Vehicle ECU Classification and Software Architectural Implications

Win-Bin See
Aerospace Industrial Development Company, Taichung, Taiwan, ROC
winbinsee@ms.aidc.com.tw

ABSTRACT

Modern vehicle integrates a large amount of electronic devices to improve the driving safety and comfort. This growing number of electronic control units (ECU) with sophisticated software escalates the vehicle system design complexity. In this article, we propose a classification scheme for vehicle ECUs into five categories; namely the emission related control, non-emission related control, telematic, infotainment and off-board ECU.

We describe specific attributes for each class and address the software architectural implication under this classification. We present a baseline software architectural model for each ECU category. This classification scheme, with associated software architectural model, will help in identifying the focus of interest for this demanding automotive embedded system application domain. This classification scheme and the baseline software architectural models will also assist in exploring the chance for software reuse in automotive ECU software development domain.

Keywords: automotive electronics; embedded software; software architecture.

1: INTRODUCTIONS

Modern vehicle integrates a large amount of electronic devices to increase the driving safety and comfort. These growing number of on-board and off-board embedded *electronic control units (ECU)* escalates the complexity and cost in vehicle system development. To reduce the vehicle development cost, automotive industrial consortiums are working on standards for automotive electronic system and software architecture. These standards would increase the commonality and reusability in ECU design, and reduce the system cost accordingly.

In Japan, JASPAR [1] (Japan Automotive Software Platform and Architecture) organization has been formed by September 2004, with Nissan and Toyota as founding members. In Europe, AUTOSAR [2] (Automotive Open System Architecture) organization has been formed by September 2003. Major vehicle system manufactures and subsystem suppliers have been joined as members of AUTOSAR organization.

Classification of ECU will assist the automotive system architecture standardization effort, increase the possibility of software reuse, and reduce the system development cost accordingly. In this article, we classify these devices into control-oriented ECU and infotainment oriented ECU, describe their differences and software architectural implications.

This article is organized as follows: Section 2 describes automotive electronic system architecture. Section 3 presents our ECU classification scheme. Section 4 describes the software architectural implications based on our ECU classification scheme. Section 5 gives the conclusion.

2: AUTOMOTIVE ELECTRONIC SYSTEM ARCHITECTURE

Vehicle electronic system consists of vehicle on-board devices, and off-board devices. With growing number of on-board devices, multiplex data bus is used to connect them into a collaborative and integrated system. Vehicle bus drastically reduces the complexity and weight of wiring harness, compared to previous point-to-point interconnections. Controller area network (CAN) is one of the most widely used vehicle standard bus [3][4]. Vehicle having multiple buses is common situation nowadays. Gateway ECU bridges the busses into an integrated system. Figure 1 gives an illustrative vehicle electronic system architecture diagram. Two buses, *Bus1* and *Bus2*, have been presented in the system. *Gateway ECU* relays the information flows between the two buses.

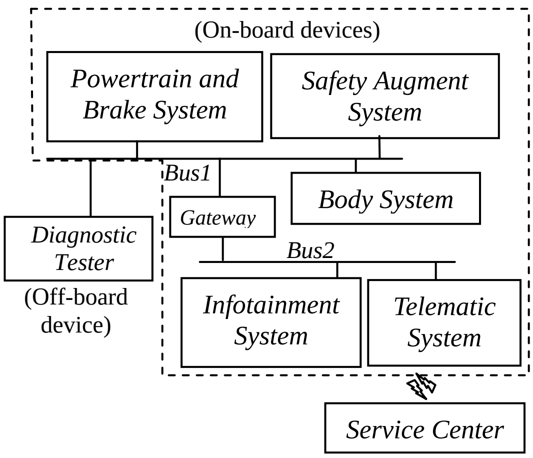

Figure 1. Vehicle electronic system architecture illustration

Bus1 connects *Powertrain and Brake System*, *Safety Augment System*, and the *Body System*. *Bus2* connects the *Infotainment System* and *Telematic System*. In the lower left, the *Diagnostic Tester* is an off-board ECU that connects to the vehicle by workshop service engineer while troubleshooting the vehicle electronic system. Following on-board diagnostic (OBD) software standards [5][6], *Diagnostic Tester* can examine the whole vehicle electronic system, through a connector under the dashboard, and talk to all the ECUs attached to vehicle busses.

Powertrain System consists of engine control module, transmission control module and collaborates with the brake system to act as the core of automotive system control. *Brake System* belongs to the vehicle chassis system that helps the driver to prevent from wheel skidding by *antilock brake system* (ABS), and maintains the vehicle stability during brake through *electronic brake force distribution* (EBD). *Antilock brake system* automatically adjusts the brake pressures in up to ten times a second to relieve the wheel from dangerous skidding situation. *Body System* controls the vehicle interior, exterior lights, doors and ignition key. Typical *Safety Augment Systems* are tire pressure monitoring system (TPMS), back sonar system, head up display (HUD) system, etc. *Infotainment System* consists of on-board entertainment devices are radio receiver and CD/DVD player. Recently, MP3, digital broadcasting TV, wireless LAN and 3G enabled web connection are maturing into the vehicle. This makes the infotainment capability becomes an indicator for carmakers market competitiveness. On-board *Telematics System* ECU serves as a remote agent to the *Service Center* of telematics service provider. Telematics application system integrates telecommunication and informatics technology to provide vehicle driver with value-added service and information [7].

3: ECU CLASSIFICATION SCHEME

Having the vehicle electronic system architecture as shown in Figure 1, we further analyze the vehicle subsystem attributes that pertain to the role of the different ECU and then classify them into five categories; namely, 1) the *Emission Related Control*, 2) *Non-Emission Related Control*, 3) *Telematic*, 4) *Infotainment*, and 5) *Off-board* ECU. Table 1 presents our ECU categorization from the perspective of ECU role and attributes that could affect the software architectural design. This classification scheme helps to identify the area of interest and focus the effort for this fast growing automotive embedded system domain.

The first category is the *Emission Related Control ECU*, it has the attributes of being safety critical, and it needs to comply with the local government emission regulations, it has less user interface, and having product lifecycle as the life span of the vehicle. The second category, the *Non Emission Related Control ECU*, aims at improving the vehicle usability. Its safety regulation compliance is conditioned by the usage of ECU. Body control is an example of this ECU class; it controls interior lights to better serve the vehicle driver and passengers. The third one, *Telematic ECU* class furnishes vehicle driver with additional value-added services, it needs to fulfill the service commitment promoted by the carmaker or the service provider. The fourth class, Infotainment ECU is the class that accommodates most consumer electronics widgets. It is less safety related, and it needs to have fancy user interface and neat feature size. This class of ECU has shorter lifecycle than those ECUs that directly serve the vehicle driving operation. The last category is the *Off-board ECU*. *Diagnostic Tester*, used in the vehicle maintenance shop, belongs to this. It interacts with the on-board electronics through vehicle bus to examine their operation status. It hooks to the vehicle bus through the cable that connects to the vehicle test connector. The diagnostic tester software developer needs to accommodate multiple vehicle models in one tester, and to make their design flexible to future vehicle system evolutions. Recently, the cost of near field wireless communication solution has been lowered to become an optional cable replacement for off-board ECU to vehicle interconnection.

ECU Software Category	Attributes
1. *Emission Related Control ECU*	-Safety critical -Emission regulation compliance -Less user interface -Long product life cycle.
2. *Non Emission Related Control ECU*	-Vehicle usability concern -Safety regulation (conditional) -Less user interface -Long product life cycle
3. Telematic ECU	-Service commitment -Conditional user interface complexity -Medium product life cycle
4. Infotainment ECU	-Less Safety Related -Fancy user interface -Neat feature size -Adapting consumer electronics features -Shorter product life cycle
5. Off-board ECU	-Interacts with on-board system -Adaptive to multiple vehicle models -Short range wireless link

Table 1. ECU Software Categories

4: ECU CLASSIFICATION AND SOFTWARE ARCHITECTURAL IMPLICATION

In software product development, software development life cycle process will have to reflect the required target system functional and non-functional requirements. Depend on the role and attribute of specific ECU, different ECU that resides in the same vehicle would need different level of rigorousness in software development process. Based on the classification of vehicle ECU listed in Table 1, we further explore the software architectural implication associated with each category as follows.

4.1: EMISSION RELATED CONTROL ECU SOFTWARE ARCHITECTURE

Emission Related Control ECU category includes engine control ECU and transmission control ECU. They are safety critical and need to collaborate to produce the driving power and distribute it appropriately to the wheels. Engine control ECU monitors various sensors and controls the valves in keeping the vehicle emission under government standard. Transmission control ECU implements the shift control. In an automated transmission vehicle, the transmission control ECU shall provide the driver with smooth shifting operation and ensure the transmission system durability.

This ECU class controls the vehicle driving electronics, mechanical, and hydraulic parts. It has sophisticated real-time control algorithmic logic and having a product life cycle as long as the lifetime of the vehicle. A baseline software architecture diagram for this class of ECU is shown in Figure 2. It consists of seven software components that include the *Real-Time Operating (RTOS)*, *Network Control*, *Application Control*, *Sensor Control*, *Actuator Control*, *On-Board Diagnostic* and the *Self-Test* component. *RTOS schedules* the ECU software tasks for execution. *Sensor Control* and *Actuator Control* components control the related sensors and mechanics through input and output interface circuitry. *Network Control* component manages the vehicle bus and enables the distributed ECU communication and coordination. *Application Control* component implement the algorithm for the required functionality that integrates the connected sensors and actuators and other devices on the bus. *Self-Test* component continuously examines the ECU hardware and input/output hardware. The result of *Self-Test* will be reported to vehicle driver through warning lights on the dashboard and to the off-board diagnostic tester through *On-Board Diagnostic* component.

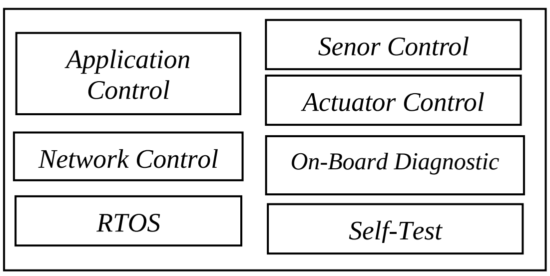

Figure 2. Emission Related Control ECU Software Architecture

To reduce the development cost of ECU real-time operating system and vehicle network management software, European automotive consortium developed a shared vision over core software architecture that covers these two software components. In 1994, OSEK/VDX standard provides a set of application program interface (API) that abstract the underlying hardware to facilitate *RTOS* design and code reuse [11][12]. The OSEK/VDX defines a real-time kernel that can be implemented in small memory footprint in 8-bit or 16-bit micro-controllers. The OSEK is the German acronym for “Open systems and the corresponding interfaces for automotive electronics”; The VDX means “Vehicle Distributed eXecutive”. Layered software architecture has been defined in the OSEK/VDX standard. However, this kind of RTOS manages small amount of tasks; many suppliers adopt in-house solutions instead.

4.2: NON- EMISSION RELATED CONTROL ECU SOFTWARE ARCHITECTURE

Some ECU controls the vehicle electrical and mechanical parts to fulfill non-emission related vehicle system functions. *Body System ECU* and Steering *Column Switch Control ECU* are examples of this category. Being with control-oriented software attribute, the architecture for this ECU software is similar to emission related control ECU. It shares the same baseline software architecture with *Emission Related Control ECU*, as shown in Figure 2.

The differences between *Emission and Non-Emission Related Control ECU* count on the application function and the safety critical level. The rigorousness in the software development cycle process will have to respect these differences. From the aspect of software task control, simple and reliable RTOS would be enough for this kind of ECU. This class of ECU has also been connected to core vehicle network, and it can be diagnostic through off-board OBD diagnostic tester. The *Network Control* and *On-Board Diagnostic* software components can be shared for these two ECU categories. Complexity of the ECU depends on the application function requirement, sensors and actuators under control.

4.3: TELEMATIC ECU SOFTWARE ARCHITECTURE

Intelligent Transport System (ITS) is the integration of the human, the road, and the vehicle to improve the transportation safety, efficiency, comfortableness and environmental friendliness. Telematics is a branch of ITS application domain. Telematics integrates the telecommunications and the informatics to provide information service for vehicle drivers or pedestrians through vehicle *on-board unit (OBU)* or portable device by advanced information and communication technology [7]. Telematic service system constitutes of three major constituents, the vehicle on-board unit, connection infrastructure and the service agent.

Electronic Toll Collection (ETC), vehicle born tracking and dispatching mobile data terminal (MDT) are typical telematic service equipment integrated into passenger vehicles and commercial dispatching application fleet. Vehicle on-board device wirelessly interacts with external service agent to provide integrated services to the vehicle driver; this belongs to the area of telematic applications.

ETC system consists of vehicle born on-board-unit (OBU), roadside equipment and the ETC information center. Vehicle OBU communicates with roadside equipment to carry out identification and payment accordingly, and then, payment information will be transferred to the backend ETC information center for further processing.

Since 2003, the GSM GPRS (General Packet Radio Service) mobile communication service starts to operate and provides low cost and high geographical coverage data communication. This technology provides a reliable and real-time data communication infrastructure for telematic services that vehicle geographical position tracking and data communication. Telematic services are integrating into passenger car to become a product differentiation scheme for car manufacturers. They integrate mobile communication, global positioning system (GPS) and optionally with on-line service crew from service call center. Example application service systems are the BMW *ConnectedDrive* [8] in U.S. and the Nissan *TOBE* [9] in Taiwan.

A GPRS and GPS assisted telematic interaction session can be bidirectional; it can be initiated from driver-side or from service center. In a driver-side initiated session, vehicle driver initiates a service request through request button on a *User Interface (UI) management* component. An on-board telematics service computer, or *mobile data terminal (MDT)*, collects the time and location data from GPS receiver, packs the vehicle identification and the request information with the time and location data, and then forwards them as a data packet to external *Service Center*. The service personnel ask for external help from security service agent or police department as required. Figure 3 depicts this driver-side initiated telematic service message interaction sequences. This style of interaction can be applied in passenger vehicle service system or logistic distribution fleet management system [10]. Telematic system application functionality depends on the integration of positioning device, the setup of a reliable communication link between vehicle and the service center. In the case of GPRS communication, a TCP/IP socket connection will be built for each vehicle. A fleet management system could operate thousands of moving vehicles at the same time in real-time.

Figure 4 shows the software component architecture for telematic ECU. *GPS Control* software component interacts with the GPS hardware modules to collect the geographical position of data of the vehicle. A reliable communication link would need to be well established, *Mobile Communication* component takes the responsibility. *Sensor Control* component collects the sensor data required for vehicle security and operating situation awareness. *User Interface (UI) Management* interacts with vehicle driver. The *Application Control* component integrates the data collected from *Sensor Control* and *GPS Control* components, associated them with predefined thresh hold condition to raise and send the event to remote *Service Center* automatically. *Network Control* handles the vehicle bus communication. *Self-Test* component watches the healthiness of the ECU hardware and interfaces. On-Board Diagnostic component serves to communicate with the off-board Diagnostic Tester to report the ECU healthiness. Simple RTOS will be sufficient to schedule all these on-board tasks.

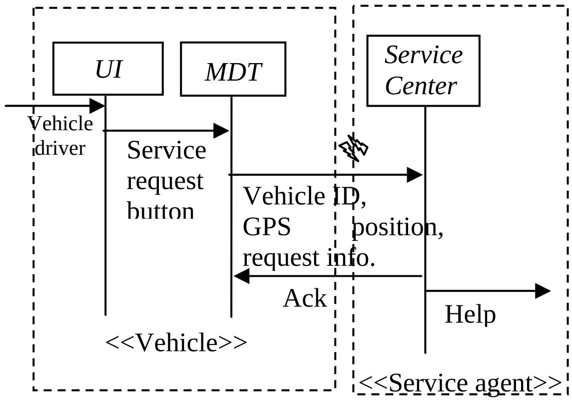

Figure 3. Telematic Service Message Interactions

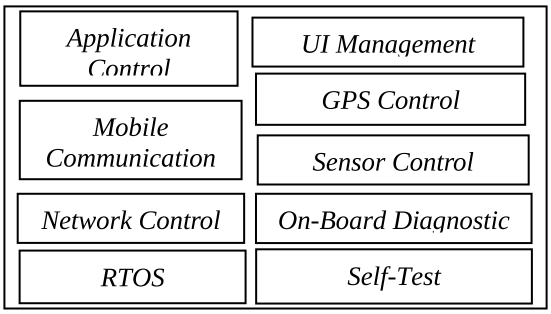

Figure 4. Telematic ECU Software Architecture

4.4: INFOTAINMENT ECU SOFTWARE ARCHITECTURE

Recently, there have been more consumer electronic products integrated into passenger vehicles; they include digital broadcasting TV, wireless LAN technologies, 3G, Bluetooth communication and TFT display. Most of these new products have been manifested in vehicle infotainment and telematic applications. Wide-Area wireless communication and digital TV broadcasting are also pushing web based services and digital TV into vehicle borne infotainment system. The richness and style in the adoption of these new technologies helps the Carmaker to differentiate their product. Carmakers in the market are required by customers to adapt these technical advancements swifter than ever before. There are also industrial consortiums that endeavor to come up with automotive infotainment software architectural standard that would reduce the system cost through the sharing of software development effort and research results. *Open Service Gateway Initiative (OSGi)* is and example of this effort. *OSGi* has been gaining a lot attention in domain of home networking and automotive applications [11][14].

Figure 5 gives an infotainment ECU software architecture that takes result of recent industrial efforts into account. In this architecture, *Network Control*, *On-Board Diagnostic* components are similar to other ECU classes. The architecture has been designed to add and modify *Service* components into the ECU. *User Interface Management* component integrates and presents the services offered by *Service* components into the infotainment display. Service components are built upon an *Intermediate Machine* to facilitate hardware independence and software portability. *Native Machine* interface component accommodates the control mechanism for hardware dependent feature. The choice of Operating System depends on the availability of the *Intermediate Machine* component and the required *Service* components for particular target platform.

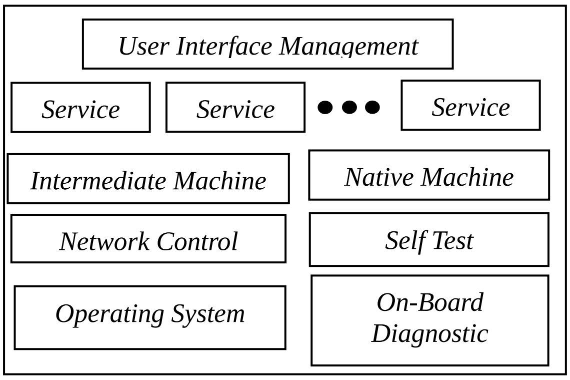

Figure 5. Infotainment ECU Software Architecture

Ford's Vehicle Consumer Services Interface (VCSI) [13] defines flexible layered software architecture through a series of generic application program interface (API). This VSCI approach aims to lets the software suppliers design their components to fulfill buyer's need and retain their own proprietary property at the same time. This Ford's VCSI architecture and the *OSGi* framework bear similar architectural style to the baseline software architecture presented in Figure 5.

4.5: OFF-BOARD ECU SOFTWARE ARCHITECTURE

Industrial standard organization defines a set of vehicle network protocol standards for vehicle diagnostic function design and implementation [4][5][6]. Based on this set of standards, on-board ECU plays the role of *Diagnostic Server* and the off-board *Diagnostic Tester* plays the role of *Diagnostic Client*. *Diagnostic Client* sends the service request and collecting service response from *Diagnostic Server* ECU. *Diagnostic Tester* connects to the vehicle bus via a single test connector, and interacts with all on-board ECUs to collect and report the ECU failure status stored on ECU. The user interface of this off-board diagnostic tester shall consider the maintenance work shop floor environment and make it friendly to the service engineer.

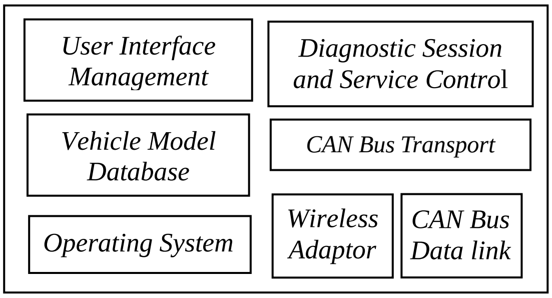

Figure 6. Diagnostic Tester Software Architecture

Figure 6 presents our baseline software architecture for the off-board ECU, OBD *Diagnostic Tester*. It consists of seven components, the 1) *User Interface Management,* 2) *Vehicle Model Database,* 3) *Diagnostic Session and Service Control,* 4) *CAN Bus Transport,* 5) *CAN Bus Data Link,* 6) *Wireless Adaptor*, and the 7) *Operating System.*

User Interface Management component receives the user service request and translates it and hands it over to the *Diagnostic Session and Service Control* component for service request command issuing. The separation of *User Interface Management* component, from other diagnostic protocol handling software components, enables the flexibility in adapting to user operation preference without disturbing underlying bus interface implementation. *Vehicle Model Database* provides the vehicle model configuration information to intelligently configure the diagnostic facilities for the user of the *Tester*. *CAN Bus Transport* component decomposes and ensembles bus data that spans the multiple frames. *CAN Bus Data Link* component is responsible for the data-link layer bus protocol handling. *Wireless Adaptor* component adapts short-range wireless communication technologies like Bluetooth [16] and Zigbee [17] to provide wireless connection between *Diagnostic Tester* and the vehicle. Wireless communication replaces the entangling cable in automotive workshop. The off-board Diagnostic Tester has been moving from proprietary hardware to commercial off the shelf one, general off-the shelf personal computer O*perating System* will be good for the integration of new handy software components as required.

5: CONCLUSION

Automotive electronics is a booming application domain that requires growing number of electronic control unit (ECU) to improve the driving safety and comfort. Classification in ECU and its software architecture will assist in focusing the research and development effort. In this article, we presented our ECU classification scheme that divides the ECU into the five categories based on the role and the attributes of the ECU. We also provide baseline software architectural model for each ECU category. This classification scheme, with associated baseline software architectural as template, could be used in deriving systematic ECU software development approach, and analyzing chances for further software reuse.

Currently, we are focusing our research in developing reusable software library and framework for various software architectural components in automotive ECU [18][19]. The usage of *aspect-oriented programming* [15] method is being promoted from programming phase to software architectural design phase in delineating the software component boundary.

REFERENCES

[1] JASPAR,http://www.marklines.com/ja/jaspar. jsp.

[2] AUTOSAR, http://www.autosar.org.

[3] L. Lavagno and R. Zurawski, "Towards networking of Embedded Systems," IEEE IES Newsletter, Vol. 54, No. 2, pp. 12-15, June 2006.

[4] *ISO 11898: Road vehicles - Controller area network (CAN)*, International Standard Organization, 2003.

[5] *ISO 15765: Road vehicles - Diagnostics on controller area network (CAN),* International Standard Organization, 2004.

[6] *ISO 14229: Road vehicles – Unified Diagnostics service (UDS)*, International Standard Organization, 2004.

[7] K. Sakamoto, "The Development of Telematics in Japan," 2006 Taipei International Automotive Electronics Forum.

[8] M. Girardet, "Telematics and Tele services," 2006 Taipei International Automotive Electronics Forum.

[9] Nissan TOBE service in Taiwan, http://www.nissan.com.tw/club/tobe/tobe_index.asp?level=2&top=4&sub=11&left=1.

[10] W. -B. See, "Wireless Technologies for Logistic Distribution Process," In *Proceedings of he 4th International Conference on Supply Chain Management and Information Systems (SCMIS 2006)*, pp.291-297, Taichung, Taiwan, ROC, July 2006.

[11] J. Lemieux, Programming in the OSEK/VDX Environment, CMP books, 2001.

[12] M. O'Donnell, "OSEK/VDX: Driving the standard for optimized embedded systems," *Embedded Computing Design*, Winter 2003.

[13] C. Simonds, "Software for the Next Generation Automobile," IEEE *IT Professional*, November/December 2003, pp. 7-11.

[14] J. Zhou, "Telematics (Infotainment Solution from Siemens VDO)," 2006 Taipei International Automotive Electronics Forum.

[15] E. Baniassad, et. al, "Discovering Early Aspects," *IEEE Software*, January/February 2006, pp.61-70.

[16] C. Bisdikian, "An Overview of the Bluetooth Wireless Technology," *IEEE Communications Magazine*, Vol. 39, No. 11, December 2001, pp. 86-94.

[17] E. Callaway, et. al., "Home Networking with IEEE 802.15.4: A Developing Standard for Low-Rate Wireless Personal Area Networks," *IEEE Communications Magazine*, Vol. 40, No.8, August 2002, pp. 70-77.

[18] P. -A. Hsiung, S. -W. Lin, C. -H. Tseng, T. -Y. Lee, J. -M. Fu and W. -B. See, "VERTAF: An Application Framework for the Design and Verification of Embedded Real-Time Software," *IEEE Trans. on Software Engineering*, Vol. 30, No. 10, pp.656-674, October 2004.

[19] W. -B. See, "UML-based Modeling Approach for Automotive Electronics System Development", In *Proceedings of the IEEE International Conference on Industrial Computing (ICIT 2005)*, pp. 448-452, Hong Kong, December 2005.