Web based Management System of a Remote Mobile Device using Device Management Protocol

Hyun-Ki Ryu†, Sung-Rae Cho†, Piao Shiquan†, Jong-Wuk Son† and Woo-Young Jung†,

Division of Advanced Industrial Science& Technology, DaeGu GyeongBuk Institute of Science & Technology, Korea

Summary
This paper is to propose and implement a remote phone management system based on the open mobile alliance (OMA) SyncML device management (DM) protocol. The proposed remote phone management system consists of DM Client module, DM Server module and remote phone device management (RPDM) server/client module. Implemented DM client module is suitable to an open platform like Symbian S60. In order to minimize the porting work, a platform adaptation layer (PAL) between basic software and DM engine is designed for DM client. The DM server is consisted of server part, data part and HTTP interface part. It mainly treats the DM protocol and the request of the web admin and the DM client. The RPDM server/client module is deigned to provide a remote management environment based on the web interface. The concerned module functions and interfaces are examined by the SyncML conformance test suite (SCTS) for DM protocol provided by OMA to evaluate whether a normal operation is performed or not. It also showed the controlling method of the mobile devices under the new mobile environment by utilizing the system of RPDM and applying the verified interface.

Key words:
mobile device, open mobile alliance, device management, SyncML, open platform.

1. Introduction

Now under the ubiquitous environment that the IT society is oriented to the possession of various mobile devices by individuals and access to information through information instruments have become the crucial parts [2]. Also, as the use of a mobile device is popular and the range of service is extended, the functions of the hardware and software are gradually becoming complex and various. The conventional mobile device was mostly used for the function of voice calling only, but now there are various requirements for an application program through the terminal because of the recent improvement of the treatment rate of the mobile device and the explosive increase of its users. Accordingly, the significant factor to determine the marketability of the mobile device is whether the application programs are loaded to meet the demand of the user in the short period. For such a reason, the matter of the operation, maintenance and management of the mobile device has become the important issues.

Manuscript received April 5, 2008
Manuscript revised April 20, 2008

But as the method to operate, maintain and manage the mobile device depends on the particular method provided by the suppliers for every individual terminal instead of every user installing the application software by a consistent way under the mobile environment, a service to check the status of the hardware and software of the mobile device can not be provided under the present environment. Also as the software structures of the existing terminals mostly use the proprietary real time operating system (RTOS) provided by the manufacturers of the terminals, and further consist of middleware software and application software, the third party with the new application solution has a limitation for adding or deleting the application programs on the terminals. Thus under the management and control environment of the mobile device in the existing method, not only users but also manufacturers and service companies are in a situation of having to make many expenditures to manage their mobile devices. These problems of management occur for car information systems, home gateways, kiosks, set-top boxes, etc., as well as mobile devices. Accordingly, this paper, in order to control the mobile device under a wireless environment, suggests a remote device management system using the SyncML DM management standard of a wireless mobile telecommunication terminal based on SyncML of OMA which is the alliance of companies related to wireless telecommunication [1][7][8][9]. This system extends the SyncML data synchronization (DS) of data synchronization technology under the mobile telecommunication environment to the purpose of management and is designed to be managed through management information exchanges between the management server and management agent. This paper also introduces the composition of OMA DM and its interface briefly in chapter 2 and shows the DM client, server structure and individual architecture in chapter 3. The structure of RPDM and the result of the applicable scenario to be implemented to verify the concerned module interface are shown in chapter 4 and the conclusion will be in chapter 5.

2. OMA DM

OMA DM, the terminal management technology of the developmental combination of the DM technology of wireless application protocol (WAP) forum [6] and SyncML DM technology of SyncML initiative, is the technology with which the service supplier could make an installation in an effective manner for the user to provide an effective management method for the terminal and the terminal application program under the various network environments and define the relevant protocol and mechanism.

2.1 Functions of OMA DM

- Set up the composition information
- Set up the operation parameter of the terminal
- Software installation and function defining parameter
- Update the software and firmware
- Set up the application program
- Set up the user's preference

2.2 Characteristics of OMA DM

- It consists of the asymmetric protocol of a server-client structure, in which the server is in charge of providing the service for terminal management and delivering management command and the client is in charge of processing the terminal management command and advising the result.
- It ensures the continuous stability of the basic specification by separating the basic specification and the silo specification, and defines the mutually independent additional specification to reflect the various service requirements.
- It is applicable to the various network environments and provides the various transmission protocols such as HTTP, WAP and object exchange (OBEX), etc., and also transmits the management command and the result of processing in the form of a SyncML message.
- Through the multi-server, it allows several servers to access one terminal and performs the control of the terminal access authority of each server through the access control list.

Figure 1 shows the relation between the composition of the DM specification and each specification.

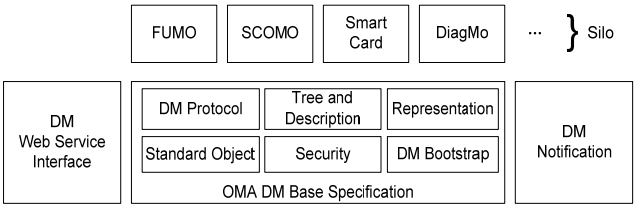


Fig 1. Composition of the OMA device management

2.3 Specification of DM Protocol

The specification of the DM protocol defines the telecommunication protocols for the terminal management interface between the DM server and DM client. The specification of the DM protocol consists of the setup phase for certifying the terminal and the exchange of terminal information, and the terminal management phase for performing the terminal management which is repeated several times, if necessary. The control words for setup and terminal management are transmitted in the form of a SyncML message. Figure 2 shows the composition phase of the DM protocol.

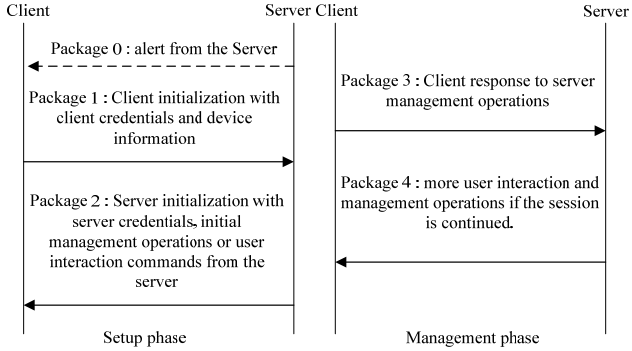


Fig 2. Composition phase of the DM protocol

3. DM Client and Server

This section shows the structure of the RPDM DM server and client to be implemented and explains the function of each detailed structure and performs the prior verifying process through the client emulator and SCTS [5] for verification of the function and mutual operations before the actual utilization of the DM server and client.

3.1 DM Client

The RPDM OMA DM client module implemented by the symbian platform [3][4] is classified as the DM engine

part and PAL. The detailed structure of RPDM DM client is shown Fig. 3. The DM engine performs the basic operation satisfying the DM enabler package release version 1.1.2 and is in charge of the functions for generating, sending and receiving, and processing. The object tree manager performs the function to manage every node of information of the device management object tree and the user interactive module performs the function of notification or confirmation to the user when the DM client starts the DM session.

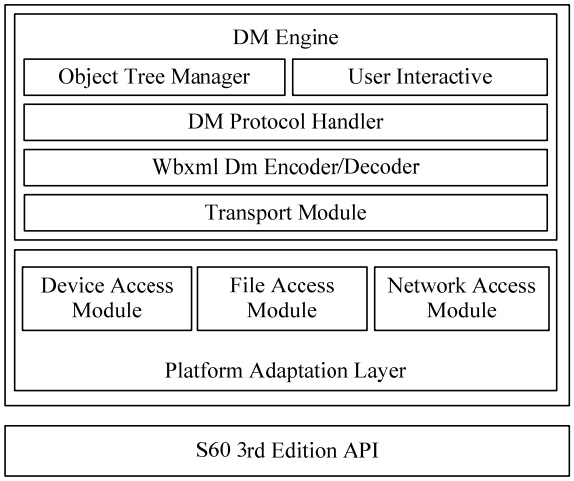


Fig 3. Structure of the RPDM DM client

The DM protocol handler performs the function of message exchange based on the DM protocol between DM server and DM client. The wireless binary extensible markup language (WBXML) DM encoder/decoder performs the function of encoding and decoding the SyncML message. The transport module performs the function of message transport through protocols such as HTTP, wireless session protocol (WSP) and OBEX. Platform adaptation layer, as the module to decrease the dependence on the platform of the DM client, is not limited to the specific terminal platform and allows the DM engine to access the platform application programming interface (API) in the same interface. It consists of the user interface (UI) control access module, device access module, file access module and network access module. The UI-control access module provides the UI-control provided at the platform. The device access module performs the function of inquiry and setup about the terminal's particular information. The file access module performs the function of file I/O about the file system of the terminal platform and the network access module provides the function to access the TCP/IP network through the API for network access by the platform.

3.1.1 Detailed System Composition of DM Engine

- Object tree manager: It manage the object tree information to satisfy the OMA SyncML device standardized objects and version 1.1.2. The object tree consists of the SyncML DM management object for setting the values of the SyncML DM client, the DevInfo management object for the terminal device information used in accessing to the DM server, and the DevDetail management object for managing other general device information.

- Through the multi-server, it allows several servers to access one terminal and performs the control of the terminal access authority of each server through the access control list.

  a) Not specified

  b) Background management action

  c) Informative management action

  d) User interactive management action

  In the case of *a)* and *b)*, no notification will be sent to the user. In the case of *c)*, the user will be notified through a communication window that the DM started, and in the case of *d)*, the DM session will be started after a confirmation key from user is inputted.

- DM protocol handler: This module performs the function of generating and sending/receiving the SyncML message between the DM server and DM client according to the DM protocol. The flow and the contents of message sending and receiving are performed according to SyncML DM protocol, version 1.1.2, SyncML representation protocol, DM usage and version 1.1.2 specification, and the main protocol command is as per the following Table 1.

- WBXML DM encoder/decoder: WBXML which means WAP binary XML content format is the technology to convert to binary format to increase the transmission rate when the XML content is transmitted in the wireless. The WBXML DM encoder/decoder encodes the SyncML message made at the high ranked DM protocol hander or decodes the SyncML message received from the server.

- Transport module: The transport module performs the function of transport protocol for sending and receiving the WBXML message to/from the message. It provides several kinds of protocols such as HTTP, WSP and OBEX, etc., and gives support the high ranked protocol to be acted upon regardless of the transport layer.

Table 1: DM protocol command

Command Type	***Command***	***Description***
Data Command Elements	*Add*	Adding the new node
	Copy	Copying the data of the source node to destination node
	Delete	Deleting node
	Replace	Replacing the value of the existing node
	Get	Getting the date of node
	Exec	Executing the service about the specific node (phone lock, etc.)
DataStore Command Elements	*Alert*	Being specifically used for notification of a like DM session requirement
	Result	Transmitting the result data on the request of a command (Get, etc.)
Process Flow Commands	*Atomic*	Performing a like command word inside of a sequence that command sets inside of an atomic let all subcommand be performed as a set unit or not be performed at all
	Sequence	All subcommand being operated sequentially in command sets inside of sequence

3.1.2 Platform Adaptation Layer

PAL is the layer to be reused to keep the independence of the DM engine against the platform and porting the DM engine to the various platforms. Various platforms such as binary runtime environment for wireless (BREW) and wireless internet platform for interoperability (WIPI), not only symbian, can be implemented as each module of PAL depending on the platform being changed can be suitable to the platform when the platform is changed. The DM engine, by exercising restraint in using the interface depending on the platform and using the interface of PAL, minimizes operations depending on the platform when it is used to porting to other platforms. PAL consists of the device access module (DAM), file access module (FAM) and network access module (NAM), and each module provides an independent interface by redefining the interfaces depending on the platform for each function.
The detailed structure of PAL is shown Fig. 4. The functions of each part are as followings.

- UI-control access module: It provides the common UI-control like the dialogue box required by the DM engine and application. By redefining the UI-control provided at the platform, it uses the UI-control independent of the platform at the DM engine and application.

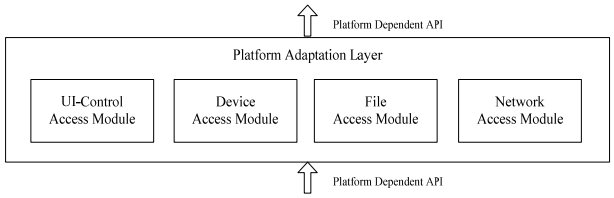


Fig 4. Detailed structure of PAL.

- DAM: It provides the interface to provide the device relevant information (manufacturer, international mobile equipment identity (IMEI), international mobile subscriber identity (IMSI), SW/HW version, OEM, model, language), and defines and provides the interface to access the device relevant information by using the device interface provided by the platform. It composes the device access module about the platform and uses it after redefining the device access module according to the platform. It consists of the object providing the information related to the device module, and the object providing the subscriber information. The module object of the device provides the information related to the product model such as manufacturer, IMEI, OEM name, model No. and HW/SW version, etc. The subscriber object of the device provides the information about the subscriber of the device and other setup. At present, the information about IMSI and language is being provided.

- FAM: It provides the interface to open/read/write /close the file. By using the interface of the file provided by symbian, it provides the interface to be able to use the file. It can inform the file access module about the various platforms.

- NAM: It sends and receives data, and controls access to the network. By redefining the network interface provided by the symbian platform, it provides the network interface independent of symbian to DM engine. It informs the network access module abut the various platform. It also defines the interfaces for the various network protocols and uses the interface for the TCP/IP protocol module. It can provide the various interfaces except for the TCP/IP protocol.

3.2 DM Server

The DM server implemented in this paper is divided into the server part, data part and HTTP Interface part, as are shown Fig. 5. The server part treats the SyncML protocol and DM protocol, and the data part stores user information terminal information and client request log, etc. The HTTP interface consists of the Servlet and XML-RPC

[10][11][12] interface which treats the request of the web admin and DM client.

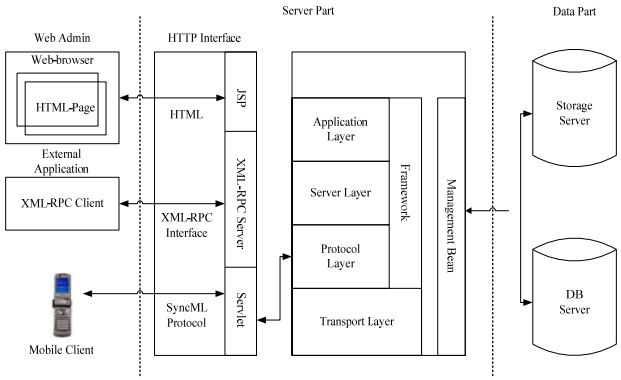


Fig 5. Structure of DM server

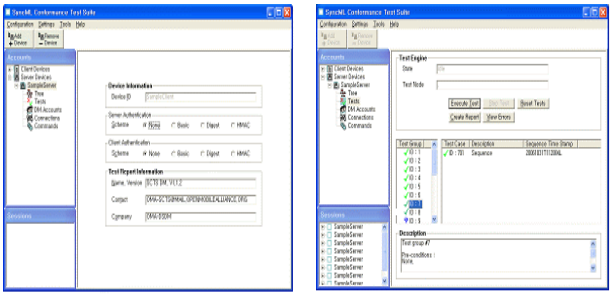
The HTTP interface sets up the DM server through the web browser or performs the role of monitoring. In order to performing such a role, it consists of a Servlet for treating the request of the DM client through the web admin and SyncML protocol embodied as JSP, and XML-RPC Server for treating the request of the mobile device DM from the external application server.

3.3 Linkage Test of DM Client and Server

Examination of the DM client used the emulator, 'S60 3rd Edition and For Maintenance Release (MR)' version included in S60 SDK. The system implemented for examination is performed on an emulator the same as with the general application, as are shown Fig 6.

Fig 6. Implementation of DM client on S60 emulator.

Generally in the DM session, the DM server sends a WAP push message to the mobile device through the push proxy gateway (PPG), and the terminal receiving the message reads the server initiation message and then links the session with the DM server. As this system implemented on an emulator, the DM session initiation was performed by an event through the user's menu. The DM server is used to add the sample server to SCTS DM of the examination tool about the mutual operation provided by OMA as in Fig. 7(a).



(a) (b)

Fig 7. Results of SCTS performing and server linkage test.

The server linage test is performed by using a test case in a test group of the DM server, after the DM server (SCTS DM) on the internet and the DM client performance on the emulator are linked through the IP network. The test case was composed to be able to test the various commands (Get, Add, and Replace, etc.) through each DM protocol. The results of the examination are as per Fig. 7(b) and the SCTS shows that each test case in the test group performed normally.

4. Experimental Results

This section examines whether each module operated normally, the mutual linkage and the functional test. The RPDM largely consists of the RPDM client, RPDM server, DM server and DM client terminal. It is linked by the internet till the RPDM web clients telecommunicate with the DM client terminal, and the DM client terminal is linked by using the GSM/GPRS network. Fig. 8 shows the general system composition of the RPDM and the 1st to 7th step of the progressing process since the request for control and information from the web client. Once there is a service request to the RPDM server by the RPDM web client through IOCP, the RPDM server sends the service request to the DM server through XML-RPC, and by using the request as a SMS hook through the SMS gateway, brings the DM client terminal information or controls the device.

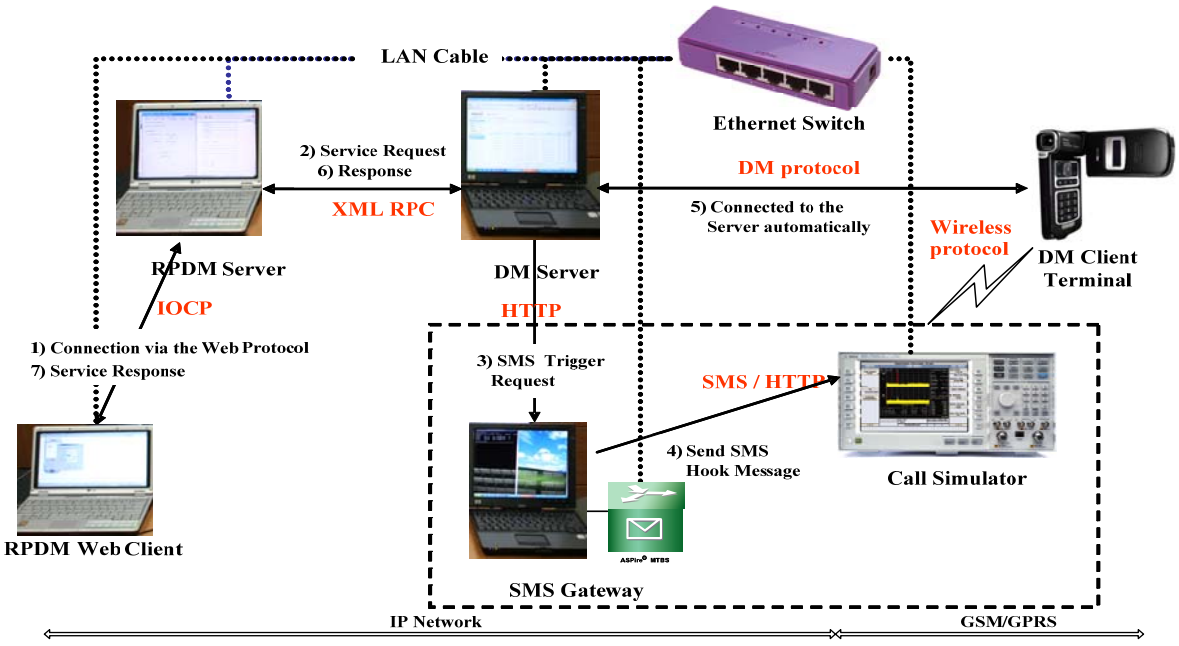


Fig 8. System composition of RPDM.

(a) (b) (c) (d)

Fig 9. DM solution in DM client terminal.

(a) (b) (c) (d)

Fig 10. The result screen of web client.

The scenario of the RPDM combination test is divided into three kinds of terminal information inquiry procedures such as a call log, SMS message and an inquiry of the address book, and two kinds of terminal setup change procedures such as a phone locking and confirmation, and a change of bell mode. The detailed procedures are as per the following.

- Terminal information inquiry procedures
  - *a) Confirmation of information (Call Log/SMS/ Address Book, etc.) at the terminal DM client being installed*
  - *b) Download and performance of the RPDM web client through web browser*
  - *c) Information inquiry request by RPDM web client*
  - *d) Confirmation of receipt contents by the RPDM web client*
  - *e) Compare the receipt contents at RPDM web client and the DM client terminal contents*
- Terminal setup change procedures
  - *a) Confirmation of lock setup information of the terminal DM client being installed*
  - *b) Download and performance of the RPDM web client*
  - *c) Confirmation of lock setup information of the present terminal through RPDM web client*
  - *d) Request for lock setup information change by the RPDM web client*
  - *e) Confirmation of lock setting at DM client terminal*
  - *f) Request for lock setup termination by the RPDM web client*
  - *g) Confirmation of lock setup termination of the DM client terminal*

Fig. 9 shows the screens of action standby, performance and close of the DM solution, which contains the processes such as the screens of DM standby as (a), DM session start as (b), DM server access as (c) and session close as (d). Fig. 10 shows the screen of the web client result. (a) shows the screen for the initial menu of the web client, (b) the detailed inquiry result, (c) the detailed inquiry result of the address book, and (d) the state of the phone lock and state change, respectively. Table 2 shows the development environment of the RPDM system.

Table 2: RPDM development environment

Component Name	***Development Environment***
RPDM Web Client	Notebook PC (Pentium 4)
	OS : Windows XP
	Platform : Web-base Active X
RPDM Server	Notebook PC (Pentium 4)
	OS : Windows XP
	Platform: DGIST platform developed by using the visual studio 6.0
	Two communication modules: 1. Between RPDM web client and RPDM server : Communication module which uses the IOCP 2. Between RPDM server and DM server : XML-RPC communication module developed by using the JDK
DM Server	DM server (Pentium 4)
	OS : Windows XP
	Communication module between DM server and DM client: communication module based on DM protocol
GSM/GPRS Call Simulator	E5515C wireless communication test set GSM/GPRS
DM Client Terminal	Symbian platform series60 (3rd edition) mobile phone (GSM/GPRS)

3. Conclusion

DM is an opened architecture not limited to the kinds of terminals, operating system, region and network system. It has been known the method to combine or substitute the equipment management technique limited to the specific networks and terminals. This paper designed the structure of an OMA DM client to adapt the existing terminal platform. Also, in order to cope with the smooth standardization of the application service, we designed and implemented the structure of an RPDM system. It also not only examined the basic protocol and interface of the results by using the tool of the SCTS DM, but also developed the remote control program for the mobile device through the web, the RPDM. This paper performed the combined linkage test by making a scenario after the

function of the remote information inquiry and control had been subdivided.

In the near future, we will be a necessity to upgrade the function of RPDM web client and server based on the web and to implement the stable system by inserting the web distribution and the security module.

References

[1] OMA “Open Mobile Alliance”, http://www.openmobileallia nce.org/.
[2] N.S. Park, K.W. Lee and H. Kim, "A middleware for supporting context-aware services in mobile and ubiquitous environment," ICMB 2005, July 2005, pp.694-697.
[3] Nokia Forum, http://forum.nokia.com/.
[4] symbian, http://www.symbian.com/.
[5] SCTS, http://sourceforge.net/projects/oma-scts
[6] WAP Forum, http://www.wapforum.org/.
[7] B. Steinke, “The utilisation of peer-to-peer techniques in autonomous mobile phone device management; Efficient framework for advanced device self-management,” Proceedings of the 10the IEEE Singapore International Conference on Communication systems, 2006, pp. 1–5.
[8] S. Bernd, "Autonomics and Self-Management extending Advanced Device Management," TENCON 2006, Nov. 2006, pp.1-4.
[9] M. Jun, J.X. Liao, X.M. Zhu, C. Wang and Y.T. Zhang, "Mobile device Capability Management for Services Enabling," ICWMC '06, July 2006, pp.74-79.
[10] S. Dissanalike, P. Wijkman, and M. Wijkman, “Utilizing XML-RPC or SOAP on an embedded system,” Proceedings of the 24th International Conference on Distributed Computing Systems Workshops, 2004, pp.438–440.
[11] M. Krajecki, O. Flauzac, P.-P. Merel, “Focus on the communication scheme in the middleware CONFIIT using XML-RPC,” Proceedings of the 18th International Parallel and Distributed Processing Symposium,2004, pp. 26-30.
[12] X.-R. H. Page. Xml-rpc specification. Internet, 2003. http://www.xmlrpc.com.

HyunKi Ryu received the M.S. degree in Computer Engineering from Kyoungbook National University, DaeGu, Korea. He currently proceeding the Ph.D. degree in Computer Engineering from Kyoungbook National University and joint work DGIST In 2005. His research interests include Multimedia network, network QoS, Wireless Sensor network.

SungRae Cho received the M.S. degrees in Computer engineering from Yeungnam University, Korea, in 2000. Since 2005, he has been with the Hybrid System Research Team in DGIST (DaeGu GyeongBuk Institute of Science & Technology) in DaeGu, Korea. He is currently a Senior Research Engineer of DGIST. His current research interests are Embedded Software Architecutre for Vehicle, Mobile Terminial Software

Shiquan Piao received the B.E. degree in Electrical Engineering from Peking University, Bejing China, in 1999, and the M.S. degree in Information and Communication Engineering from Yeungnam University, Korea, in 2002. From 2002 to 2005, he has worked towards his Ph.D degree in Information and Communication Engineering, at Yeungnam University. Since 2005, he has been a research engineer at Daegu Gyeongbuk Institute of Science & Technology (DGIST).

JongWuk Son received his M.S. degree in Electrical Engineering from Korea Advanced Institute of Science Technology (KAIST) in 2006. Currently he is working for Daegu Gyeongbuk Institute of Science & Technology (DGIST). His research interests include reusable embedded software and automotive system.

WooYoung Jung received the B.S. degree fro Seoul National University, Korea, in 1984, and the M.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), in 1986, and 1991, respectively. He is currently a principal engineer in Daegu Gyeongbuk Institute of Science and Technology (DGIST). His research interests include embedded Software for the vehicle, and intelligent vehicle based on radar and vision.