Self-Healing Methodology in Ubiquitous Sensor Network

Giljong Yoo, and Eunseok Lee
School of Information and Communication Engineering Sungkyunkwan University
300 Chunchun Jangahn Suwon, 440-746, KOREA
{gjyoo, eslee}@ece.skku.ac.kr

Abstract

This work concerns the development of a fault model of sensor for detecting and isolating sensor, actuator, and various faults in USNs (Ubiquitous Sensor Network). USN are developed to create relationships between humans, objects and computers in various fields. A management research of sensor nodes is very important because the ubiquitous sensor network has the numerous sensor nodes. However, Self-healing technologies are insufficient to restore when an error event occurs in a sensor node in a USN environment. A layered healing architecture for each node layer (3-tier) is needed, because most sensor devices have different capacities in USN. In this paper, we design a fault model and architecture of the sensor and sensor node separately for self-healing in USN. In order to evaluate our approach, we implement prototype of the USN fault management system to evaluate our approach. We compare the resource use of self-healing components in the general distributed computing (wired network) and the USN.

Keywords: *Ubiquitous Sensor Network, Self-Healing, Fault Management, Layered Architecture*

1. Introduction

This paper defines the fault model that addresses sensor problems. The form of a single sensor node varies, depending on the application requirements, from the big box to a microscopically small particle. Each device comprises a sensor platform, a processing unit, a communication unit and power unit. Sensor faults are a normal fact and unlike in traditional networks (not isolated events). In the USN a management workload of failure is high because human intervention is difficult. Fault detect ion and repair as essential to guarantee network QoS.

We propose a fault model and self-healing architecture to heal the sensor network. Through the proposed approach, it is possible to heal an abnormal state affecting a sensor capacity. An existing recovery system of wireless environment has solutions that are limited to topology modification in response to the sensor node and network situation. This solution is based on node self-configuration, not self-healing. The concept of sensor management is shown in Fig. 1.

The sink node and manager node interact in a wired network. This topic is outside the scope of this paper; we only consider wireless networks. We compare the effectiveness of our proposed model to a traditional healing method by a case study.

This work is organized as follow. Section 2 presents related work. Section 3 presents the fault model for USN healing. Section 4 discusses a method of self-healing for the USN service. Section 5 evaluates a proposed approach. Section 6 concludes the paper and identifies future work.

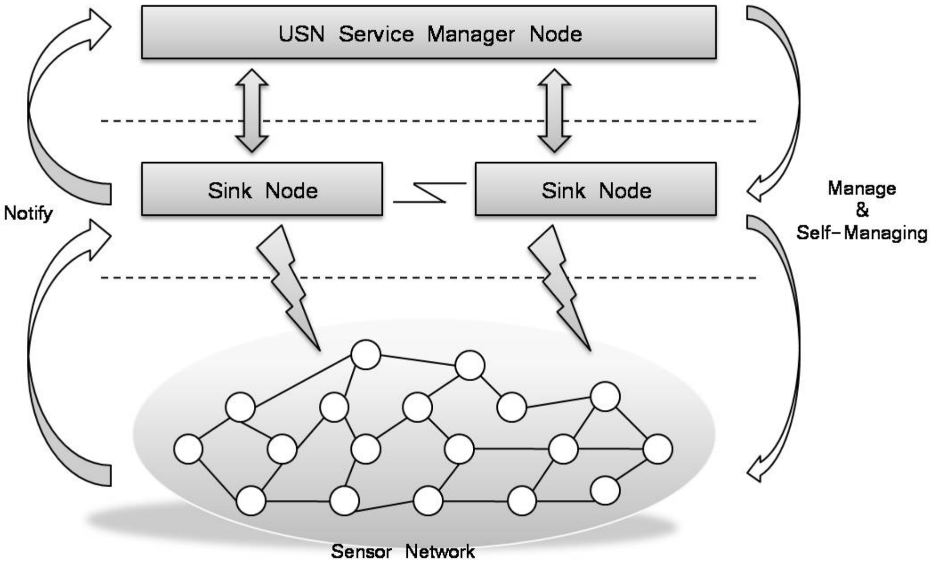

Figure 1. Layered sensor management

2. Related Work

Wireless sensor network are rapidly evolving into ubiquitous sensor networks. USN is an extensive concept than WSN. USN is a network that perceives and manages object information using electronic tag, U-sensors and BcN. We define the USN as a basic infrastructure to implement a ubiquitous society. Conversely, WSN implements a wireless based sensor network using RF technology (Zigbee [1], Bluetooth [2] and WiFi [3]), Ad hoc and mesh network technology in a wire based sensor node. Most existing research is based on WSN. It is very important since it is the basis for USN technology [4].

Aviienis [5] gives precise definitions characterizing the various concepts that address the dependability and security of computing and communication systems. His elementary fault classes are presented in Fig. 2.

The USN recently remarked a development of system integration and mobile wireless environment. The key point is how to make the strategy of software management in USN. An administrator is difficult to recover a sensor fault. Therefore a research of autonomous management needs to manage wireless sensor device. Stephen Brown [6] presents a model for standard and technique of the WSN software update such as in Fig. 3. If sensor is fault, the healing manager makes a healing strategy and recovers a fault sensor using a remote communication (software update).

3. Fault Model for healing of Sensor Network

The application of the Fault Model represents a major branch of modern autonomic computing for self-healing.

.

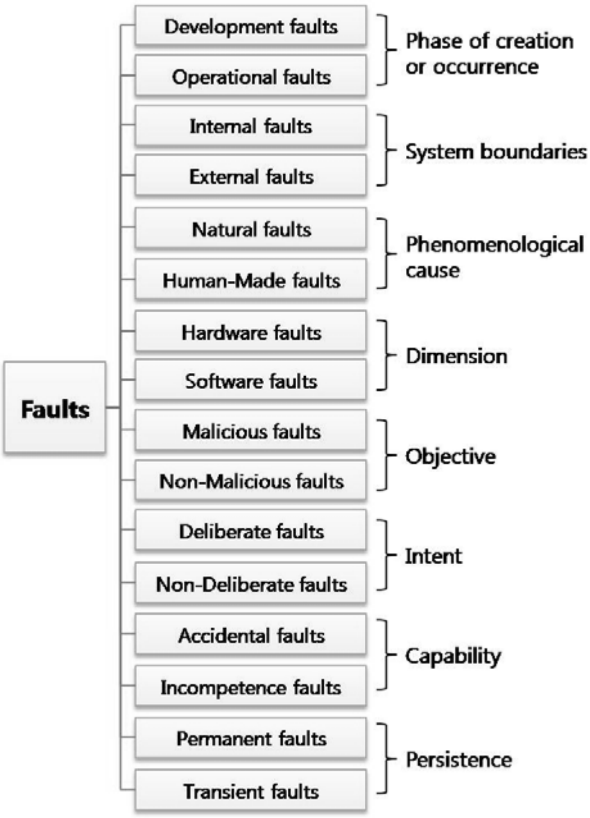

Figure 2. The elementary fault class

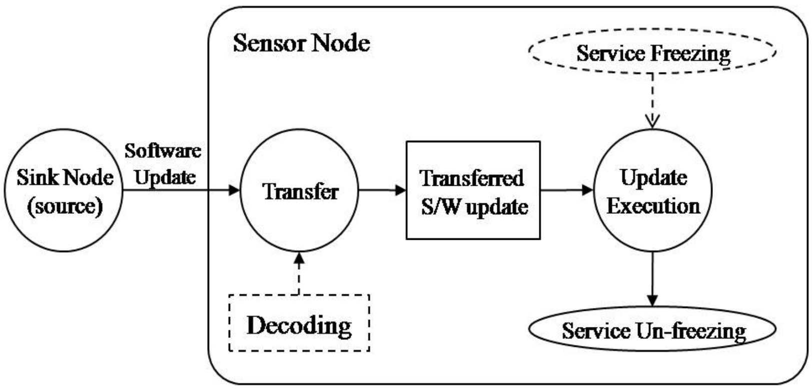

Figure 3. WSN software update model

We define an engineering fault model of issues in the system requirements, software, hardware and environment of sensor nodes for USN service. An objective of the fault recovery is to extend the sensor network lifetime through restart/re-programming of faulty sensors. A USN fault model attaches a service fault model to a basic fault model based on service classification. A basic USN fault model is depicted in Fig. 4. The USN service classifies faults by QoS, software, device and environment

- QoS (Service) Fault: The QoS fault maintains a required service quality. The QoS is based on a pre-defined service model of USN. If specific service is violated, then system error occurred.
- Software Fault: The software fault handles a development fault and operational fault. This fault means that a logical error of software, an error of design phage and implementation phage. In design phage, its software reveals a permanent fault. Therefore, it gives a bad influence on a system state and an operation behavior such as a program control flow and a communication link.
- Device Fault: The device fault relates a battery and hardware part of sensor. For example, a specific hardware fault interrupts all wire communication from one gateway to other gateway.
- Environment Fault: If network assume basic environment to maintain USN service, communication disturbance happened by environmental is the Environment Fault. The environment constraint considers a temporary network obstacle, wind and rain, etc.

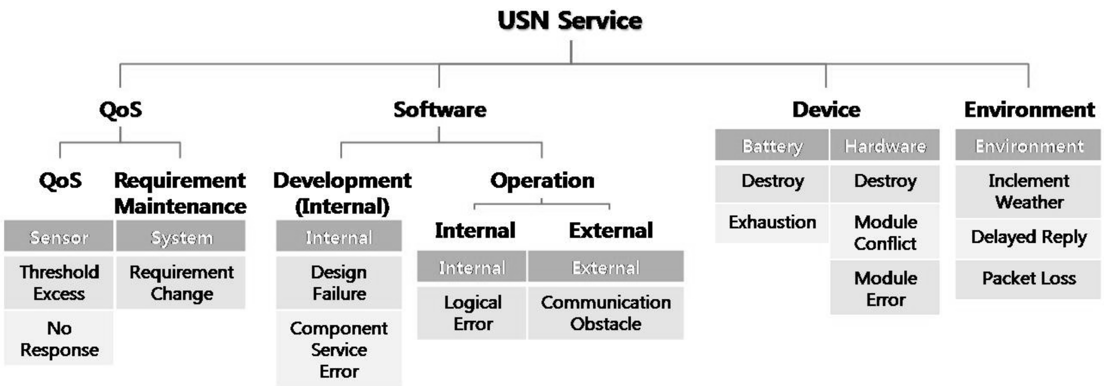

Figure 4. Basic fault model of USN

The sensor node and sink node differ in their capacities to perform USN services. We assume the sensor node has low power and the sink node, such as a PC, high power. Therefore, each node layer must have different healing components. Optimized modules must be designed to reduce the sensor node workload. We assume that a wireless network connection is substantially safe, because the sink node produces a recovery image to heal the sensor node and sends it to the sensor node by wireless.

4. A Self-Healing and Management in USN

4.1. Autonomic USN Management

A sensor node consists of *Application platform, Processing Unit, Storage, Sensor Unit, Wireless Network,* and *Power Supply.*

We attach a healing architecture to sensor module such as Fig. 5. This architecture repair a fault generated in sensor node. A goal of our approach is the AUMS (Autonomic USN Management System) that is consisted with basic functionality of sensor, the USN fault model, and the self-Healing technology.

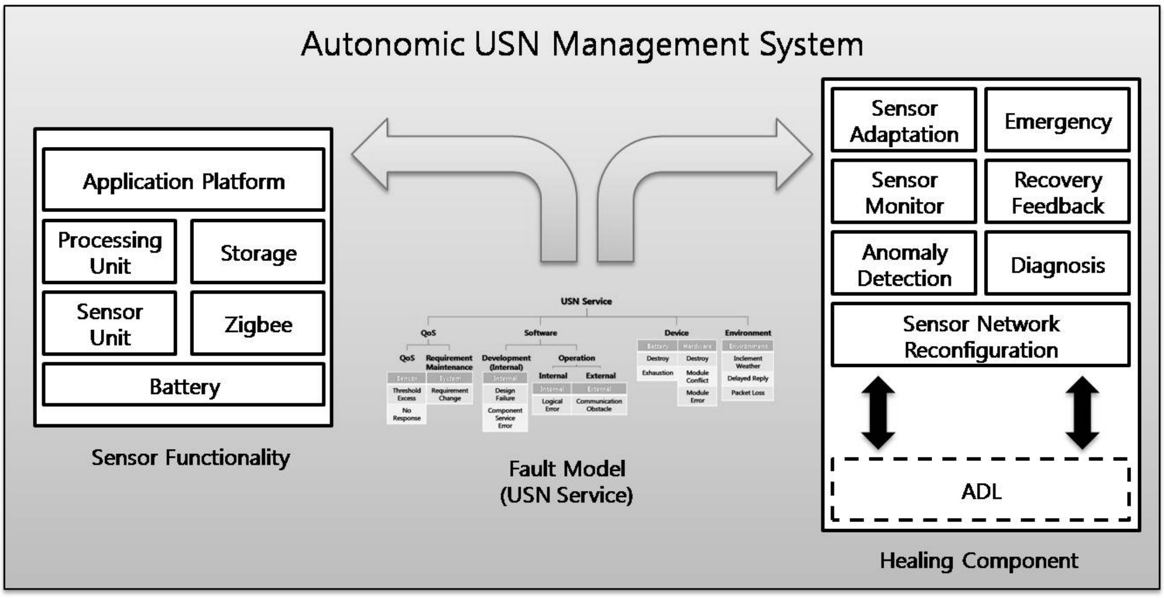

Figure 5. Autonomic USN management system

4.2. Self-Healing Architecture in Sensor Node

In the Smart Dust project [7], a sensor of USN is various from low-cost and the smallest sensor to heavy weight. Among them, we assume a sensor that performs the simple service.

The sensor is composed of minimum healing modules to perform the self healing such as in Fig. 6. The healing module is attached to this sensor. It has no adaptation strategy and only performs self monitoring and execution of adaptation strategy.

The Sensor Monitor /Feedback observes the sensor information via the operating system (①). If healing is failed, then it reverts to the version prior to healing (⑫). Sensor has state diagram that describes own service. The Semi-Diagnosis sends the monitored information to the communication (IEEE 802.15.4-2006 standard, Zigbee) when a state chart is in error (②). Sensor Adaptation adapts a recovery image if it detects an anomalous symptom (⑪).

4.3. Self-Healing Architecture in Sink Node

A sink node (Fig. 7) is responsible for the sensor healing. Also, own situation sends to a manager node and requests a healing and a managing. An interaction between sink node and manager node is operated in the wire network. So this subject is out of the scope of this paper. After all, high level nodes manage and heal low level nodes.

A sink node has more high processing and communication ability than a sink node. The Sensor Gauge collects the monitored information of sensor node (⑥). Then, Sensor Diagnosis reads this information and determines if an error has been generated (⑦). The Adaptation Generator generates an adaptation strategy and a healing code (image) (⑧). The Sensor Reconfigurator performs reconfiguration to heal a faulty sensor node. It performs reconfiguration if adaptation is terminated.

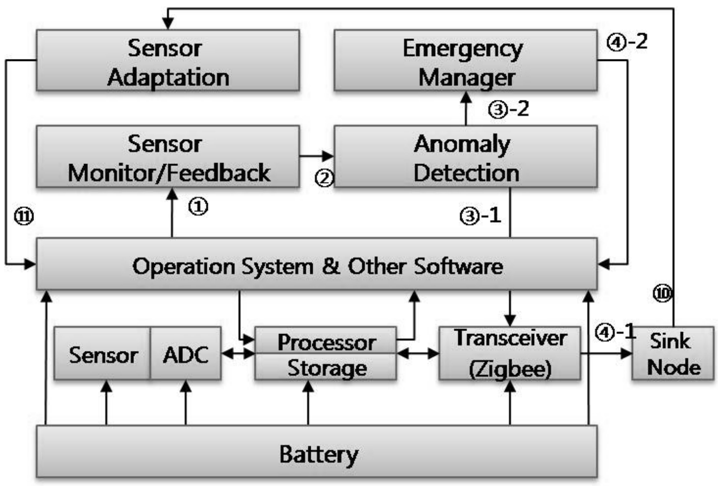

Figure 6. Sensor node self-healing architecture

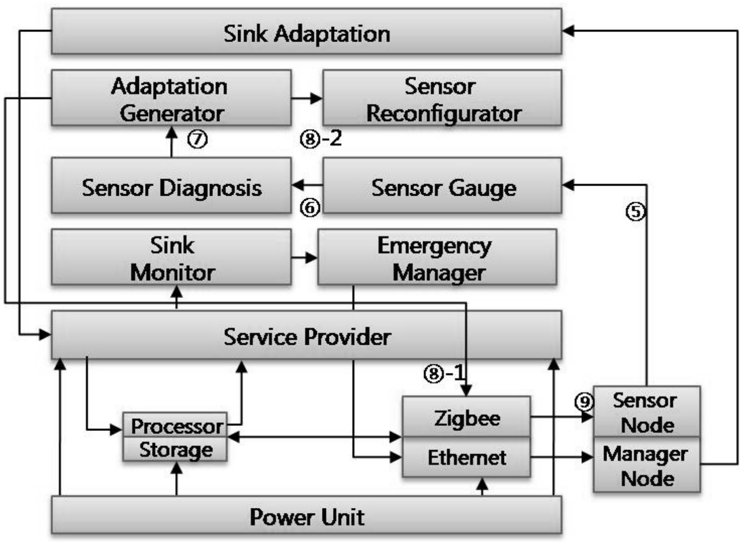

Figure 7. Sink node self-healing architecture

5. Evaluation

We develop the USN self-healing system prototype using the Microsoft .NET. A healing process between the sensor and sink node is depicted in Fig. 8. Left side process is in sensor and right side process is in sink.

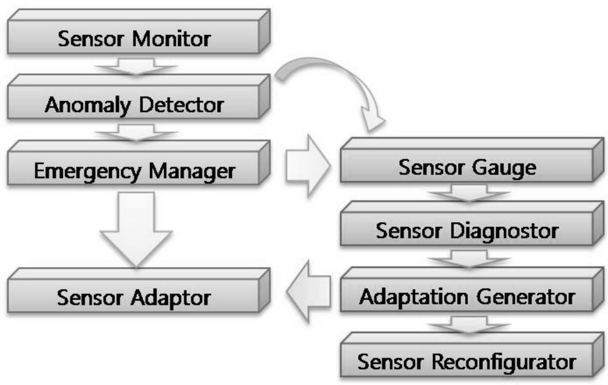

Figure 8. Prototype of healing process

```
#include "nos.h"

// Interrupt occurs when detected object is gone.
void pir_intr_handler(void)
{
        uart_puts("\n\r\tPIR Detected!\n\r");
}
int main(void)
{
        nos_init();
        led_on(1);
        uart_puts("\n\r*** Nano OS ***\n\r");
        pir_power_on(); // power on PIR SENSOR sensor
        pir_callback(pir_intr_handler);
        while (TRUE)
        {
                uart_putc(_BS);
                uart_putc('-');
                delay_ms(200);
                uart_putc(_BS);
                uart_putc('\\');
                delay_ms(200);
                uart_putc(_BS);
                uart_putc('|');
                delay_ms(200);
                uart_putc(_BS);
                uart_putc('/');
                delay_ms(200);
        }
        pir_power_off(); // power off PIR SENSOR sensor
        return 0;
}
```

Figure 9. Self-healing component resource usage

Figure 9 show a sample code by NanoQplus in ETRI [9]. This code has 4 primary functions; led-on(), pir_power_on(), pir_callback(), and pir_power_off(). We use AOP to monitor the core sensor operation/function code such as Fig. 9. Monitoring code is generated by the predefined fault model. Once a sensor fault occurs, the Anomaly Detector detects this fault. Then it determines the fault state and delivers the information to the Emergency Manager or the Sensor Gauge.

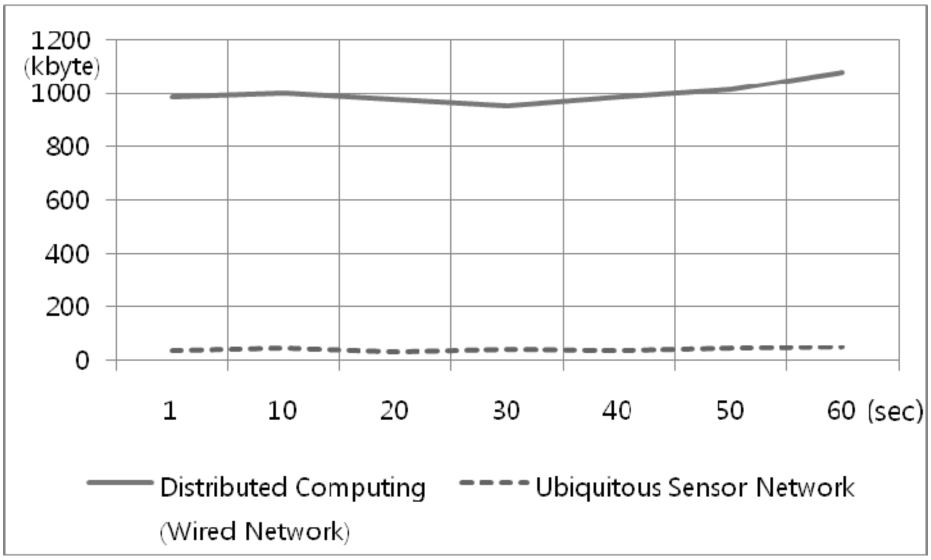

Figure 10. Self-healing component resource usage

In this paper, we use the ADL to manage a structure of sensor network like software architecture. About 10 ADLs is used to describe architecture. Among them, the ACME and C2 style is the most popular. ACME lacks dynamism to manage architecture. Therefore, we think that the C2 is the best choice since USN has a hierarchical structure and dynamic reconfiguration.

Most of the related researches mostly focus a temporarily recovery. On the other hand, in high level service approach, a self-healing is very insufficient. In addition, a self-healing methodology of an existing distributed computing environment is difficult that it adapt to the USN.

We designed a layered healing architecture for USN healing. We compare the resource usage of a self-healing component in our USN approach to the general distributed computing (wired network). The experimental results are shown in Fig. 10. This result differs remarkably between distributed computing and USN. The reason is that both computing devices and healing process workload are very different in each case. We design the layered healing architecture for healing the USN. There is significant saving in the resource usage to heal sensor fault in the USN.

6. Conclusion

In this paper we described a technique for a layered architecture to heal sensor networks. USN has three layers: sensor node, sink node, and manager node. Until now, USN and their applications have been developed without consideration of a fault management solution. This is a critical problem since networks comprising tens of thousands of nodes are expected to be used in some applications. This paper focuses on the approach to satisfy the need of a different healing architecture for USN since nodes has differing capacity to process services. Our approach is compared to existing fault management methods in wired computing. Its effectiveness is demonstrated by experiment. We need further research on layered self-healing architecture.

Reference

[1] ZigBee Alliance, http://www.caba.org/standard/zigbee.html/

[2] Bluetooth, http://www.bluetooth.com/bluetooth/

[3] WiFi Alliance, http://www.wi-fi.org/

[4] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, “A Survey on Sensor Networks”, IEEE Communication Magazine, pp.102-114, Aug.2002

[5] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," IEEE Transactions on Dependable and Secure Computing, Vol. 1, No. 1, pp. 11-33, 2004

[6] Stephen Brown and Cormac J. Sreenan, “A new model for updating software in wireless sensor networks”, IEEE Network, Vol.20, pp.42-47, Nov-Dec.2006

[7] Smart Dust Project, http://www-bsac.eecs.berkeley.edu/archive/users/warneke-brett/SmartDust/index.html

[8] Teruyoshi Zenmyo, Hideki Yoshida, and Tetsuro Kimura, “A self-healing technique using reusable component-level operation knowledge”, Cluster Computing, Vol.10, pp. 385-394, Dec.2007

[9] nanoQplus, http://qplus.or.kr/

Authors

Giljong Yoo received his B.S. degree in Computer Engineering from Paichai University, Korea, in 2005, and M.S. degree in Computer Engineering from Sungkyunkwan University, Korea, in 2007. He is currently a Ph.D. student in Electrical, Electronic, and Computer Engineering at the Sungkyunkwan University. His research interests include Autonomic/Ubiquitous Computing, Reflective Middleware, and Ubiquitous Sensor Network.

Eunseok Lee received his Ph.D. and M.S. degrees in Department of Information Engineering from Tohoku University, Japan, in 1992 and 1988, respectively, and his B.S. degree in Electronic Engineering from Sungkyunkwan University, Korea, in 1985. He is a Professor of the Department of Computer Engineering of Sungkyunkwan University, Korea. From 1994 to 1995, he was an assistance professor of the Department of Information Engineering of Tohoku University, Japan. He was a research scientist in Information and Electronics Laboratory of Mitsubishi Electric Corporation, Japan, from 1992 to 1994. His areas of research include Software Engineering, Autonomic/Ubiquitous Computing and Agent-oriented Intelligence System.