International Journal of Control and Automation
Vol. 7, No. 11 (2014), pp. 341-354
http://dx.doi.org/10.14257/ijca.2014.7.11.34

# Development of Set-Top Box-Typed Hardware Systems with Internet of Things-Based Embedded Middleware

Hyu-Chan Kim[1], Mi-Jung Kim[2] and Hyun-Jeong Shin[3]

[1]*Dept.of Digital Contents, Cheju Halla University, South Korea*
[2]*Dept.of Health Administration, Cheju Halla University, South Korea*
[3]*School of IT Convergence Engineering, Shinhan University, South Korea*
*khc@chu.ac.kr, mjkim@chu.ac.kr, hjshin@shinhan.ac.kr*

***Abstract***

*This paper developed the set-top box-typed hardware terminal which process data collected from heterogeneous sensors and RFID readers. The developed terminal can reduce expenses because it can process tag data of heterogeneous sensors and RFID readers as the integrated terminal which collect and process data and be provided with users by the method which support several client applications with requirements of various tag data.*

*This study developed the hardware terminal that support of common data formats which are necessary for sophistication of data processing modules of various sensors, stabilization of communication modules, and status management was improved. It extracted the core function of the middleware which are commonly necessary for various typed USN application services.*

***Keywords:*** *Heterogeneous sensors, RFID, Middleware, Embedded middleware, Set-top box*

## 1. Introduction

Interoperability technology in home network means a technology that enables mutual recognition control monitoring between arbitrary home network devices which exist in the home. At present, with regard to home network, various sensor networks such as entertainment network and appliances control network coexist with heterogeneous networks and devices in the environment where various technologies thrive. We need a technology that can integrate these in an actual condition of an increase in needs for providing interoperability between devices based on these heterogeneous networks and middleware [1].

Middleware controls various sensors and collects data, using various protocols from sensor. Besides, it shall extract meaningful information or information in the form easy for application to use from collected and unprocessed data, and shall basically carry out a function of transmitting the information to application service [2, 3].

As it abstracts readers with the concept of logic readers, it does not make clients get involved in use of hardware in regard to specific logic positions. Accordingly, as hardware does not affect client application even though it is changed, it has the cost saving effect because it can process heterogeneous sensors and RFID tag data as the integrated terminal which collects and process data.

RFID/USN middleware is designed to be compatible with TAP models as the software which were developed based on Java. It is provided for users by the method which support several client applications which can have requirements of various tag data. It can be operated in any environment ranging from ROI which should be made under the necessity of companies including exact data and activity data to RFID facilities.

ISSN: 2005-4297 IJCA

It was designed to process road balancing readers' processing and handle data services automatically by sharing placement and expansion among several terminals with others in a site by composing it.

This paper studied relevant studies and technologies in Chapter 2, composed the proposed system in Chapter 3, and realized the proposed system in Chapter. And this study's conclusion and next research direction are briefly described in Chapter 5.

## 2. Related Studies and Technologies

### 2.1. RFID/USN Software Technology

We need a middleware function that can connect RFID/USN hardware to application or enterprise system, that is to say, RFID/USN software technology in order to easily establish an ubiquitous application service[4].

RFID/USN middleware shall have ability to filter a large amount of data collected in a large number of heterogeneous RFID/sensor devices, and to process event data, and then to summarize meaningful information, and to transmit the information to a service system[4].

The following is required in order that RFID/USN technology is effectively utilized in application field. We shall organically establish RFID tags attached to things in order to identify each thing, a RFID reader to recognize these, and RFID/USN software that can provide useful information service to an application after processing the data collected from sensor devices [5].

That is, reduction of expenses to collect data automatically by introducing the RFID/USN technologies and effective accomplishment of purposes including improvement of existing business processes and creation of new services cannot be solved just by introducing and installing leader devices and sensor devices with excellent performance or extending the existing enterprise systems [6].

Now, for building RFID/USN systems and developing their technologies, the closed types which restrictively share information in specific fields focusing on manual RFID technologies targeting logistics and distribution services have been built. As a result, RFID/USN middleware has always and newly been designed and developed as it was specialized in specific companies and partial fields. Therefore, it just concentrates on the basic function such as approach to data through RFID/USN devices and storage and processing of data collected from devices and it is very subordinate to specific hardware and applications[7].

To solve these problems, RFID/USN middleware should guarantee interoperability in various types of device interface, interlocking with various data and networks, and several application platforms [8].

### 2.2. RFID/USN Middleware

Sensor Edge Server[9] of Oracle is a sensor-based service integration platform which supports functions such as data collection, event handling, and data dispatching. It is possible to extract data which an application requires by using a filter in advance. And in case of intending to process more complicated data, it is possible to prepare a logic filter personally.

TagsWare[10] of CapTech consists of links to send RFID tag data to applications, drivers as a standard interface between RFID devices and applications, and base components which enable application to use links and drivers.

Sun's Java System RFID Software [11] processes sensor data stream from various sensors in a event manager and consists of a reader adaptor, a filter, a logger, and an enterprise gateway. It can ask question about delta and smoothing, get EPC data values which satisfy specific mask

conditions by connecting filters to each other, and supports the tools which can develop user-defined filters.

ETRI's SSi(Software System Infrastructure) platform is the software which manages heterogeneous RFID reader devices unitarily and carries out the functions to deliver processed information by analyzing information of things by individual article unit collected from them to RFID application services which need them. SSI platform manages devices based on internal and external standards in the heterogeneous RFID infrastructure environment of the various application fields and provides data processing. And its functions were more reinforced to recompose the structures flexibly according to various business requirements because it is easy to reuse and extend them as it was structuralized as the component-based subsystems[12].

To provide various ubiquitous services, RFID/USN middleware should guarantee interoperability in various types of device interface, interlocking with various data and networks, and several application platforms[8].

## 3. Design of Suggested System

### 3.1. System Configuration

Data provision servers provide internet information for RealTime Updator by processing them first to meet trend services or directly do them for trend DB. Real Time Updater secondly process the data from the data provision servers, insert them into the trend DB or keep it up-to-date by running in the data provision servers' process if necessary.

Integrated service managers directly get involved in Real Time Update and trend DB or run in or manage the process with set values.

User services provide the services to analyze trends focusing on DB.

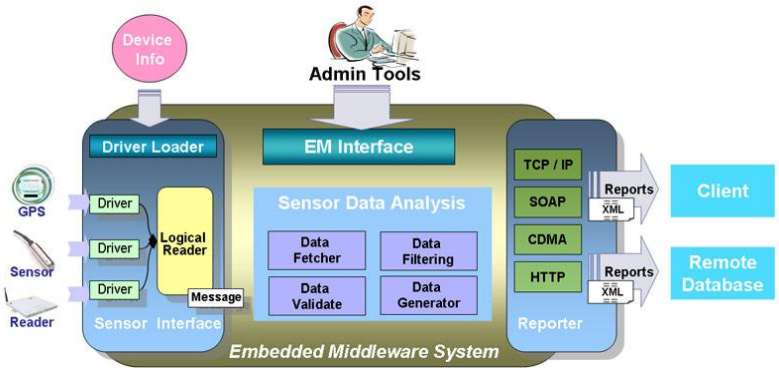



**Figure 1. Embedded Middleware System Configuration**

(1) Develop GUI-typed sensors and RFID reader management modules.

- Register, delete, and change readers to manage physical devices.
- Set models and communication standards of physical devices and manage them by naming them.
- Map physical reader driver class.

(2) Develop modules to manage (register, delete, and change) logical readers by UI.

- Group one or more physical readers to be a common format and manage logical readers by creating them.
- Set data buffers and manage logical readers by naming them.

(3) Develop modules to manage (register, delete, and change) reporters by UI.

- Map one or more logical readers and set the method to transmit the collected data to remote areas.
- Set report cycles and map report driver class.

# 4. System Implementation

## 4.1. Major Interface Window

### 4.1.1. Development of Heterogeneous Sensor and RFID Reader Control Module in the form of GUI

First of all, a reader shall be physically connected in order to register the reader. Functions are implemented so that reader can be registered in the middleware in the following order as shown in Figure 2 and Figure 3, Figure 4.

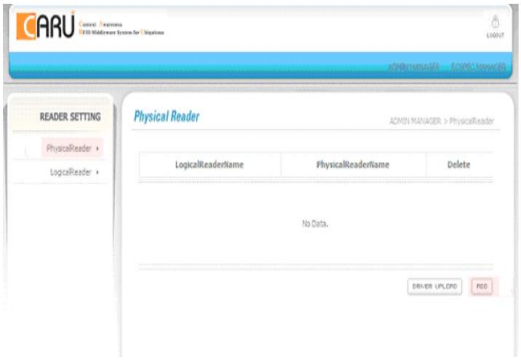



Figure 2. Execute the Registration of Reader

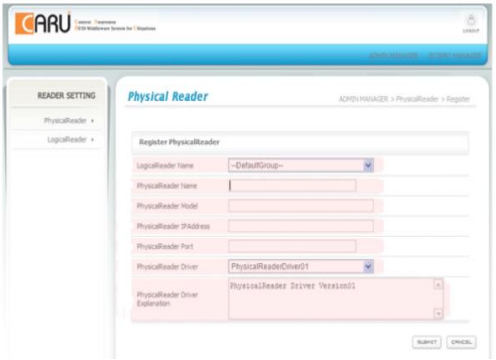



Figure 3. Reader Registration /Input


Figure 4. Reader Registration Confirmation

### 4.1.2. Developing the Dynamic Driving System of Heterogeneous RFDI/USN Sensor

- We implemented a function of easily registering and controlling a dynamic heterogeneous reader so that a physical reader could be registered as a logical reader for perfect linkage with middleware after

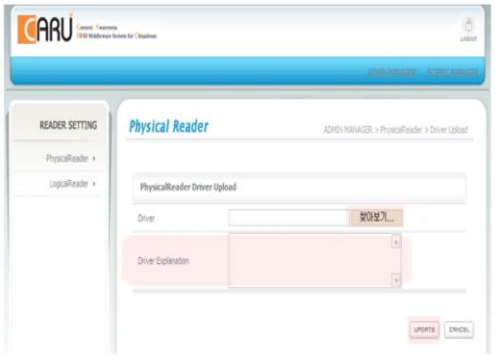



Figure 5. Physical Reader Search

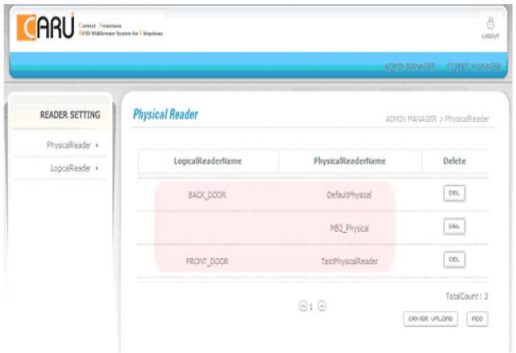



Figure 6. Physical Reader Search Results

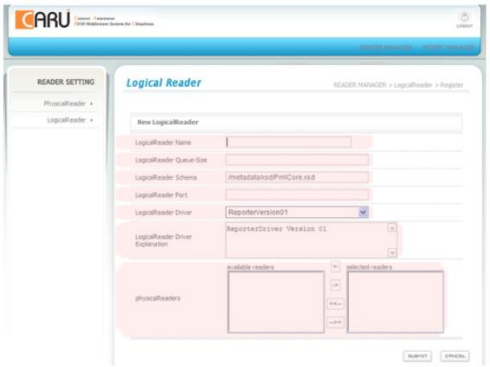



Figure 7. Logical Reader Registration

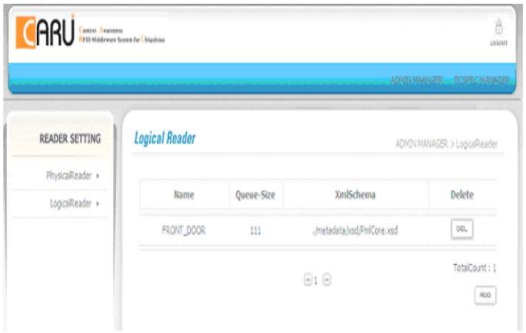



Figure 8. Reader Registration Confirm

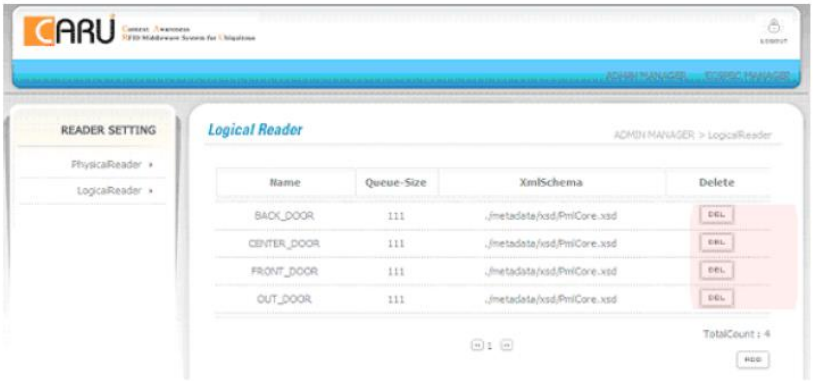



Figure 9. Registration Status of Various Readers

### 4.1.3. Data Collection Standard Protocol and Processing Module Development

- We decided that collected data standard should be XML format, and standardized data, and implemented a function of defining the collected information so as to be standard.

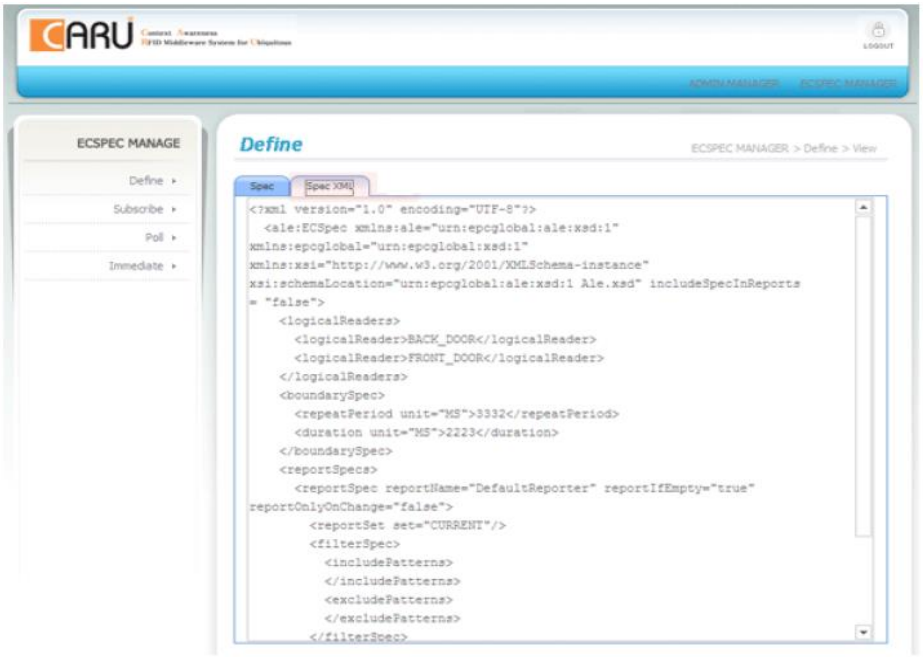
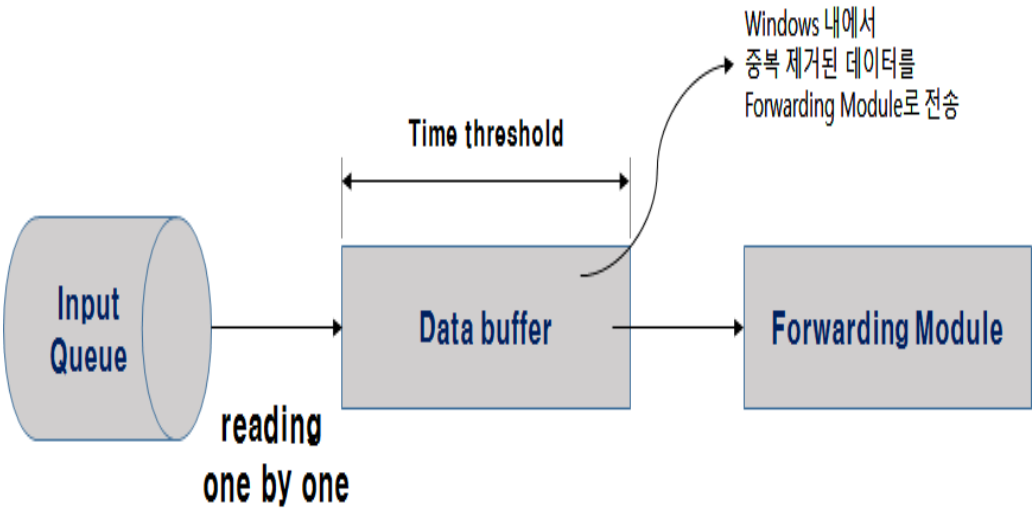



Figure 10. Definition of Standards to Collect Data

- The reader interface filters error data by checking data models by vendor/model of readers basically and remove the tag data which were filtered like this by using a overlapping filter first. And deliver the data which are delivered to applications by filtering the data collected from various logical readers doubly by using users' setting time again before delivering them.



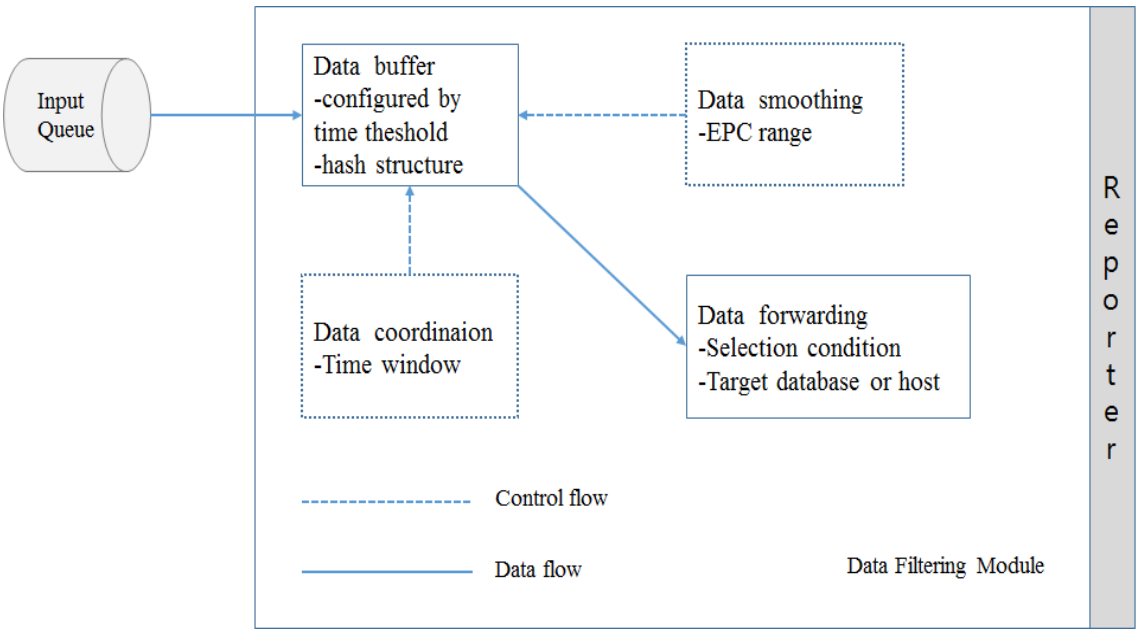



**Figure 11. Sliding Window**

- This function in advance define the collected RFID data to be utilized in middleware core as various information. Can realize the function to use the collected information that you want easily in various applications.


**Figure 12. Select the Function of Data Processing**

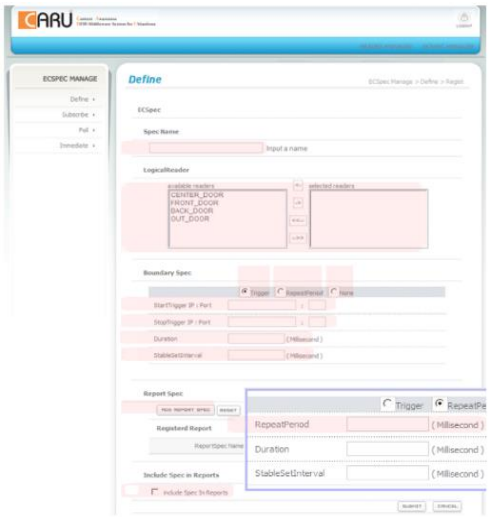



**Figure 13. Definition 1 of Data Processing**

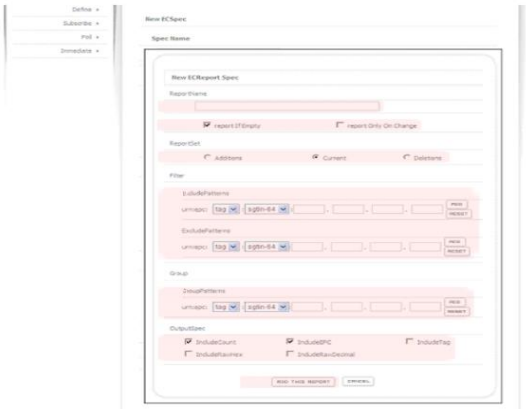


Figure 14. Definition 2 of Data Processing

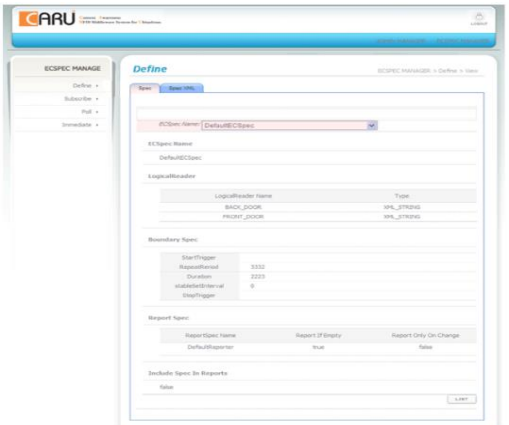


Figure 15. Check the Definition of Data Processing.

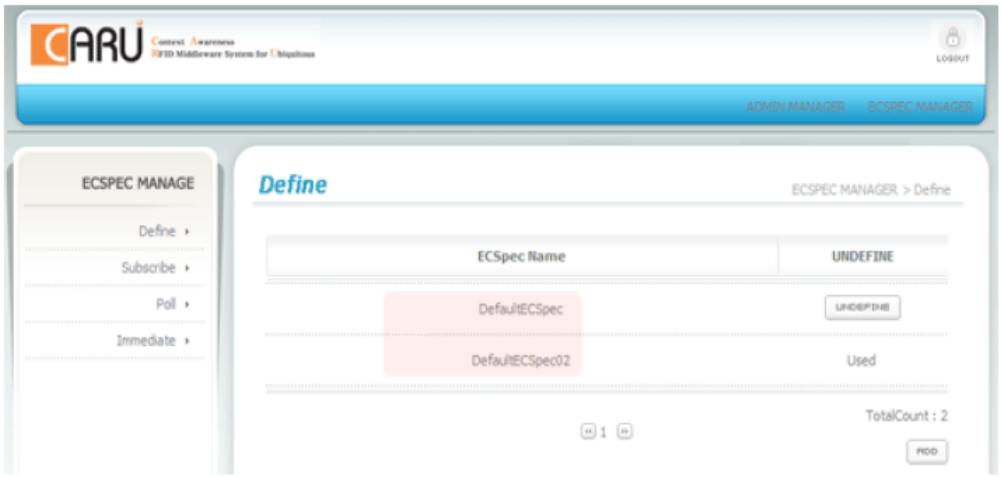



Figure 16. The List of Data Processing

### 4.2. Design and make the Onboard Terminal

(1) Select the model with CPU and hard disk that memory expansion is possible
- Make various types of processing possible even in the systems which need complex processing without adding separate hardware.

(2) Installed RFID/USN Embedded Middleware
- Install Onboard RFID/USN middleware in the set-top box-typed embedded device.
- Was designed to process the existing server client-typed RFID & sensor data processing in one unit of hardware.

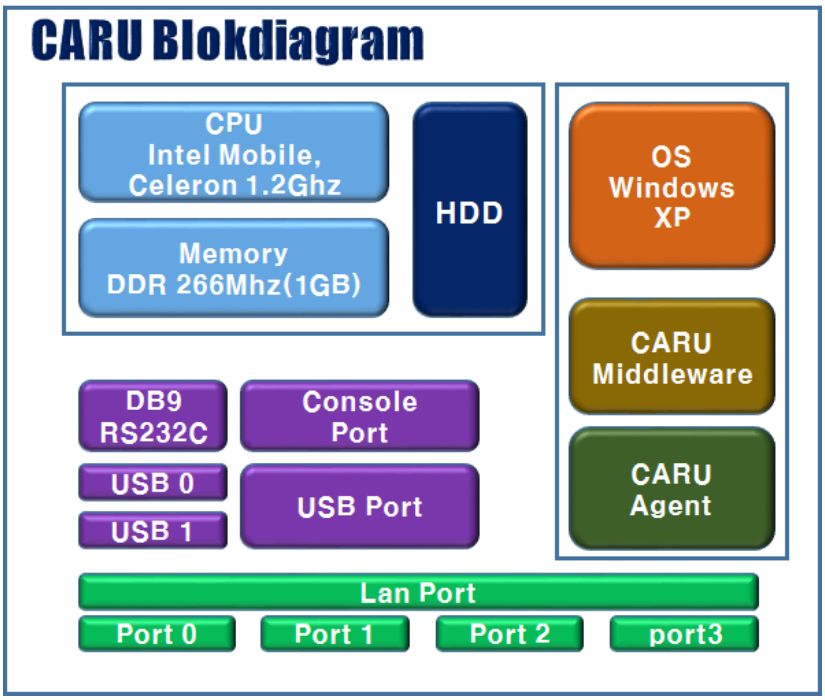



**Figure 17. Hardware Block Diagram**

(3) Design various types of external interface (Ethernet, RS-232c, USB, PCI)
- Compose them focusing on the communication interface which are most used in sensors and RFID devices
- Need hardware with the communication interface which can accommodate external interface of these sensors to connect various sensors and readers.
- Design to remove the lot of data in networks by using TAP Ethernet ports

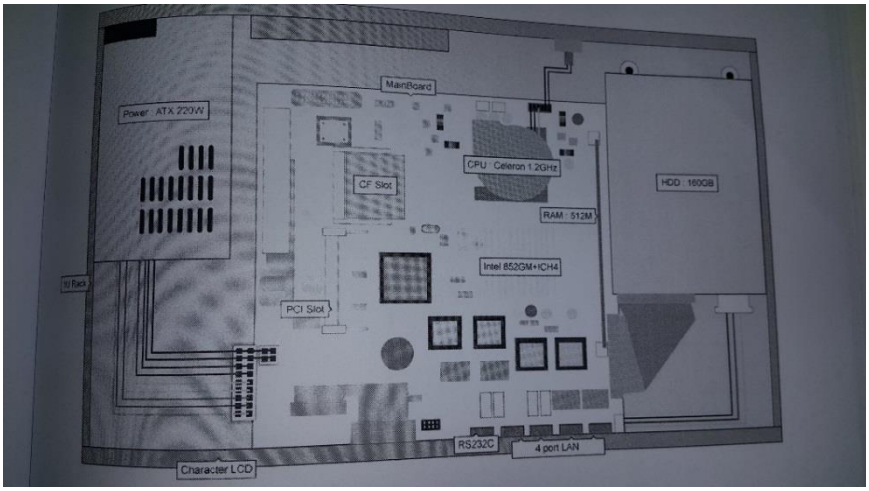



**Figure 18. The Design Plan of the Terminal**

Figure 19. The Inside of the Terminal


Figure 20. The Top of the Terminal


Figure 21. The Front of the Terminal

Figure 22. The Back of the Terminal

### 4.3. Middleware Installation and Operation

#### 4.3.1. System Specification

PCs which are higher than the following specifications should be used to run CARU middleware and CARU Web Application.

**Table 1. System Minimum Specification**

| | |
|---|---|
| O/S | Windows XP, Linux 2.6, etc |
| DK/JRE | JDK 1.5 / JRE 1.5 |
| CPU | Pentium 1.0 Ghz over |
| Memory | 521Mbyte over |
| Web Container | Jetty6.1.6 |

#### 4.3.2. CARU Middleware Installation

##### 4.3.2.1. The Directory Structure

If you unzip the distributed archive file, caru_middleware.zip, the directory structure is created like [Figure 23].

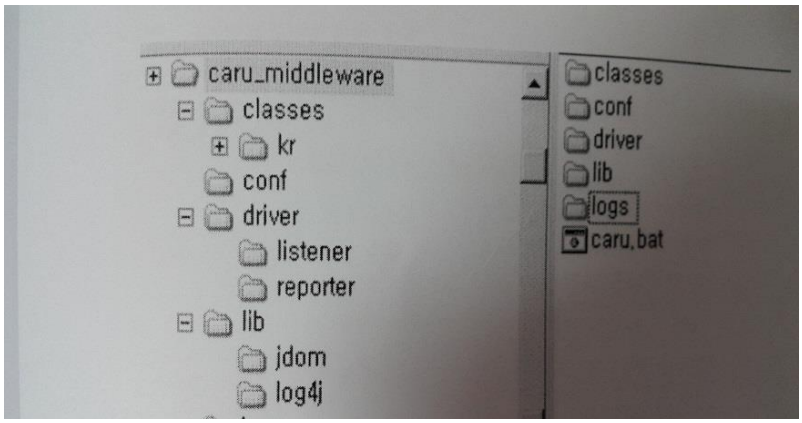



**Figure 23. CARU Middleware Directory Structure**

If caru.bat runs in [Figure 23], CARU middleware is performed and there are the configuration files which are necessary for running the system in the <conf> folder.

#### 4.3.2.2. The CARU Middleware System Configuration

**Table 2. CARU -service.xml File**

```
<?xml version=”1.0” encoding=”utf-8”?>
< CARU -Service>
<!-Logical Reader, Physical Reader management service - ->
<service name=”readerManager”
class=”kr.co.javainfo.middleware.app.background.imp1.ReaderManagerService” />
<!-CARU –em.xml management service- ->
<service name=”xmlManager”
class=”kr.co.javainfo.middleware.app.background.imp1.EdgeXmlFileManagerService” />
<!- CARU Web Application and communications services - ->
<service name=”controller”
class=”kr.co.javainfo.middleware.app.background.imp1.ControllerManagerService” />
<!- CARU web application and port number to communicate - ->
<prpperty name=”port” value=”8089” />
</service>
</ CARU -Service>
```

Information on logic readers and physical readers are stored in caru-em.xml. If there are those files, you shall create them newly.

The several services used for operating CARU middleware are set up in caru-service.xml. And the following details are the contents for basic operation.

#### 4.3.2.3. CARU Middleware Operation

If % caru_middleware path%/caru.bat runs, CARU middleware is operated. The screen that CARU middleware is normally operated is shown in [Figure 24].

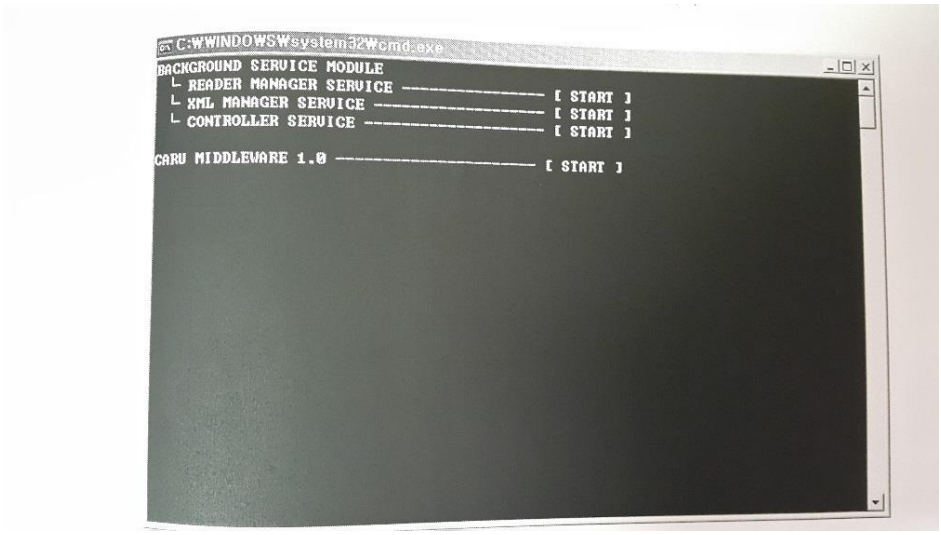



**Figure 24. CARU Middleware Operation Screen**

Please be warned, CARU middleware and CATU Web Application can normally be operated when they are installed in the same PCs.

#### 4.3.2.4. CARU Web Application Installation

If you unzip the distributed archive file, caru_Web_application.zip, the directory structure is created like [Figure 25].

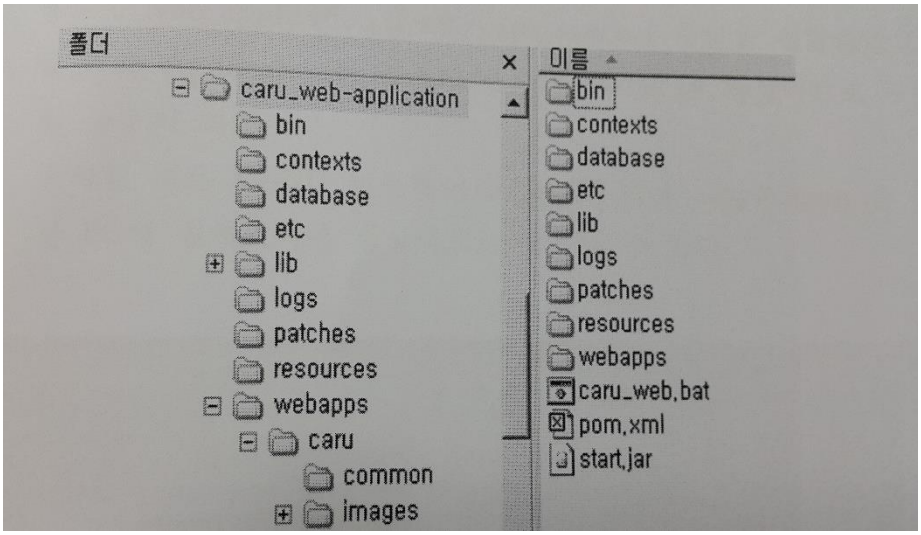



**Figure 25. CARU Middleware Web Application Directory Structure**

#### 4.3.2.5. CARU Web Application Operation

If %caru_web_application path%/caru.bat runs, CARU Web Application is operated. The screen that CARU Web application is normally operated is shown in [Figure 26].

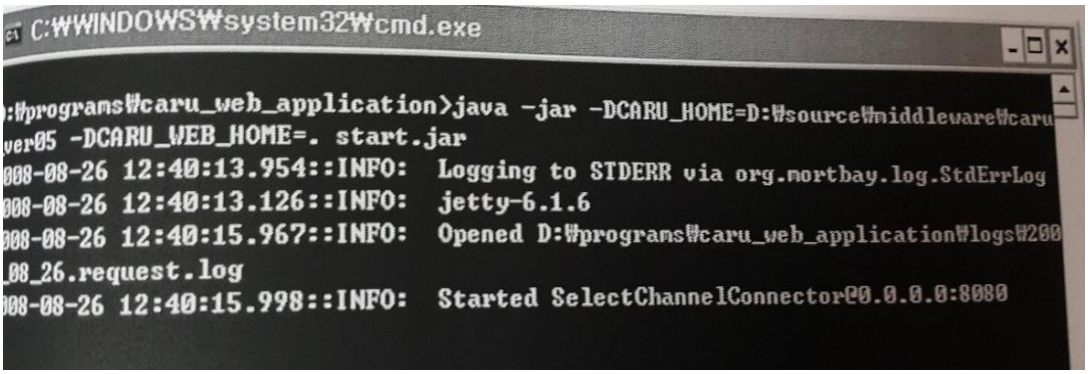



**Figure 26. CARU Web Application Operation Screen**

#### 4.3.2.6. Interoperability Test

If CARU middleware tries to run, Please operate the Web management program by execute caru_edge.bat first and then caru_web.bat.

The operation status was checked by applying it to the hotel service actually and taking the interoperability test with UHF readers.

## 5. Conclusions

This system can immediately apply accumulated and filtered data to client by reporting them with the formats defined by users regardless of creation of data collected from heterogeneous sensors and RFID readers in which hardware and the processing methods.

The heterogeneous sensor set-top box hardware terminal developed in this paper can reduce expenses because it can process tag data of heterogeneous sensors and RFID tag readers. And it can process big tag data by connecting several units of RFID middleware to each other in parallel and continue to handle extension of system infrastructure and maintenance of system stability by reducing loads which are applied to the system.

This paper developed heterogeneous sensor set-top box system which processed data collected in heterogeneous sensor and RFID reader as the development of information transmission media relating to RFID/USN. It was made possible to promptly apply data collected in heterogeneous sensor and RFID reader to a client by reporting the accumulated and filtered data in a user-defined format in this system regardless of what hardware generated the data and how the data was processed.

This study supported common data formats which are necessary for sophistication of data processing modules of various sensors, stabilization of communication modules, and status management. And it is expected that large retail stores, warehouses, distribution centers, manufacturing plants, food disposal centers, airports, military bases, hospitals, and other facilities will use this system usefully.

## Acknowledgements

This paper is a revised and expanded version of a paper entitled “Design and implementation of heterogeneous sensor set-top box system using embedded middleware” presented at NGCIT2014, Hochiminh (Vietnam), 24-26, Oct., 2014.

## References

[1] Intel, “Building the Digital Supply Chain: An Intel Perspective,” Intel Corp., **(2005)**.
[2] M. Kim, I. You and H. Choi, “Design and Implementation of the Linux Interface Middleware to Support Time Determinism”, Proceedings of The Korean Institute of Information Scientists and Engineers Fall Conference KCC2006, vol. 33, no. 2(A), **(2006)**, pp. 313-317.

[3] S. Lee and C. Lee, "Design and Implementation of the Portable Middleware on Realtime Operating System QNX", Proceedings of The Korean Institute of Information Scientists and Engineers Fall Conference KCC2006, vol. 33, no. 2 (A), **(2006)**, pp. 454-458.
[4] S. Hong, T. Jeong, Y. Kim and Y. Lee, "Trends of RFID Middleware Technology and Its Applications", ERTI Electronics and Telecommunications Trends, vol. 20, no. 3, **(2005)**.
[5] O. Sewon, J. Park and Y. Lee, "RFID SW Technology and Standardization Trend", Journal of KICS, vol. 24, no. 7, **(2007)**.
[6] N. D. Evans, "Middleware Is the Key to RFID", RFID Journal, **(2004)** April.
[7] H. Kim, W. Ko, S. Ko and M. Yang, "Design and implementation of a dynamic management method for heterogeneous RFID/USN devices", Journal of KSCI, vol. 14, no. 7, **(2009)**, pp. 144-145.
[8] J. Seong and D. Kim, "EPC Sensor Network for intrastructure which combined RFID with USN", Journal of KICS, vol. 13, no. 12, **(2006)**.
[9] Enterprise Information Architecture for RFID and Sensor-Based Services, Oracle White Paper, **(2006)**.
[10] TagsWare, Agile RFID Solution, http://www.tagsware.com, **(2009)**.
[11] A. Gupta and M. Srivastava, "Developing Auto-ID Solutions using Sun Java System RFID Software", **(2004)**.
[12] http:///www.boannews.com/media/view.asp?idx=24028&kind=5, **(2010)**.

## Authors

**Hyu-Chan, Kim**
1999.3 ~ 2014.10 Cheju Halla University Dept.of Digital Contents, Professor

**Mi-Jung, Kim**
2009.9 ~ 2014.10 Cheju Halla University Dept.of Health Administration, Professor

**Hyun-Jeong, Shin**
1993.3 ~ 2014.10 ShinHan University School of Convergence Engineering/Computer Science & Engineering, Ph.D. Professor