2012

A cloud-based design of smart car information services

Yi Xu
University of Wollongong, yx661@uowmail.edu.au

Jun Yan
University of Wollongong, jyan@uow.edu.au

Publication Details

Xu, Y. & Yan, J. (2012). A cloud-based design of smart car information services. Journal of Internet Technology, 13 (2), 317-326.

A cloud-based design of smart car information services

Abstract

"Although modern Smart Car technologies are growing rapidly, there are many challenges to be addressed. The limited processing capabilities of the contemporary in-car computers in smart cars may cause bottlenecks in smart control. The content in smart car applications can hardly be extended. In addition, the traditional software installation approach that is used in smart cars is neither economical nor convenient. To address these issues in smart cars, this paper presents a Cloud-based Information Integration Platform that has the ability to improve flexibility and enhance the value of smart cars. In this work, the architecture of the Cloud-based Information Integration Platform was designed based on new technologies such as Cloud computing, Web Services, and Controller Area Network Applications. This service-oriented solution has a vertically integrated architecture that uses full information gathering and service driven synergies. To achieve the smart control and information sharing in smart cars, we also designed a Smart Car Information Service as the user interface to implement smart car applications through customized business processes. This fills the information processing gap between the smart car and cloud computing. The proposed platform can be used for different usage scenarios, such as car diagnosis, car repair facilitation, etc."

Keywords

design, smart, car, information, services, cloud

Disciplines

Physical Sciences and Mathematics

Publication Details

Xu, Y. & Yan, J. (2012). A cloud-based design of smart car information services. Journal of Internet Technology, 13 (2), 317-326.

A Cloud-based Design of Smart Car Information Services

Yi Xu, Jun Yan
School of Information Systems and Technology (SISAT)
University of Wollongong
AUSTRALIA
yx661@uowmail.edu.au; jyan@uow.edu.au

Abstract

Although modern Smart Car technologies are growing rapidly, there are many challenges to be addressed. The limited processing capabilities of the contemporary in-car computers in smart cars may cause bottlenecks in smart control. The content in smart car applications can hardly be extended. In addition, the traditional software installation approach that is used in smart cars is neither economical nor convenient. To address these issues in smart cars, this paper presents a Cloud-based Information Integration Platform that has the ability to improve flexibility and enhance the value of smart cars. In this work, the architecture of the Cloud-based Information Integration Platform was designed based on new technologies such as Cloud computing, Web Services, and Controller Area Network Applications. This service-oriented solution has a vertically integrated architecture that uses full information gathering and service driven synergies. To achieve the smart control and information sharing in smart cars, we also designed a Smart Car Information Service as the user interface to implement smart car applications through customized business processes. This fills the information processing gap between the smart car and cloud computing. The proposed platform can be used for different usage scenarios, such as car diagnosis, car repair facilitation, etc.

Keywords: CAN bus, Cloud Computing, Web Services, Smart Cars

1 Introduction

Challenges of urban traffic mainly come from three points: energy consumption, emissions, safety and congestion. In response to these challenges, the research related to automotive technology has changed from the traditional internal combustion engine to electricity and hydrogen for the electrification energy and intelligence research. The characteristic point of green cars is low-emission (zero emission). "Green" and "Smart" are complementary with common development. The cutting-edge technology has just arrived to help us reduce the fuel consumption, resources and time.

The smart car is the future trend in automotive improvement. The development of automotive electronic technology and the technology of smart car is evolving rapidly. The use of new technologies in the smart car has been a focus of study in recent years.

Currently, a smart car mainly uses the Controller Area Network (CAN) bus as the network connection in the body. A recent surge of research on CAN in Auto (CiA) has given us new opportunities and challenges. The focus of car design has shifted from an emphasis on transportation to safety and comfort. In parallel, cloud computing is getting considerable attention, not only from the computer science community but also from the general public.

On the one hand, the demand for in-car computer resources also increases. The traditional in-car computers have limited processing capabilities to support smart car functions, such as navigation and display of real time information. It is not smart enough without information sharing among the real world and the car computers. Little research has been done on topic development in cloud-based services for smart cars. In other words, little is known about the conceptual framework and working methods of smart cars' applications. Only few isolated efforts have continued to address the single PC applications for smart cars. While considerable attention has been paid in the past to research issues related to GPS and other car PC functions, literature on issues of services for smart cars has emerged very slowly [2]. The use of cloud computing in the car as a field of research has had limited exploration.

The aim of this research is to design a new information integration platform for smart cars to enhance the ability of processing the real-time information from the smart cars. This enhanced ability will allow for the development and deployment of diverse smart car applications with little effort, thus offering the smart car drivers benefits of safety and an enjoyable driving experience. In this paper, we present a Cloud-based Information Integration Platform for Smart Cars as a new technology that enables cloud services to seamlessly integrate with the smart car environment. By adopting the Cloud-based Information Integration Platform, drivers are expected to be able to use their favorite smart car applications and services anytime, anywhere.

The rest of the paper is organized as follows. The second section reviews major technologies that are related to this research. In Section 3, we describe the architecture of the proposed platform in details. Then we present a use case scenario as a sample application to demonstrate our designed platform in Section 4. Finally, Section 5 concludes this paper.

2 Related Technologies

The proposed Cloud based information integration platform for smart cars provides a novel approach to connecting the in-car system with external systems. This can lead to the development of new smart car applications. The platform is developed based on the technologies and components as follows.

2.1 Cloud Computing

Recent years have seen increased attention being given to the Cloud Computing. Cloud Computing is getting considerable attention not only from academia and industry but also from people in general public.

Currently, we have some public utility computing include Amazon Elastic Compute Cloud (EC2), Google AppEngine, and Microsoft Azure. Amazon's EC2 service (www.amazon.com/ec2) is a good example of the cloud computing. Amazon Elastic Compute Cloud is a web service that provides re-sizable computer capacity in the cloud, which was designed to make web-scale computing easier for developers [1, 2]. Amazon EC2 allows us to pay only for capacity that we actually use. It changes the economics of cloud computing.

2.2 Software as a Service (SaaS)

The use of external software on the Internet servers is called Software as a Service (SAAS) [3, 4]. SaaS changed the way that software is delivered to the customers.

The advantages of SaaS for both end users and service providers are well understood. Service providers could simplify software installation and maintenance and have centralized control over versioning. End users can have "anytime - anywhere" access to the service, share data and act as a team more easily, and keep their data stored safely in the cloud. With SaaS, cloud-based services platforms can provide applications to users for use as services on demand, either through a time subscription or a "pay-as-you-go" model. In other words, SaaS is a model of software deployment over the Internet. Instead of purchasing the hardware and software to run an application, customers need only a computer with Internet access to run the software[5, 6].

2.3 Controller Area Network (CAN) Applications

Controller Area Network (CAN) is a serial communication bus designed to provide simple, efficient and robust communications for in-car networks [7-9]. It is a serial bus system especially suited for networking "intelligent" devices as well as actuators within a system or sub-system [10, 11].

CAN bus in a smart car provides an information sharing platform for all kinds of electronic control systems. Information sharing is the basic function of CAN bus network. CAN application layer (CAL) is an application-independent application layer that has been specified and is now maintained by the international users and manufacturers group CAN in Automation (CiA) [10]. CANopen is an implementation of CAL and is defined by the CANopen Communications Profile in CiA DS-301 [10]. In CANopen networks, all stations are equal and data exchange is organized directly between devices. Moreover, CANopen can be implemented even on devices with low computing power and storage capacity.

In early 1990's, the automotive industry published the method used for diagnosing vehicle errors electronically. This is embodied in the On-Board Diagnostics (OBD) standard [12]. Originally, the OBD system was implemented to help cars control their emissions. However, the potential of the system was quickly realized and now OBD is the hub of a car's electrical control system [13]. The second generation of the OBD standard, OBD-II, was introduced in 1995 [12].

2.4 In-Car Computer and Wireless Communication

The in-car computer is the core of a smart car. Traditionally, it is a mini PC or a SCM (Single-Chip Microcomputer) [14, 15]. Currently, an in-car computer is commonly equipped with small Global Positioning System (GPS) devices. The navigation system in a car provides route planning, and even voice guidance. By using GPS, users could get their precise position in the world twenty-four hours a day in three dimensions and precise time traceable to global time standards.

3 Cloud-based Information Integration Platform for Smart Cars

We present the novel and unique platform to deal with information sharing from a smart car. As shown in Figure 1, all the car applications will be published as services in the cloud. Remote control, as another access approach for using smart cars applications was provided in this platform.

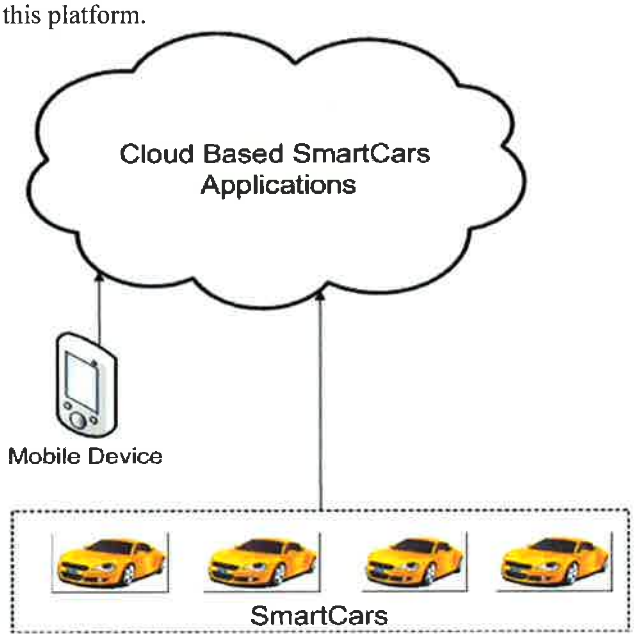

Figure 1 Overall view of Cloud-based Platform for Smart Cars

3.1 Architecture of Cloud-based Information Integration Platform

The Cloud-based Information Integration Platform for Smart Cars has two different layers, namely the hardware layer and the services & software layer. Figure 2 shows the architecture of our designed platform.

From the perspective of the network topology, the entire communication system is organized by the in-car computer. GPS, wireless, gateway controller, CAN bus and automotive electrical components are controlled by the in-car computer in a smart car.

The hardware layer in this architecture acts as the basis of the Cloud-based Information Integration Platform for Smart Cars. This layer includes the CAN bus sub-system and an in-car computer. The CAN bus sub-system has frequent internal communication, thus requiring high-demanding real-time control. Most of the car electronics belong to the CAN bus sub-system. In a smart car, an in-car computer will collect messages from sensors via CAN bus. The gateway controller which is between CAN bus and in-car computer can convert CAN bus messages, allowing the in-car computer to process the CAN bus messages in the IT data bus. By using the CAN bus, in-car computer and gateway controller, the hardware layer can provide a versatile, low-cost hardware as a base in the information integration platform architecture.

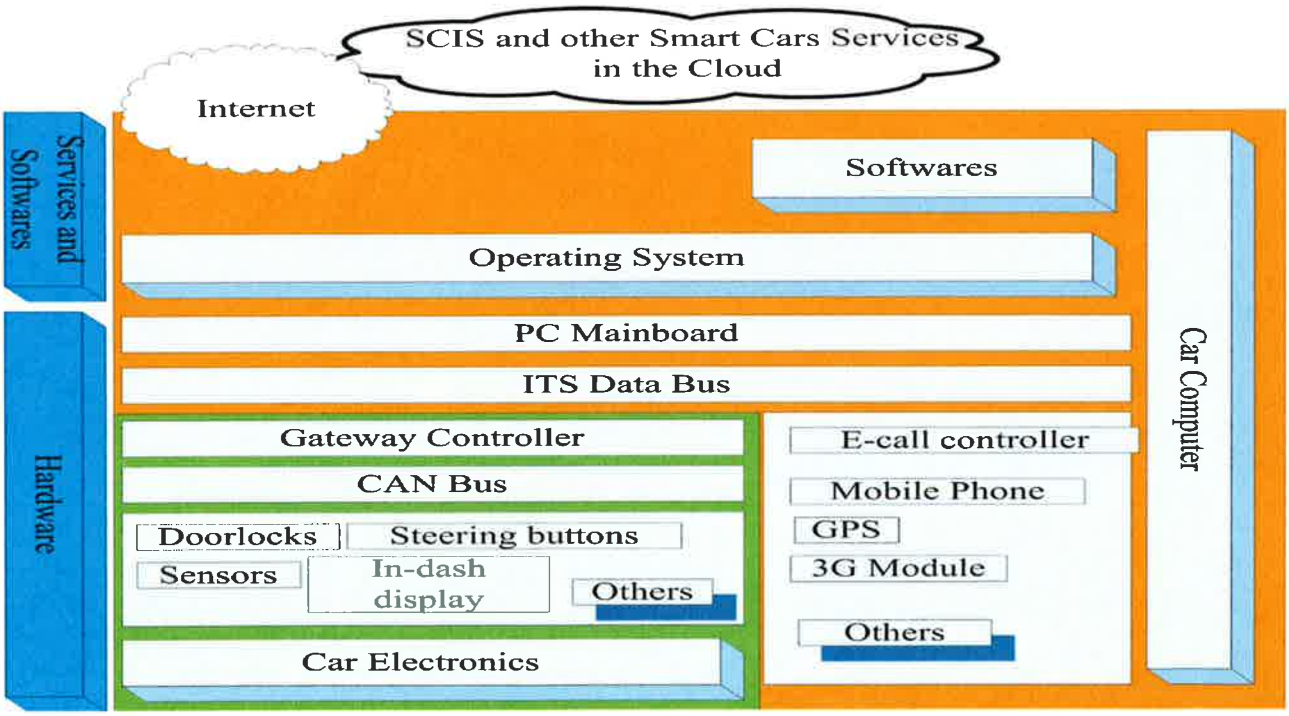

Figure 2 Architecture of Cloud-based Car Information Integrated Platform

The CAN-bus messages base or extended frame formats will be determined by the use of a higher-layer designed protocol. In our designed platform, the gateway controller provides one procedure to deal with standard CAN bus messages. Normally, different car companies have their own standards, and they may use different format messages in CAN bus. It is possible to process the multifarious CAN bus messages from different companies by using different procedures in the gateway controller.

The gateway controller is positioned between the in-car computer and the CAN bus in a smart car. It is responsible for message conversion between the CAN bus and the IT data bus. The main function of the gateway controller is to analyze all kinds of messages from the CAN bus and convert them into another format which can be dealt with in the IT data bus. The gateway controller can also issue a directive to CAN, coordinating the work of various ECU's.

Using the gateway controller, messages from each node in the CAN bus can be read or written by the in-car computer. This process is a two-way transmission in CAN bus which provides an information sharing platform for all kinds of electronic control systems to share information [16].

Currently, most cars use two different CAN buses in network connection, a high-speed CAN bus and a low-speed CAN bus. The main mission of the high-speed CAN bus is to connect the engine controller (ECU), ASR and ABS controllers, airbag controllers with those instrument clusters that have the same basic features. The high-speed CAN bus in a car could run more real-time-critical functions to enhance the safety, comfort, and added-value of the passenger car and improve its service quality and performance [7, 16]. For less speed demands, the low-speed CAN bus is used to run a few small wirings that are connected together to form several sub-systems. These sub-systems use the gateway to share information and coordinate ECUs. The low speed CAN system deals with the connection with central locking, power windows, mirrors and lighting, etc. [9]. The speed of a high-speed CAN bus in a drive system can reach 500kbps, while the speed of the low-speed CAN bus in an electronics system is 100 kpbs [16, 17].

For processing the information in smart cars, the in-car computer will collect information from the gateway controller and deliver those CAN bus messages to the smart car information service (SCIS) in the cloud. It is a bridge between a smart car platform and the cloud-based information service for smart cars.

Global Positioning System (GPS) and wireless communication technology play an important role in the in-car computer. The communication device in the in-car computer has the ability of accessing 3.5G and 3.75G networks, can also provide mobile networking access of several Mbit/s to laptop computers and smart phones.

The software and services layer is the higher layer in the architecture of Cloud-based Information Integration Platform for Smart Cars. The designed platform supports the development of various smart car applications in multiple categories through the software and services layer. By using Web Services in the Cloud-based Information Integration Platform, smart car users can choose and use smart car applications as services from the cloud.

The Cloud-based Information Integration Platform allows a driver to use smart car applications as services in the cloud. It provides intelligent processing, cloud computing capability and infrastructure to users and various smart car services.

3.1 Message Exchanges and Data Synchronisation to the Cloud

We present the SCIS as the interface of the Cloud-based Information Integration Platform for Smart Cars to execute smart car applications in the cloud. In addition, there are many existing technologies in the cloud that can be modified or enhanced to support new smart car applications. As shown in Figure 3, one smart car service can act as a service consumer to use the functionality exposed by the other core smart car services. In other words, each cloud-based service for smart cars can invoke other designed in-car applications by a defined business process in the cloud.

Figure 3 Cloud Services for Smart Cars

To achieve the smart car control, we connect the IT data bus and the CAN bus via a gateway controller in the designed platform. Currently, the in-car computer uses the IT data bus to exchange messages among GPS, 3G module, Bluetooth and other in-car devices. The IT data bus requires high-data transfer rate of modules connected together. The gateway controller exchanges messages between IT data bus and CAN bus. In our designed platform, the CAN bus uses the standard protocol with the high data transmission rate to control the smart car electric system, such as door locking, LED displays, steering buttons and so on.

The CAN message service is used for data exchange, applications integration and smart car diagnoses. It saves its content data from CAN bus in XML format in the cloud. Access is available through the designed smart car information service as the interface.

Some smart car services in the cloud can periodically update context and data. For example, a traffic condition report service could similarly send to the navigation service periodic updates of local traffic conditions so that affected users could adapt their travel plans correspondingly.

In our platform, the readily available information attributes from CAN bus includes: Time, Channel, Name, Send node, Data and so on. In a CAN bus registry, the message is stored in the XML format, and data types are defined by using XML Schema. Ideally, we use the CAN message service to deal with the CAN messages XML file. Other smart car services can invoke this CAN message service to archive the smart car information sharing in the cloud. The process is shown in Figure 4.

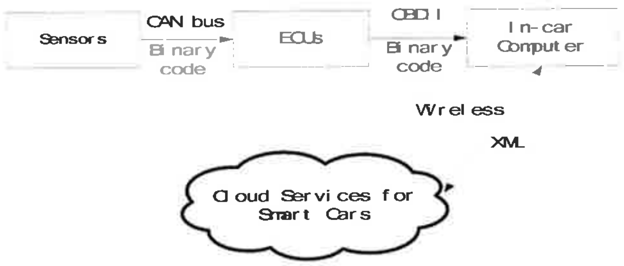

Figure 4 Data conversion process

4 Core Services for the Platform

4.1 Smart Car Information Service (SCIS)

The proposed information integration platform offers the infrastructural support for smart car services to boost both information sharing and overall performance in smart controls. At the heart of this platform is SCIS which provides a single-access point for the user to access various smart car applications. The main functions provided by SCIS include:

- User profile management,
- Smart car services registration and discovery,
- Smart car services access

User profile management is enabled through SCIS and provides an easy, reliable way for managing smart car users' information in the designed platform. User profile management function in SCIS ensures that the user's personal information can be applied to the information integration platform and smart car applications, for the purposes such as user identification, authentication, and authorisation. User authentication is one of the core features in the user profile management that provides SCIS with the ability to identify and control the state of users. Once authenticated, the smart car user will automatically have access to the respective smart car services in the cloud, to which they are entitled. At the same time, SCIS is able to correlate the user with his/her data and information stored in the cloud. Depending on the use case scenario, a user's data and information could include the CAN information of the car that the user is driving, personal configuration information, entertainment data such as music, route points (POI) files, and so on. Therefore, it is easy to synchronise data and services in the cloud when needed. For example, in the cloud-based Information Integration Platform for Smart Cars, users can access their smart car applications from mobile phone or other devices, SCIS ensures users get a consistent experience every time. When users login into SCIS or launch a smart car application, they will have everything with their own personal settings.

SCIS also allows smart car services to be registered. The registry contains information about the smart car service. Smart car users can discover a needed service by searching the registry. As we described above, various smart car services are self-contained modules that can be described, published, located, orchestrated, and programmed using XML based technologies. Thus, it is possible to provide a smart car application list or store for smart car user to select their needed services.

As a single access point to users, SCIS presents to users other smart car services. This is very convenient when a user drives another smart car, as those smart car services he/she already has can be accessed without re-downloading the applications. Many SaaS applications for smart cars will be available at little or no cost in the online application market.

SCIS is designed to be interrogated to provide access to other smart car services. It provides a directory service to smart car users and acts as a proxy for accessing those services. It intends to replace the traditional smart car software, and you can access SCIS via other devices, such as mobile phones and a table PC. There can be potentially many designed smart car services hosted in the cloud, with well-defined interfaces and in-built business processes. List 1 below shows the Smart Car Information Service Interface. According to the SCIS interface, the port "Smart car information service" defines the operation as “Login”. The login operation requires an input message “LoginRequest". The output message as the response is “LoginResponse”. The user’s information will be sent to SCIS though the interface to authentication the smart car user.

```
- <definitions targetNamespace="http://175.41.143.46/soap/Smart car information service">
  - <types>
    - <xsd:schema targetNamespace="http://175.41.143.46/soap/Smart car information service">
        <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
        <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/"/>
      </xsd:schema>
    </types>
  - <message name="LoginRequest">
      <part name="UserName" type="xsd:string"/>
      <part name="Password" type="xsd:string"/>
    </message>
  - <message name="LoginResponse">
      <part name="return" type="xsd:string"/>
    </message>
  - <portType name="Smart car information servicePortType">
    - <operation name="Login">
        <documentation>user login.</documentation>
        <input message="tns:LoginRequest"/>
        <output message="tns:LoginResponse"/>
      </operation>
    </portType>
  - <binding name="Smart car information serviceBinding" type="tns:Smart car information servicePortType">
      <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    - <operation name="Login">
        <soap:operation soapAction="urn:sinus_wsdl#Login" style="rpc"/>
      - <input>
          <soap:body use="encoded" namespace="urn:sinus_wsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
        </input>
      - <output>
          <soap:body use="encoded" namespace="urn:sinus_wsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
        </output>
      </operation>
    </binding>
  - <service name="Smart car information service">
    - <port name="Smart car information servicePort" binding="tns:Smart car information serviceBinding">
        <soap:address location="http://175.41.143.46/webservices/login.php"/>
      </port>
    </service>
</definitions>
```

List 1 Smart Car Information Service Interface

4.2 CAN Message Service

The CAN message service is one of the core smart car services and will be loaded at start up. We designed the CAN message service to collect the CAN data file from smart cars periodically. The CAN data can then be managed as a smart car information resource in our designed platform for various smart car services. The CAN message service can also diagnose the status of the smart car based on the collected CAN data, and notify the smart car user of any abnormal condition via SCIS. Moreover, the CAN message service is able to identify a smart car. Before processing the CAN message file, the CAN message service will require the user to login to identify the specific CAN message file in its database. By adopting authentication and authorization for accessing CAN message resources, smart car user permission can be verified before granting access to CAN message service, and user activity can be monitored through various logging records. For user authentication, CAN message service provides a Login() function. One user can have different CAN message database from different smart cars in the cloud.

The cloud-based information integration platform creates the opportunity for making an information-delivery approach from the smart car to the cloud. The CAN message service is important for smart cars to achieve the smart processing with the powerful computing ability of the cloud.

The advantage of using Web service interfaces to deal with CAN messages is that we can receive standard-compliant results from the web services. We can easily integrate these results along with other standard-based mapping components, into other smart car applications to create more powerful smart car applications for users. As we mentioned before, the CAN message service has the ability to identify different smart cars. Once the CAN message service receives the forwarded message from SCIS, the CAN message file in the cloud will be linked to the personal account. The CAN message service will process the correct CAN message file in the cloud since the user login SCIS. The CAN message file contains parameters of the CAN bus under the XML format.

4.3 Location Service

Location is critical information for many smart car services such as roadside assistance and travel planning. Therefore, a location service is one of the core facilitating smart car services. We use the following approach to access the location information.

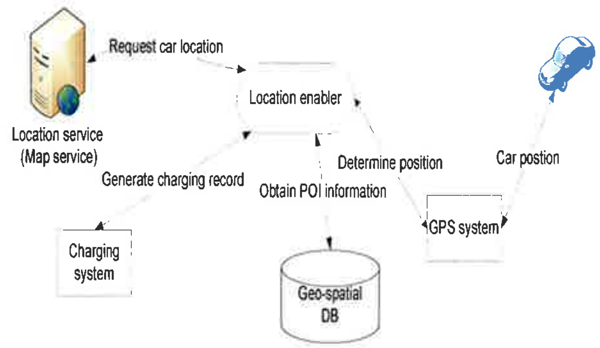

Figure 5 The location request and response flow in the platform [18]

The context model diagram in Figure 5 illustrates our location request and response flow. The location enabler provides the location information from other coordinate data sources such as charging system, geo-spatial database and GPS system. The location service collects the correct coordinates of the smart car from the location enabler by sending a car location request. Then other smart car services just invoke it to get the location data. Other smart car services are designed to use the same location service in sync, which is also a natural fit for smart cars. We choose the Google map service as our location service in our Cloud-based Information Integration Platform for Smart Cars.

5 Case study

5.1 Car Repair Facilitation Scenario

The car repair facilitation scenario demonstrates the working process in the Smart Car Information Integration Platform.

Figure 6 The Business Process of the Car Repair Facilitation Scenario

The car repair facilitation scenario starts when receiving the input and request from the smart car. Generally, the input includes CAN message file and location information. Once the CAN message service is invoked by SCIS, SCIS will forward the CAN message file to the CAN message service. The CAN message file will be stored in the cloud to analyse and judge whether the status of the smart car is normal. If not, the smart car repair service will find out where the problem is and indicate if the car can continue moving. A Smart car user can request the roadside assistance service when they cannot drive the car. In this situation, the roadside assistance company, as the service provider, will go to the user's place. The roadside assistance company can arrange their task based on the received GPS location information and error message from the smart car. Sometimes the smart car users can continue to drive their cars to the nearby car repair station to seek help. In this case, the car repair service will find out the correct car repair station by invoking the map service and car repair station service, and then the map service will provide the navigation to the user. The business process of this scenario is described as in Figure 6.

As we outlined above, in our designed platform the Web service interactions can be described in business processes. Our proposed car repair process in the BPEL exports and imports information by using smart car web service interfaces exclusively.

5.2 Message Flow

To support the car repair facilitation scenario, a number of messages needs to be exchanged between the in car computer and the services in the cloud in order to share information and assist decision making. An open and simple protocol is designed to govern the message exchange. The message flow in our scenario is shown in Figure 7.

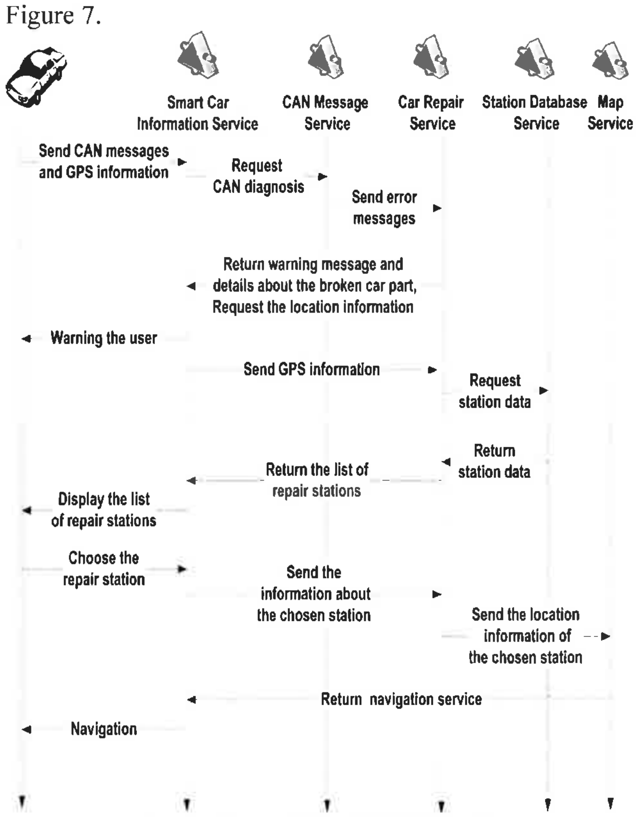

Figure 7 Message Flow of the Case Study

Firstly, SCIS will collect messages from the CAN bus in real time. Meantime, the message from the GPS device is provided to the smart car information service. The CAN message file and the location information will be collected to smart car information service periodically. The smart car information service will forward the CAN message file to the CAN message service for storage and diagnosis. If an abnormality is found as the result of the car diagnosis, the CAN message service will invoke the car repair service to process the error message. After that, the car repair service will notify the user of the abnormality and request the map service and station database service automatically. During this period, the user could choose their prefer car repair station based on the provided information from the station database service. The GPS location information will be provided to the car repair service to find nearby car repair stations.

The objective of the car repair service is to establish if the car is still safe to drive and to find all the information to make the car be ready to be repaired. To achieve it, the car repair service will find out the car parts that are in need of repair via the CAN message service. With the car parts information, the car repair service will find an appropriate car repair stations list by querying a station database service. The selection of stations could be based on the location, user pre-specified preferences and feasibility of the travel range. Once the car repair station is chosen by the smart car user, the smart car information service will send the location information request to the map service. Finally, the location information is returned to the smart car information service and then to the smart car for information display and navigation.

5.3 Context in the Scenario

The user-context contains the dynamic context and static context in our scenario. For example, the tyre which is of low pressure is part of user's dynamic context. Other dynamic context elements include the maximum distance the car could travel, as well as the time and date used for checking the car repair stations' opening hours.

Figure 8 shows an activity diagram for the scenario and the associated static and dynamic elements of the context.

As dynamic context for user, car location, travel range, time and broken atuo parts are included. The details about the car location are street, city, suburb and zip. The travel range depends on the petrol level in the smart car. Normally, the sub-class time includes data, time of day and day of week. The information about the broken part is very important.

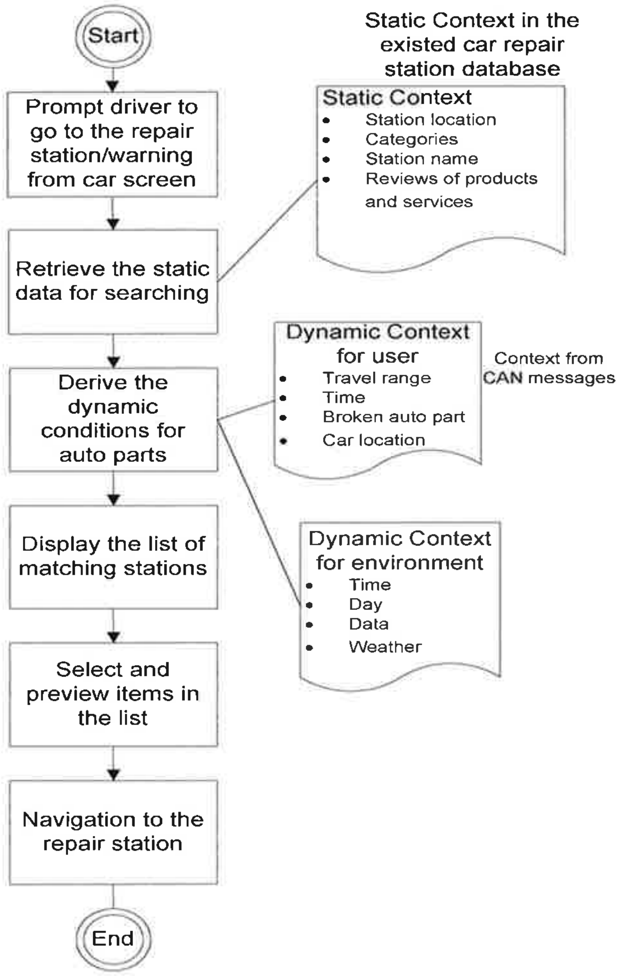

Figure 8 Application Activity Diagram with Example Context Elements

The RDFS graph in figure 9 depicts RDFS resources of user dynamic context in the described scenario.

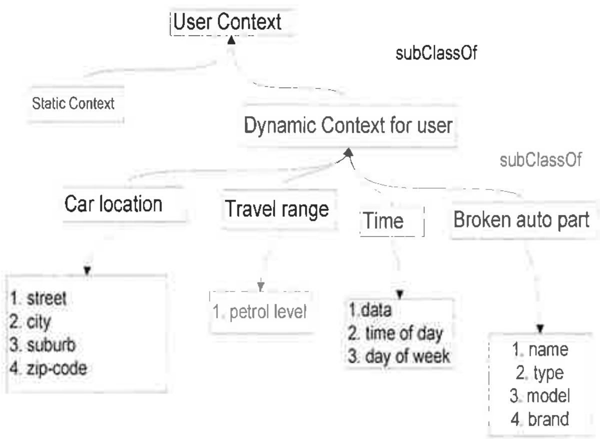

Figure 9 RDFS Graph of User Dynamic Context in Scenario

We store the static context in the car repair service. The RDFS graph in Figure 10 depicts RDFS resources of the user static context in the described scenario.

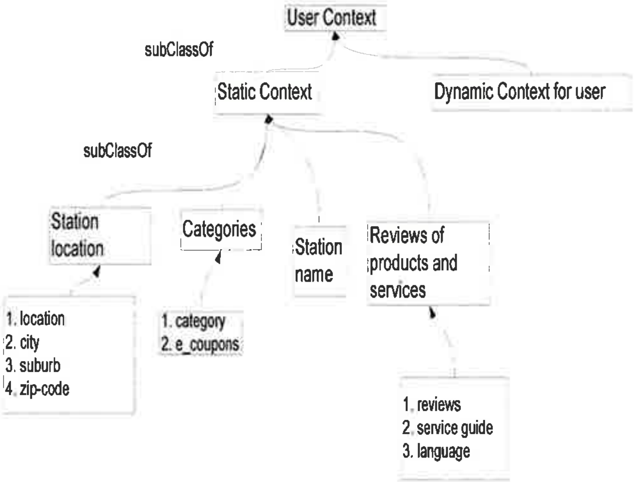

Figure 10 RDFS Graph of User Static Context in Scenario

SCIS finds addresses of repair stations based on the existing database. It then returns a point location that contains the latitude-longitude coordinates of those stations. Figure 11 is the result page of designed SCIS. The smart car user could check the car status manually by viewing the details of the smart car. We can see that the map service is involved to display the locations of car repair stations. A list of nearby car repair stations is shown below.

Car Information Service

MITSUBISHI MAGNA TL 3.5 Lit. Sedan Auto 2003 View Details VIEW BY MAP

Name	Address	Price (AUD inc. gst)	Description
Wollongong Auto Excellence	257 Keira Street, WOLLONGONG DC	$172.29	4 5
David Carlon Motors	105-107 Princes Highway, UNANDERRA	$163.95	2 3
Practic Automotive Services	32 Albert St, CORRIMAL	$135.27	2 4
Gary's Mechanical Repairs & Service	Unit 2, 232 Shellharbour Rd, WARILLA	$162.91	2 2
Kelly's Car Repairs	1 Loftus Drive, BARRACK HEIGHTS	$172.95	1 2
Cougar Automotive	77 Old Lake Entrance Rd, OAK FLATS	$126.85	3 3
Avendel Automotive Service	3/113 Industrial Road, OAK FLATS	$158.45	2 2
ULTRA TUNE ALBION PARK	UNIT 4/29-31 DURGADIN DRIVE, ALBION PARK RAIL, ALBION PARK RAIL	$184.85	2 4
Helensburgh Car Services	107 Parkes Street, HELENSBURGH	$167.95	2 3
G.A.S	6 Maxwell Place, NARELLAN	$151.89	2 2

Figure 11 Search Result of Designed Service

6 Conclusion

This paper analysed the challenges and issues of the contemporary smart car technologies, and advocated that cloud computing, as the next generation of the computing paradigm, complements the smart car technology. Motivated by this finding, this research explored the architecture for Cloud-based Information Integration Platform. With this platform, two core services, the smart car information service and the CAN message service, were implemented to support to information integration. The Smart Car Information Service (SCIS) is acting as an interface at the heart of this platform.

The proposed platform brings a number of potential benefits. The designed Cloud-based Information Integration Platform for Smart Cars is economical and practical for both car users and smart car service providers. By adopting SOA and SaaS in Cloud-based Information Integration Platform, the cost of smart car applications, design, development, deployment and services support will be reduced. Meanwhile, smart car application developers can design different products by using common hardware standards to achieve flexibility, functionality, low cost and success.

We recognise that there some limitations in our research. The major research limitation of this study is the failure to collect and analyse data that would yield the relationship between smart cars and an ICT-based system. Furthermore, we did not address the reliability and security issues in the Cloud-based Information Integration Platform. In the design of the platform and smart car core services, performance was not considered. In the real environment, some factors will impact on the performance of the designed platform, such as the network speed, which might not be fast enough in some areas.

References

[1] *Amazon Elastic Compute Cloud (Amazon EC2).* 2010; Available from: http://aws.amazon.com/ec2/.

[2] Zhang Shufen, Zhang Shuai, Chen Xuebin, and Wu Shangzhuo. *Analysis and Research of Cloud Computing System Instance.* in *Future Networks, 2010. ICFN '10. Second International Conference on.* 2010.

[3] Khmelevsky Youry and Voytenko Volodymyr, *Cloud computing infrastructure prototype for university education and research*, in *Proceedings of the 15th Western Canadian Conference on Computing Education.* 2010, ACM: Kelowna, British Columbia, Canada.

[4] Anand V. Hudli, Shivaradhya Balasubrahmanya, and Raghu V. Hudli, *Level-4 SaaS applications for healthcare industry*, in *Proceedings of the 2nd Bangalore Annual Compute Conference.* 2009, ACM: Bangalore, India.

[5] Nitu, *Configurability in SaaS (software as a service) applications*, in *Proceedings of the 2nd India software engineering conference.* 2009, ACM: Pune, India.

[6] *What is SAAS?* 2010; Available from: http://www.whatissaas.net/.

[7] Gabriel Leen and Dona Heffernan, *Expanding automotive electronic systems.* Computer, 2002. **35**(1): p. 88-93.

[8]. Robert Davis, Alan Burns, Reinder Bril, and Johan Lukkien, *Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised.* Real-Time Systems, 2007. **35**(3): p. 239-272.

[9] Dominique Paret, *Multiplexed networks for embedded systems : CAN, LIN, flexray, safe-by-wire...* 2007, Chichester: John Wiley & Sons, Ltd.

[10] Nilüfer Çenesİz and Murat Esin, *Controller area network (CAN) for computer integrated manufacturing systems.* Journal of Intelligent Manufacturing, 2004. **15**(4): p. 481-489.

[11] Ken Tindell and Alan Burns. *Introduction Guaranteeing Message Latencies on Control Area Network (CAN).* Available from: http://www.control.aau.dk/~jdn/edu/courses/e07/distrsys/can-iccb-tb-sep94.pdf.

[12] Gavin D. J. Harper, *Build Your Own Car PC.* 2006, Sydney: McGraw-Hill.

[13] Hull Bret, Bychkovsky Vladimir, Zhang Yang, Chen Kevin, Goraczko Michel, Miu Allen, Shih Eugene, Balakrishnan Hari, and Madden Samuel, *CarTel: a distributed mobile sensor computing system*, in *Proceedings of the 4th international conference on Embedded networked sensor systems.* 2006, ACM: Boulder, Colorado, USA.

[14] *Capabilities of our vehicle computers.* 2010; Available from: http://www.in-carpc.co.uk/features.htm.

[15] Albrecht Schmidt, Anind K. Dey, Andrew L. Kun, and Wolfgang Spiessl, *Automotive user interfaces: human computer interaction in the car*, in *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems.* 2010, ACM: Atlanta, Georgia, USA.

[16] Jian Hu and Gangyan Li. *CAN-based passenger car starter information integrated control method and its implementation.* in *Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on.* 2009.

[17] Niall Murphy, *A Short trip on the CAN bus.* Embedded Systems Programming, 2003. **16**(9): p. 9-+.

[18] Ariel Pashtan, *Mobile Web Services.* 2005, Cambridge: Cambirdge University Press.

Biographies

Yi Xu received the MS degree in Information and Communication Technology from University of Wollongong in 2010. He is currently a research student in School of Information Systems and Technology, University of Wollongong, Australia.

Jun Yan received BS and MS in computer and its application from Southeast University, China, in 1998 and 2001, respectively, and the PhD degree from Swinburne University of Technology, Australia, in 2004. He is currently a senior lecturer in School of Information Systems and Technology, University of Wollongong, Australia. His research interests include workflow management, service-oriented computing, SLA management, and cloud computing.