered to the subscribing application.

Context situation: training and recognition

The context atom layer mentioned above provides the elementary conceptual data pieces. However, it is unable to represent complex knowledge, such as the current state of the car. Thus, a situation layer is expanded on the top of the atom layer. The main purpose of the situation layer is to fuse individual context atoms into meaningful context situations, i.e., recognition of context situations from context atoms. We can denote the process as

$$A_{t_1,i_1} \otimes A_{t_2,i_2} \otimes \ldots \otimes A_{t_m,i_m} \rightarrow C_i,\ \ A_{t_k,i_k} \in A,\ \ C_i \in C.$$

There are two kinds of situations in the situation layer: trigger and non-trigger situations. Trigger

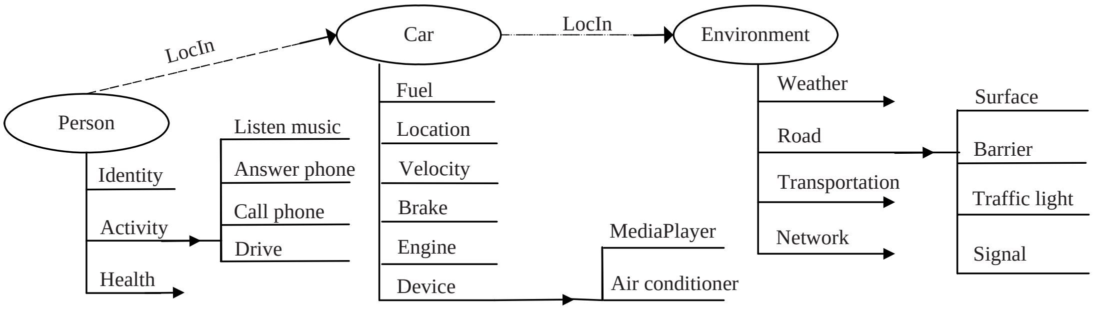
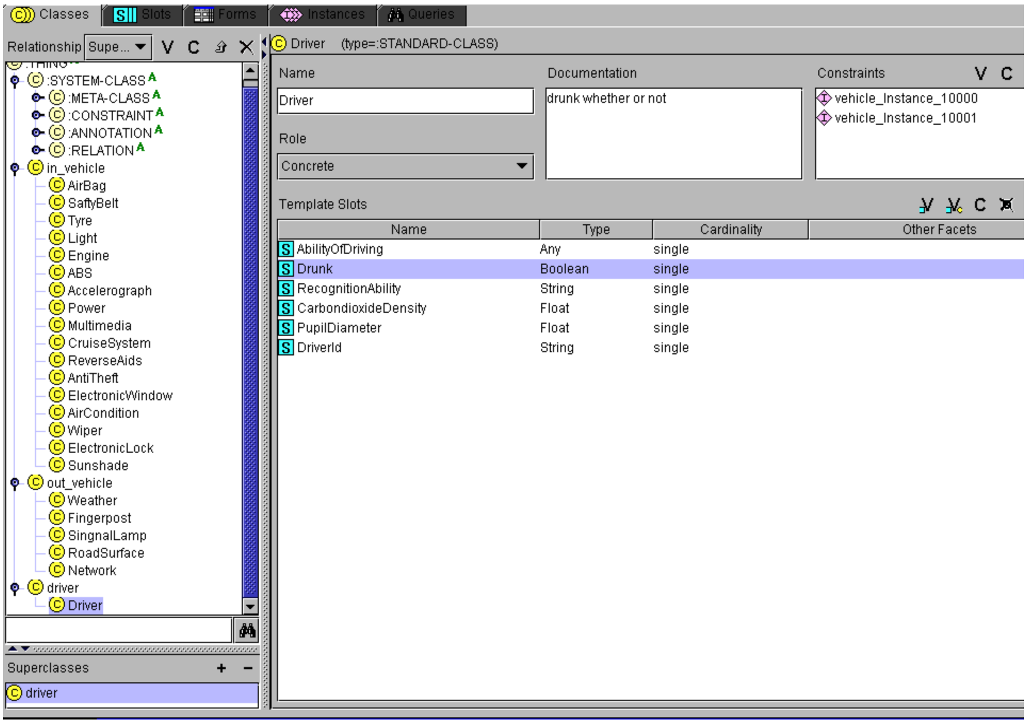


Fig.3 Three ontologies for context atoms



Driver (type=:STANDARD-CLASS)

Name: Driver

Documentation: drunk whether or not

Role: Concrete

Constraints: vehicle_Instance_10000, vehicle_Instance_10001

Template Slots

Name	Type	Cardinality	Other Facets
AbilityOfDriving	Any	single	
Drunk	Boolean	single	
RecognitionAbility	String	single	
CarbondioxideDensity	Float	single	
PupilDiameter	Float	single	
DriverId	String	single	

Fig.4 Smart car ontologies implemented with Protégé

situations are those that need to respond to and invoke specific services. Non-trigger situations do not need a response, and are used to describe the intermediate state transition.

Context situation recognition is a reasoning process and should be real time. This research uses a pattern-based inference engine and includes two parts: offline statistic-based situation pattern training and online situation recognition (Fig.5). The training phase is used to learn the statistical relationship between context atoms and situations and hence to generate the pattern of every single situation. The online recognition phase is used to recognize the current situation according to its pattern in the running time of a smart car.

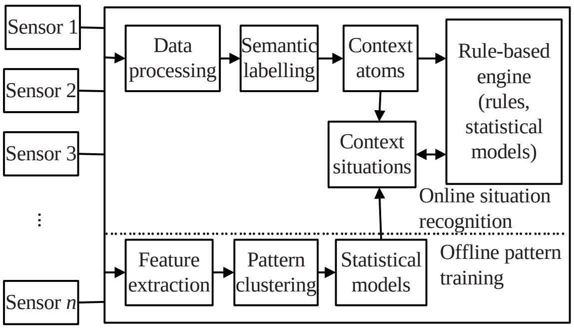


Fig.5 Flowchart of the context situation inference engine

1. Situation training

We should train situation patterns before we can recognize them. The situation training model is a grid of $M\times M$ nodes, with a vector of atom sets $\boldsymbol{A}_k(t)$ and a weight vector $\boldsymbol{X}_k(t)$ associated with node k (k=1, 2, …, m), where $\boldsymbol{A}_k(t)=\{a_1^k(t), a_2^k(t), ..., a_n^k(t)\}$ and $a_j^k(t)=\ \{a_{j,1}^k, a_{j,2}^k, ..., a_{j,n}^k\}$ (n=n(k, j, t)). $a_{j,i}^k$ is the ith value from context source j at time t of node k. $X_k(t)=\{x_1^k(t), x_2^k(t), ..., x_n^k(t)\}$ and $x_j^k(t)=$ $\{x_{j,1}^k, x_{j,2}^k, ..., x_{j,n}^k\}$ (n=n(k, j, t)), $x_{j,i}^k\in[0,1]$. We denote the input training vector at time t as $\bar{\boldsymbol{A}}_k(t)=\{\bar{a}_1(t), \bar{a}_2(t), ..., \bar{a}_n(t)\}$. The similarity functions are defined as

$$G_k(t)=\left(\sum_{i=1}^{\bar{n}(t)}\sum_{j=1}^{n(k,t)}\bar{\delta}_{k,i}(t)\delta_{k,i}(t)\right)^2\Bigg/n_{k,i}(t)\bar{n}(t),$$

$$\delta_{k,i}(t)=\begin{cases}1, a_j(t)\in a_j^k(t),\\0, a_j(t)\notin a_j^k(t).\end{cases}$$

$\bar{\delta}_{k,i}(t)$ is defined in the same way. A node with the highest similarity will be chosen: $v(t)=\arg\max\limits_{1\le k\le M^2} G_k(t)$. The update algorithm of the weight vector is described below:

For each node k=1, 2, …, M^2

(a) If $a_{j,i}^k(t)\in\bar{A}(t)$, then

$x_{j,i}^k(t)=x_{j,i}^k(t)+\alpha(t)h_m(dl)(1.0-x_{j,i}^k(t))$;

(b) If $a_{j,i}^k(t)\notin\bar{A}(t)$, then

$x_{j,i}^k(t)=x_{j,i}^k(t)(1.0-x_{j,i}^k(t))\,|\,h_m(dl)$;

(c) If $a_{j,i}^k(t)\notin A_j^k(t)$ and $h_m(dl)>0$, then

$n(k,j,t)=n(k,j,t)+1, a_{j,n(k,j,t)}^k(t)=\bar{a}_j(t)$, ,

$x_{j,n(k,j,t)}^k(t)=\alpha(t)h_m(dl)$;

(d) If $x_j(t)<\beta(t)$, then $n(k,j,t)=n(k,j,t)-1$,

where $h_m(i)=(1-a\times b\times i^2)\exp(-a\times i^2)$ and dl is the Euclidean distance between the winner node and the node k.

The training procedure orders the inputs by assigning map-units to each kind of input, and the resulting map will be topologically ordered, i.e., similar inputs activate neighboring units. After a few iterations, neurons start to organize themselves in a structured, topological way: different inputs activate different neurons. A cluster of approximate atom sets corresponds to a situation.

2. Situation recognition

We define 13 trigger situations in the scenario, as shown in Fig.6, including five driver situations, five car situations and three transportation situations. Collisions-1, -2, and -3 are risk assessments and mean the danger levels. Since assessments apply different features and have different targets, we train three statistical models and use the corresponding classifiers to estimate the three kinds of situations.

At any given time, the classifiers will recognize the current situation if the situation stays unchanged for a continuous period of time. Only those transitions that last for a period of time will be recognized as the transitions that really happen, while the temporary transitions will be ignored.

It is possible that a cluster has not been given a label yet. In such a case, a distance-weighted K-nearest neighbor (KNN) algorithm is used for

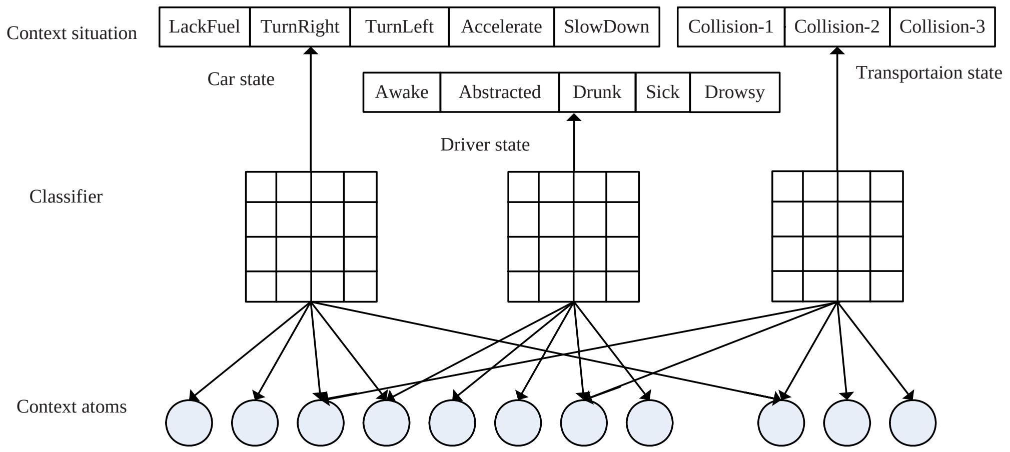


Fig.6 Context situation recognition architecture

searching the cluster map. A voting among the *k* closest labels on the map results in the most probable label.

SMART CAR PROTOTYPE

We integrate devices, sensors and network to build a smart car prototype, as shown in Fig.7. A software platform is developed to manage devices, support network communication and provide a runtime environment for the context model and applications.

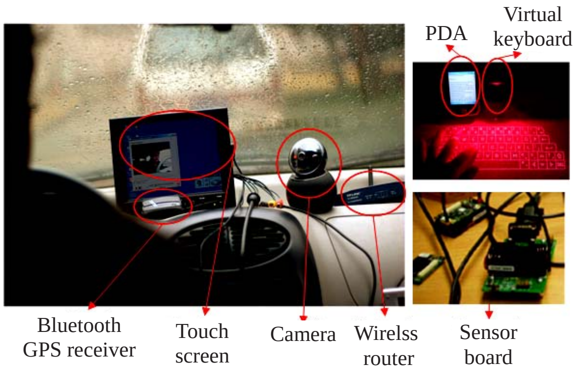


Fig.7 Smart car prototype

Hardware infrastructure

The smart car includes intelligent devices, sensors, and processors named electronic control units (ECU). The devices used in the prototype are listed in Table 1. A notebook PC ran the software platform and applications. A PDA conducted the cooperation with the software platform in the smart car. The virtual keyboard and the touch screen were used to provide the passengers with easier interaction. For communication, a wireless router was adopted to provide users with information services in the car. Three ECUs were built using an MPC555 processor. A CAN hub was used to combine several communication nodes in a car. A CAN analyzer was the necessary analysis tool for messaging.

Table 1 Device list in the smart car prototype

Device	Type
Notebook PC	SONY VAIO FJ
ECU	Motorola MPC555, Atmel AT89C51
PDA	400 MHz Intel, XScal-PXA255
Wireless router	D-Link DIR-300
CAN analyzer	CANalsyt-II
CAN connection	CANHub-S5
Virtual keyboard	I-tech virtual laser keyboard
Touch screen	FA801-NP/C/T
Camera	Logitech QuickCam Orbit

Various sensors were deployed in our prototype, as listed in Table 2. The environment contexts were acquired by a Crossbow sensor board, which is combined with a wireless module. It provides sensing capabilities including ambient light, barometric pressure, magnetic field, photo-sensitive light, humidity and temperature. Ultrasound sensors were used to estimate the distance between two adjacent cars.

The car contexts such as information about fuel consumption, wheel rotation and velocity were obtained from the car CAN bus. A Bluetooth GPS receiver was used to determine the location of the car.

Table 2 Sensor list in the smart car prototype

	Sensor	Context
Environment	Thermometer	Temperature
	Hygrometer	Humidity
	Microphone	Loudness
	Ambient light sensor	Light
	IR sensors	Car following
Car	Hall-effect sensor	Velocity
	Hall-effect sensor	Wheel rotation
	Fuel level sensor	Fuel consumption
	Bluetooth GPS receiver	Car-location
	Accelerometer	Acceleration
Driver	Camera	Head, gaze, identity
	Microphone	Vocal command
	RFID (radio frequency identification)	Identity
	Hand pressure sensor	Grip force
	Alcohol sensor	Alcohol density

As for driver contexts, RFIDs were used to recognize the identity of the driver and passengers. A hand pressure sensor was used to determine the finger force on the steering wheel. An alcohol sensor was used to detect the alcohol level of the driver.

For complex sensors such as cameras and microphones, complicated processing is required after the sensor data are received.

Cameras were installed to track the movement of eyes and head of the driver. Real-time face detecting and eye tracking were carried out by fusion of active shape model (ASM) (Cootes *et al.*, 2001) and cascaded Adaboost (Viola and Jones, 2004). Eye blink behaviours were detected using the method described in (Pan *et al.*, 2007). For image-based user recognition, we used the Fisherface algorithm (Belhumeur *et al.*, 1997) to compare the detected faces with the registered users' face images.

A microphone was used to receive the vocal command from a driver. About 20 keywords, including "weather", "music on", "music off", "parking space", and so on, were defined for a driver to access services by voice rather than by hand. The SONAR (Liu, 2007), a speech processing and speaker recognition toolkit developed by our lab, was used to build the keyword recognition engine.

Software platform

We have developed a context-aware software platform for the smart car, as shown in Fig.8. The platform includes four layers: network layer, broker layer, context infrastructure, and services layer.

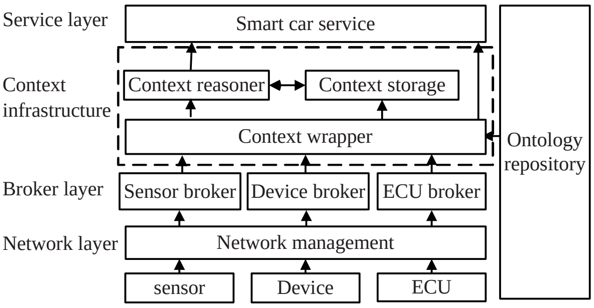


Fig.8 Software platform for the smart car

1. Network layer

The smart car supports different communication approaches. A ZigBee wireless sensor network connects all mini-sensors. The smart vehicle network is a serial-bus system for the communication between mechanical nodes (such as the engine and the steering system). The WLAN 802.11a/b/g network supports the communication between digital devices. The CDMA 1xRTT network is responsible for wide-area communication.

2. Broker layer

The sensor broker is responsible for discovery and registration of new sensors added into the smart car. One broker manages one category of sensors. Sensors can transmit data via WLAN, serial port, Ethernet, and USB. The broker will assign a globally unique address or identity to a sensor, specify the updating frequency, and define the way for the sensor to transmit data and for the system to parse data. The device broker is responsible for discovering new devices and registering them for cooperation in the smart car. The ECU broker aims at managing processors and collecting specific contexts such as spare memories.

3. Context infrastructure

The context infrastructure has been implemented on the basis of a context toolkit (Daniel *et al.*, 1999) and consists of three parts: (1) the context wrapper, which transforms sensor data into semantic context atoms; (2) the context reasoner, which trains and recognizes context situations by aggregating various types of context atoms; (3) the context storage, which is a repository for historical contexts and provides the advanced query services.

4. Service layer

Smart cars intend to create a safer, more efficient and more convenient driving environment for drivers, so specific services should be developed. In a smart car, most services, such as slowing down when the distance to the front adjacent car is less than the safety limit, need to transfer signals via CAN to control a certain actuator. Each actuator is managed by an ECU. In order to execute the service, a message including the service Application Programming Interface (API) and parameters should be sent to the ECU that manages the actuator. The ECU parses the message and sends control signals to a relay, which will control the actuator to change its state. We have implemented turning on and off the lamp according to daylight, as shown in Fig.9.

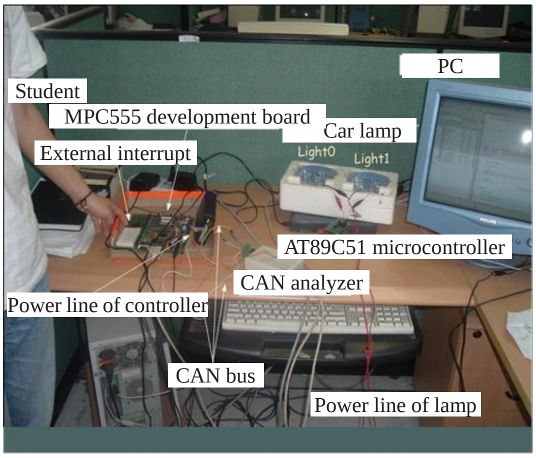


Fig.9 Using CAN to control the car lamp

PERFORMANCE EVALUATION

Twenty volunteers, each for 1.5 h, were invited to drive the smart car at an average speed of 60 km/h. For studying the impact of the passenger on the driver, two persons (one driver and one passenger) were arranged to sit in the same car. In each test session, they may chat, talk on the phone, look around, drink, and eat, which cover the most common activities in a car. This scenario was repeated 15 times. Then one person alone repeated 15 times again. Two performance metrics were evaluated: accuracy of the context situation recognition and efficiency of the smart car.

Experiment 1: Evaluation of accuracy of context situation recognition

We defined 13 trigger situations of the smart car, including five driver situations, five car situations and three transportation situations. The driver situation is the most complex one since it deals with face image processing. So we will use driver recognition as the worst-case evaluation.

From Fig.10 we can see that the feature dimension of context situations would impact the accuracy. Appropriate features will contribute to reducing the error probability. Naturally, the most useful features are those that distinguish one situation from another, especially for situations that are most frequently confused. An obvious way to reduce the error rate is to introduce new and independent features providing additional information. Although increasing the number of features will increase the computational cost of both the feature extraction and the classification, it is often reasonable to believe that the performance will improve.

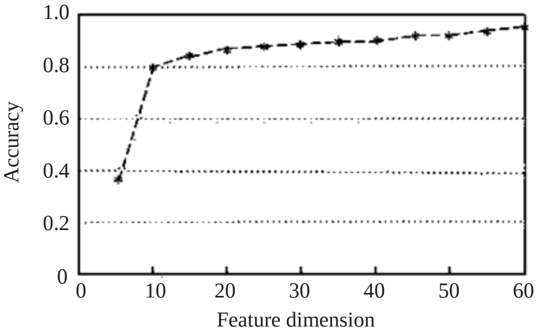


Fig.10 Average accuracy of context situation recognition

Experiment 2: Evaluation of efficiency of the smart car

We used the time consumption for a context-aware application to assess efficiency of the smart car, e.g., the aggregate time for the processing sequence of context sensing, processing, situation clustering, and service invoking. The response time equation is denoted as

$$T=T_s+T_{wap}+T_{recg}+T_{parse}+T_{net},$$

where T_s is the delay time for conveying the data from the sensor to the context infrastructure, T_{wap} is the delay time for context atom management, including mapping sensor data into semantic atoms and publishing them to those subscribing the context, T_{recg} is the delay time to match the current context atoms into a situation pattern, T_{parse} is the delay time to parse the situation profile to find the appropriate service of the situation, and T_{net} is the delay time for conveying the service message to a certain actuator.

Table 3 shows the response time performance. The delay appears close in each session, so the averages are listed for evaluation.

Table 3 Response time in the smart car (unit: s)

	T_{s}	T_{wap}	T_{recg}	T_{parse}	T_{net}	Total
One person	0.1	0.2	0.8	0.08	0.2	1.4
Two persons	0.1	0.4	0.8	0.09	0.2	1.6

Context situation recognition is a computationally intensive task. However, it is still feasible for non-time-critical applications, so the efficiency (with a delay of nearly 1.5 s) is acceptable. For time-critical applications such as collision avoidance systems and navigating systems, we need to improve the sensing technology and the analysis method of situation recognition.

CONCLUSION

As a promising application domain of ubiquitous computing, smart cars draw more and more attention. This research attempts to build a smart car from the view of context-awareness. Our contributions are threefold: (1) a general architecture of a smart car is set up; (2) a three-layer context model is proposed, which can represent a complex driving environment; (3) a smart car prototype, including software and hardware infrastructures, is built to implement the capability of context-awareness.

Our future work includes applying more sophisticated sensing technologies to detect the physiological and psychological statuses of the driver to enhance the smart car prototype. Knowledge-based inference approaches will be developed and inserted into the risk assessment module for more reliable decision making. More considerations and efforts will be made for driver intention prediction.

References

Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J., 1997. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, **19**(7):711-720. [doi:10.1109/34.598228]

Bellotti, F., de Gloria, A., Montanari, R., Dosio, N., Morreale, D., 2005. COMUNICAR: designing a multimedia, context-aware human-machine interface for cars. *Cogn. Technol. Work*, **7**(1):36-45. [doi:10.1007/s10111-004-0168-9]

Benz, 2007. Mercedes-Benz to Present Driver Assistance Systems of Tomorrow at the ITS World Congress 2008. Available from: http://e-carzone.com/news/auto_5911.shtml.

Bryan, P., Adrian, P., 2005. Challenges in Securing Vehicular Networks. Proc. HotNets-IV, 2005.

COM2REACT Project, 2006. COM2REACT: V2V Communication for Cooperative Local Traffic Management. Available from: http://hal.archives-ouvertes.fr/docs/00/18/00/49/PDF/ITSWC2007-COM2REACT-v2.pdf.

COOPERS Project, 2006. Cooperative Systems for Intelligent Road Safety. Available from: http://www.first.fraunhofer.de/owx_download/projektblatt_coopers_082008_screen_en.pdf.

Cootes, T.F., Edwards, G.J., Taylor, C.J., 2001. Active appearance models. *IEEE Trans. Med. Imag.*, **23**(6):681-685. [doi:10.1109/34.927467]

COVER Project, 2006. Semantic Driven Cooperative Vehicle, Infrastructure Systems for Advanced e-Safety Applications. Available from: http://www.esafetysupport.org/en/esafety_activities/related_projects/research_and_development/cover.htm.

CVIS Project, 2006. CVIS-Cooperative Vehicle-Infrastructure Systems. Available from: http://www.esafetysupport.org/en/esafety_activities/related_projects/research_and_development/cvis.htm.

Daniel, S., Anind, K.D., Gregory, D.A., 1999. The Context Toolkit: Aiding the Development of Context-enabled Applications. Proc. SIGCHI Conf. on Human Factors in Computing Systems, p.434-441. [doi:10.1145/302979.303126]

Giulio, V., 2007. The SAFESPOT integrated project: an overview. *IEEE Intelligent Vehicles Symp.*, p.14. [doi:10.1109/IVS.2007.4290069]

Hoch, S., Althoff, F., Rigoll, G., 2007. The Connected Drive Context Server-flexible Software Architecture for a Context Aware Vehicle. Advanced Microsystems for Automotive Applications, Springer, Berlin, Heidelberg. [doi:10.1007/978-3-540-71325-8]

Hubaux, J.P., Capkun, S., Jun, L., 2004. The security and privacy of smart vehicles. *IEEE Secur. Priv. Mag.*, **2**(3):49-55. [doi:10.1109/MSP.2004.26]

Kumagai, T., Akamatsu, M., 2006. Prediction of human driving behavior using dynamic bayesian networks. *IEICE-Trans. Inf. Syst.*, **E89-D**(2):857-860. [doi:10.1093/ietisy/e89-d.2.857]

Kutila, M.H., Jokela, M., Kinen, T.M., Viitanen, J., Markkula, G., Victor, T.W., 2007. Driver cognitive distraction detection: feature estimation and implementation. *Proc. IMechE J. Autom. Eng.*, **221**(9):1027-1040. [doi:10.1243/09544070JAUTO332]

Küçükay, F., Bergholz, J., 2004. Driver Assistant Systems. Int. Conf. on Automotive Technologies. Istanbul, Turkey.

Lee, J., Reyers, M., Smyser, T., Liang, Y., Thornburg, K., 2004. Safety Vehicles Using Adaptive Interface Technology (Task Final Report: Phase 1 of the SAVE-IT project). Available from: www.volpe.dot.gov/opsad/saveit.

Lexus, 2007. Lexus Advanced Active Safety Technologies. Available from: http://www.lexushelp.com/models/2008/LSh/Description.htm.

Liu, Y.Y., 2007. Sonar 2.0: The Speaker Recognition Software Platform. MS Thesis, Zhejiang University, China (in Chinese).

Magda, E.Z., Sharad, M., Gene, T., Nalini, V., 2002. Security Issues in a Future Vehicular Network. Proc. European Wireless.

Matthias, S., 2006. Contribution of PReVENT to the Safe Cars of the Future. Presentation in Special Session of 13th ITS World Congress, London, UK.

MIT, 2004. City Car. Available from: http://cities.media.mit.edu/projects/citycar.html.

Miyajima, C., Nishiwaki, Y., Ozawa, K., Wakita, T., Itou, K., Takeda, K., Itakura, F., 2007. Driver modeling based on driving behavior and its evaluation in driver identification. *Proc. IEEE*, **95**(2):427-437. [doi:10.1109/JPROC.2006.888405]

Moite, S., 1992. How Smart Can a Car Be. Proc. Intell. Vehicles Symp., p.277-279. [doi:10.1109/IVS.1992.252271]

Oliver, N., Pentland, A.P., 2000. Graphical Models for Driver Behavior Recognition in a Smart Car. Proc. lEEE Intell. Vehicles Symp., p.7-12. [doi:10.1109/IVS.2000.898310]

Pan, G., Sun, L., Wu, Z.H., Lao, S.H., 2007. Eyeblink-based Anti-spoofing in face recognition from a Generic Webcamera. 11th IEEE Int. Conf. on Computer Vision, Rio de Janeiro, Brazil, p.1-8. [doi:10.1109/ICCV.2007.4409068]

Panos, P., Virgil, G., Jean-Pierre, H., 2006. Securing Vehicular Communications—Assumptions, Requirements, and Principles. 4th Workshop on Embedded Security in Cars, Berlin, Germany.

Parent, M., Fortelle, A.D.L., 2005. Cybercars: Past, Present and Future of the Technology Parent. ITS World Congress.

Rau, P.S., 1998. A Heavy Vehicle Drowsy Driver Detection and Warning System: Scientific Issues and Technical Challenges. Proc. 16th Int. Technical Conf. on the Enhanced Safety of Vehicles, Ontario, Canada.

Raya, M., Hubaux, J.P., 2005. The Security of Vehicular Ad-hoc Networks. Proc. SASN, p.11-21. [doi:10.1145/1102219.1102223]

Reichardt, D., Miglietta, M., Moretti, L., Morsink, P., Schulz, W., 2002. CarTALK 2000: safe and comfortable driving based upon inter-vehicle-communication. *IEEE Intell. Vehicle Symp.*, **2**(17-21):545-550.

Rusconi, G., Brugnoli, M.C., Dosso, P., Kretzschmar, K., Bougia, P., Fotiadis, D.I., Salgado, L., Jaureguizar, F., de Feo, M., 2007. I-WAY, intelligent co-operative system for road safety. *IEEE Intell. Vehicles Symp.*. **13**(15):1056-1061. [doi:10.1109/IVS.2007.4290256]

Tang, S.M., Wang, F.Y., Miao, Q.H., 2006. ITSC 05: current issues and research trends. *IEEE Intell. Syst.*, **21**(2):96-102. [doi:10.1109/MIS.2006.31]

Viola, P., Jones, M.J., 2004. Robust real-time face detection. *Int. J. Comput. Vis.*, **57**(2):137-154. [doi:10.1023/B:VISI.0000013087.49260.fb]

Volvo, 2007. Volvo Cars Focuses on Preventive Safety. Available from: http://www.volvocars.com/intl/corporation/NewsEvents/News/Pages/default.aspx?item=33

Wang, F.Y., 2006. Driving into the future with ITS. *IEEE Intell. Syst.*, **21**(3):94-95. [doi:10.1109/MIS.2006.45]

Wang, F.Y., Zeng, D., Yang, L.Q., 2006. Smart cars on smart roads: an IEEE intelligent transportation systems society update. *IEEE Perv. Comput.*, **5**(4):68-69. [doi:10.1109/MPRV.2006.84]

WATCH-OVER Project, 2006. WATCH-OVER—Vehicle-to-vulnerable road user cooperative communication and sensing technologies to improve transport safety. Available from: http://www.esafetysupport.org/en/esafety_activities/related_projects/research_and_development/watch-over.htm