Delivery and Storage Architecture for sensing information using SNMP

Hongseok Jang[†], *Kugsang Jeong*[††] *and, Deokjai Choi*[††]

{jwang, handeum}@iat.chonnam.ac.kr *dchoi@chonnam.ac.kr*

Chonnam National University, 300 Yongbong-dong, Buk-gu, Gwangju, Korea

Summary

Widespread use of sensor and micro processor leads to increasing concerns about context aware computing. Many researches about context aware computing are carried out around the world. Among them, Context-Toolkit and Semantic Space provide separation of concerns between sensor and application. They support making application development easier. However, they have a problem for lack of simplicity in communication, compatibility and flexibility in building systems. To solve it, we propose one delivery and storage structure for sensing information using standardized simple network management protocol which is useful to store and manage information in order to transmit and save sensing information. We also verify that this architecture is efficient in wireless sensor network to deliver and store environmental information through an implementation of a SNMP agent.

Key words:

sensing information management, sensor network management, SNMP

1. Introduction

Ubiquitous computing is an environment where various computers are melted in and connected to humans, objects and surroundings to allow computing anytime, anywhere. In order to realize such ubiquitous computing environment, two technology areas are required. One is the sensing technology where information on user and surrounding environment are sensed and provided, and the other is the context aware computing technology where such information are properly provided to users.

Different from previous application systems where users directly send orders to provide the same service to all users, context aware computing secures differentiated and automated service according to the users' status. Currently, context aware computing technology is considered as the technology realizing ubiquitous computing environment.

Some of the representative researches related to context aware technology in and out of the country includes Context Toolkit [1] of Georgia Tech, US, RCSM (Reconfigurable Context Sensitive Middleware) project [2] of Arizona State University, CIS (Contextual Information Service) [3] of CMU and Semantic Space [4] of NUS. Among these, Context-Toolkit and Semantic Space use a medium to manage context information between the sensor and application in order to resolve the problem of dependency. However, to let the application use context information, the medium in between must be newly established, or a complex connection must be made with the previous medium.

By developing the function of collecting/analyzing sensor's information and transmitting them to the application as a reusable component, Context-Toolkit supports the convenient development of related context aware applications. Also, by separating the sensor and application, the developer doesn't need to know about the sensor's platform, having the merit of 'sensor abstraction.' The inputting part of the sensor is composed with a component called 'widget.' When the application needs sensor information, it can retrieve it any time. However, even though Context-Toolkit promoted the convenience in development, it hasn't been commonly used because of the separation between the application and platform developer. Also, it had a demerit of making developers not familiar with the application or sensor platform have a hard time working with it.

Furthermore, in Context-Toolkit, there was a problem of increased connection points for communications. An example for this can be the understanding of overall resources necessary to transmit or save information. In order to process sensing data, Context-Toolkit must execute resource discovery to understand the overall resources. For such execution, the widget must understand various connection points to other components, such as the aggregator, interpreter, etc.

The widget of Context-Toolkit has a communication module and data-processing module, which are vulnerable to the application and communication protocol of Context-Toolkit. Since a fixed communication type must be followed in Context-Toolkit, flexibility in development was decreased.

In Semantic Space, a Context Wrapper, which is like the widget in Context-Toolkit, transmits sensing information to the application or aggregator. To add or delete sensors, Context Wrapper is added to or deleted from a context aware system called smart space through UPnP service. In here also, the problems of resource discovery and communication type compatibility exist, just like in Context-Toolkit. Also, different types of

Manuscript received April 5, 2006.

Manuscript revised April 30 , 2006.

context might be created from different developers, since a standard information structure is not used. Finally, there is no query or event function because Context Wrapper merely turns sensing value into a context.

Thus, by establishing a sensor platform using the transmitting and saving functions of the well-known network management protocol, SNMP (Simple Network Management Protocol) [5], we have tried to improve the problems in Context-Toolkit and Semantic Space.

SNMP is a protocol used in network management, and its application uses agent tools by passing through network elements. To manage specific resources given to the devices composing the network, the agent uses information of MIB (Management Information Base).

The following is the advantages of SNMP in transmitting/saving sensing information.

First, as a standardized protocol used for a long time in network management, it has high compatibility and various related tools. Since it uses a common protocol between the sensor and middleware where sensing information is processed, dependability to sensor's environment can be eliminated. [6] This allows flexible adding and deleting of sensors.

Second, a more simplified communication is available since the only connection point, MIB view, needs to be understood in SNMP's agent. In SNMP, only the agent address and information type are necessary when retrieving information.

Third, compatibility in communication is secured since a standardized SNMP communication method is used in transmitting information. In other words, mutual information transmission is compatible by using many previous SNMP tools.

Forth, flexible transmission of sensing information is available through various SNMP communication mechanisms with interrupt and falling. Having an event trap function based on threshold which is similar with the event processing of sensor, it is easy to receive information on sensor errors or user preference environment.

Fifth, efficient information saving and searching is available because only the environment information significant to users is saved.

Finally, SNMP manager can easily access the sensor information saved and managed in MIB of the sensor nod. This allows all applications using SNMP protocol to easily access MIB of the sensor and retrieve information, providing flexible development as well as possibility of a new application development. Sensor developers can simply disclose the MIB value so that middleware or application developers can change the receiving MIB value accordingly regardless of sensor API. In Context-Toolkit, many changes must be made on the widget program when sensors are changed. However, when using SNMP, application can be made by simply changing the MIB value.

By securing the transparency of sensor through servicing wireless sensing information received by the host PC with a SNMP agent, and by managing the sensing information of wireless network through MIB, this paper shows the possibility of flexible transmission and saving of sensing information through SNMP.

Following the introduction in Chapter 1, we have described the architecture points and realization in Chapter 2 and 3. Finally, in Chapter 4, we will suggest our conclusion and future studies.

2. Delivery and Storage Architecture

2.1 Hardware and Software Configuration

The composition of the architecture which proposes in this paper is same below [Figure1].

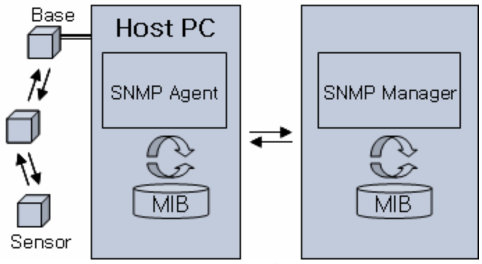


Fig. 1 Composition of architecture

SNMP agent of host PC, which collects and manages environment sensed information, achieves server's role. And SNMP manager of outside systems, which can approach to information, approaches to SNMP agent and achieves client's role.

SNMP agent module of server (host PC) side is consisted of SNMP agent, data transmittance, and MIB. SNMP agent manages and controls sensor. Data transmittance passes environment information to manager side periodically. MIB store environment information of sensor.

SNMP manager module of client side asks environment information from the SNMP agent. SNMP manager module is consisted of SNMP manager and MIB. The first is used in administrable and the second can do save information from agent.

In this paper, I compose server side's system (agent) to confirm that SNMP agent could serve environment information from wireless environments to client side's system (manager).

Among sensor management architectures, the agent part, which reads the values from sensors, is consisted of one or many sensor modules and only one host system. [Figure 2] shows composition of agent part.

Sensor modules pass sensed information to a base sensor module or other sensor modules through wireless communications using 2.4GHz, IEEE 802.15 .4 methods.

The host system do communication using the UART as RS 232C(Micaz) or USB (Telos) etc. to the base sensor module.

Sensor modules install application programs managed by TinyOS. In this paper, I measure temperature, illumination, internal voltage etc. by installing a OscilloscopeRF application program. The sensor modules sense from environments and deliver sensed information to other sensor modules or the base one.

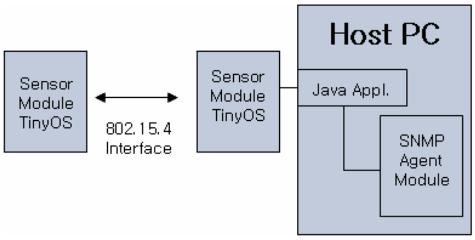


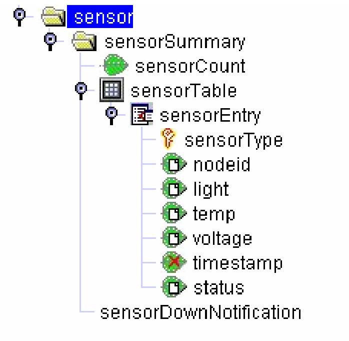
Fig. 2 Composition of Agent part

The composition of host PC side is discussed below. The base sensor module acting like a gateway passes the sensed information acquired by the UART using Java interface of TinyDB, MoteIF etc. to the SNMP agent application module.

I composed the system that passes the sensed information using general MoteIF application program here.

MIB information has four kinds of environment information (ID of node, temperature, illumination, internal voltage) reading from sensors.

Below [Figure3] is MIB structure applied in this architecture.



```
sensor
  sensorSummary
    sensorCount
  sensorTable
    sensorEntry
      sensorType
      nodeid
      light
      temp
      voltage
      timestamp
      status
  sensorDownNotification
```

Fig. 3 Applied MIB structure

2.2 Action process

SNMP agent stores and manages environment information as well as watches and controls states of sensors. SNMP manager manages sensor with information received from SNMP agents. Trap operator is used as notification method for the sensor network management. When the specific value designated by user is perceived from the sensors, the manager will be notified from agents using Trap messages.

The method of different information transmission also is the case of using Get operator. If user asks query about the value that he wants to know by using manager program, the agent will respond to manager's question. Information can be collected from various agents by these information transmission methods.

We need to alter and control the pre-setting values because of lots environment changes. To satisfy this, SET operator is used as the method of information control. The values of agent's MIB are changed easily in manager side and are readjusted.

Below [figure 4] displays the message stream between SNMP agents and SNMP managers.

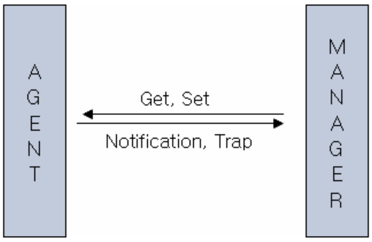


Fig. 4 Action process

3. Implementation

With previous chapter's architecture design, I embody agent part that supports wireless sensor network.

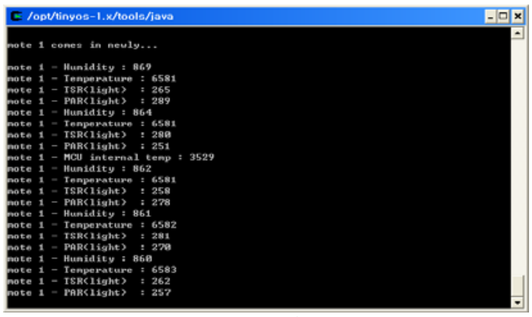
I use sensor modules composed of two different sensors having different MCU and those can be able to compose the sensor network.

The differences between these sensors are platforms of the application program loaded in sensor module and UART. But whole system configuration is equal in sensors of different kinds. The reason that I use these differences sensors is that they are not subjected in specification environments.

1) Ustar2400 wireless sensor module (Huins) - using Chipcon2420 RF module, ATmega128 MCU and 2.4GHz IEEE 802.15.4 communication methods (Micaz pattern)
2) TIP50CM wireless sensor module (MAXFOR) - using Chipcon2420 RF module, TI's MSP430 and 2.4GHz IEEE 802.15.4 communication methods (Telos A pattern)

Except the base sensor module, the others are installed in OscilloscopeRF program that could send the sensed values. The base sensor module is installed in base station program. It is part of a bridge that can receive messages coming from sensor modules or sending messages from PC to other sensor modules. If you wish to compose systems using TinyDB in your host PC, just install TinyDB application programs to your sensor modules. But TinyDB is not operated in the Telos sensor module because its resources are restricted.

I tested the Telos sensor module. Host PC brought environment information from the base sensor module by running a serial forward program. Below [Figure5] is sensed information read at a serial forward program through the MoteIF interface.



```
mote 1 comes in newly...

mote 1 - Humidity : 869
mote 1 - Temperature : 6581
mote 1 - TSR(light)  : 265
mote 1 - PAR(light)  : 289
mote 1 - Humidity : 864
mote 1 - Temperature : 6581
mote 1 - TSR(light)  : 280
mote 1 - PAR(light)  : 251
mote 1 - MCU internal temp : 3529
mote 1 - Humidity : 862
mote 1 - Temperature : 6581
mote 1 - TSR(light)  : 258
mote 1 - PAR(light)  : 278
mote 1 - Humidity : 861
mote 1 - Temperature : 6582
mote 1 - TSR(light)  : 281
mote 1 - PAR(light)  : 270
mote 1 - Humidity : 860
mote 1 - Temperature : 6583
mote 1 - TSR(light)  : 262
mote 1 - PAR(light)  : 257
```

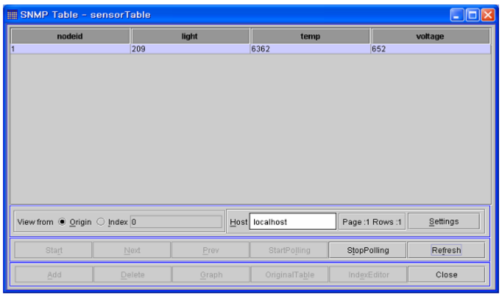
Fig. 5 Sensed information

Environment information accepted in this host PC was passed to a SNMP agent part and the sensed values was managed by the SNMP agent. After monitored by an automatic polling program, we received the result managed by SNMP agent in this sensor information table, [Figure6]. The periodic polling test shows that the SNMP agent brings environment information from the sensor properly.

The SNMP manager acts like a client collects environment information from the SNMP agent and manages sensors through these commands Trap, Get, Set etc.

SNMP is widely used all round as a standard protocol, and the several application programs are possible. SNMP manager can access to the SNMP agent through basic information (IP, community information, MIB OID etc.) that can approach to the MIB.

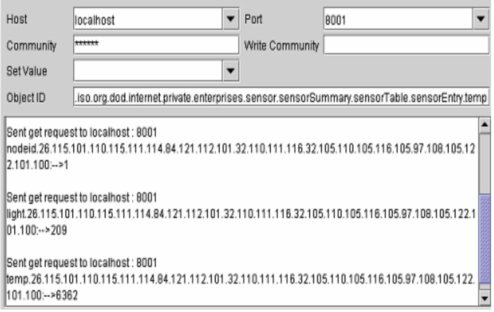
Simply inserting SNMP manager module to an application or a middleware will get the environmental information of the sensor easily and the information will be able to use. [Figure 7] is the environment information of sensor got from MIB browser (client).



nodeid	light	temp	voltage
1	209	6362	652

View from ● Origin ○ Index 0
Host localhost
Page :1 Rows :1

Fig. 6 Monitoring of sensor table



Host localhost
Port 8001
Community ******
Write Community
Set Value
Object ID .iso.org.dod.internet.private.enterprises.sensor.sensorSummary.sensorTable.sensorEntry.temp

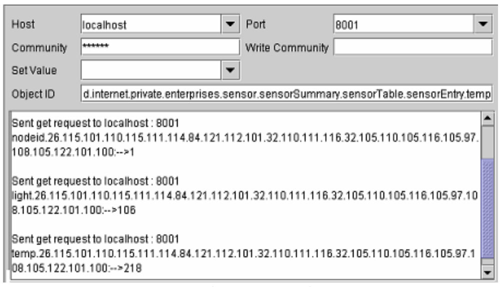
```
Sent get request to localhost : 8001
nodeid.26.115.101.110.115.111.114.84.121.112.101.32.110.111.116.32.105.110.105.116.105.97.108.105.12
2.101.100:-->1

Sent get request to localhost : 8001
light.26.115.101.110.115.111.114.84.121.112.101.32.110.111.116.32.105.110.105.116.105.97.108.105.122.1
01.100:-->209

Sent get request to localhost : 8001
temp.26.115.101.110.115.111.114.84.121.112.101.32.110.111.116.32.105.110.105.116.105.97.108.105.122.
101.100:-->6362
```

Fig. 7 Sensed information that uses Get command

[Figure 8] below is environment information of sensor using the Micaz sensor module.



Host localhost
Port 8001
Community ******
Write Community
Set Value
Object ID d.internet.private.enterprises.sensor.sensorSummary.sensorTable.sensorEntry.temp

```
Sent get request to localhost : 8001
nodeid.26.115.101.110.115.111.114.84.121.112.101.32.110.111.116.32.105.110.105.116.105.97.
108.105.122.101.100:-->1

Sent get request to localhost : 8001
light.26.115.101.110.115.111.114.84.121.112.101.32.110.111.116.32.105.110.105.116.105.97.10
8.105.122.101.100:-->106

Sent get request to localhost : 8001
temp.26.115.101.110.115.111.114.84.121.112.101.32.110.111.116.32.105.110.105.116.105.97.1
08.105.122.101.100:-->218
```

Fig. 8 Sensed information of Micaz sensor

Through the experiments above, I confirm that the SNMP module could search and control the environment information of sensors regardless of sensor types.

The wrapping of the agent using SNMP provides a transparent characteristic to the developers of application programs who do not need to know every vendor's API in input processing of various sensors. This gives the efficiency to developing systems.

4. Conclusion and future studies

In order to manage sensing information that supports wireless sensor network, this paper suggests a method of transmitting and saving sensing information through SNMP protocol. Also, through realizing agent parts, it suggests that the structure using previous SNMP protocol is also efficient in transmitting and saving sensing information. Through using such architectures, a sensor management structure can be built to satisfy the functions of information transmission, information saving, conditional informing and control.

By using a standardized SNMP protocol, problems in previous context aware computing technologies can be resolved, such as resource discovery, incompatibility and singleness of communication type, insufficient flexibility in development, etc.

Future studies will focus on a management system that can execute intelligent functions such as SNMP through using resources in wireless sensor itself. Also, by comparing previous method of sensor information transmission with the method using SNMP, we will look into the practical problems in efficient sensor management.

Acknowledgments

This research was supported by the Program for the Training of Graduate Students in Regional Innovation which was conducted by the Ministry of Commerce Industry and Energy of the Korean Government

References

[1] Anind K. Dey, "Providing Architectural Support for Building Context-Aware Applications," PhD thesis, College of Computing, Georgia Institute of Technology, Dec 2000.

[2] S. S. Yau, F. Karim, Y. Wang, B. Wang, and S.Gupta, "Reconfigurable Context-Sensitive Middleware for Pervasive Computing," IEEE Pervasive Computing, joint special issue with IEEE Personal Communications, 1(3), pp.33-40, July - September 2002.

[3] Nicholas Hanssens, Ajay Kulkarni, Rattapoom Tuchinda, and Tyler Horton, "Building Agent-Based Intelligent Workspaces," In ABA Conference Proceedings, June 2002.

[4] X. Wang, D. Zhang, J. S. Dong, C. Chin and S. R. Hettiarachchi. "Semantic Space: A Semantic Web Infrastructure for Smart Spaces", IEEE Pervasive Computing, 3(3):32-39, July-September 2004.

[5] William Stallings, "SNMP, SNMPv2, SNMPv3, and RMON1 and 2 Third Edition," Addison Wesley New York

[6] Kugsang Jeong, Deokjai Choi, Soohyung Kim and Gueesang Lee, "An SNMP agent for context delivery in ubiquitous computing environments," ICCSA2005, proceeding part V, pp.203, May 2005.

Hongseok Jang received the B.S. in Computer Science from Chonnam Nat'l University in 2002. He will be graduating in August 2006 with his Master's degree in Computer Science. His research interests include sensor network management and application.

Kugsang Jeong received the B.S. and M.S. degrees in Computer Science from Chonnam Nat'l University in 1999 and 2001, respectively. He is currently pursuing his PhD in Computer Science at the Chonnam Nat'l University. His research interests include ubiquitous middleware and application.

Deokjai Choi received the B.S. degree in Computer Engineering from Seoul Nat'l University in 1982, M.S. degree in Computer Science from KAIST in 1994 and, P.H. degree in Computer Engineering from Missouri-Kansas University in 1995, respectively. He is currently a Professor of Electrical and Computer Engineering at the Chonnam Nat'l University.