A Cooperative Multi-agent System Infrastructure for Heterogeneous Sensor Environment *

Taishi Ito[†], Yasuhiro Kurita[†], Hideyuki Takahashi[†,‡],
Kazuto Sasai[†,‡], Gen Kitagata[†,‡], Tetsuo Kinoshita[†,‡]
†*Graduate School of Information Sciences, Tohoku University, Japan*
‡*Research Institute of Electrical Communication, Tohoku University, Japan*
{itot, kurita}@k.riec.tohoku.ac.jp,
{hideyuki, kazuto, minatsu, kino}@riec.tohoku.ac.jp

Abstract

We aim to provide an advanced context-aware service based on various sensors and home appliances. We propose a concept of an agent-based cooperative framework for heterogeneous sensors. Our framework can adjust detection, addition, and removal of sensors to provide context-aware services. We implemented a prototype system and performed experiments to confirm the effectiveness of our framework.

Keywords: Ubiquitous Computing, Sensor Network, Cooperative Method, Multi-agent

1: Introduction

Recently, services that utilize various sensors and information have been emerging. Further many computing fields like ubiquitous computing, pervasive computing and ambient-computing have been under research and development. In these fields, researchers have proposed life support services such as watching services, healthcare and multimedia services. Nowadays, the interests in environmental burden, energy savings and electric power dealing and there are several technologies or mechanisms for this, for example Green ICT or Home Energy Management System (HEMS). Hence, we need a mechanism for utilizing various kinds of information from heterogeneous devices to match these service requirements. Our goal is to provide a flexible cooperative system for heterogeneous sensors and home appliances, and we propose a sensor network infrastructure system based on a concept of multi-agents. The system empowers cooperative work between various sensors (smart meters, sensors in living environments, positioning sensors, body area sensors etc.) and appliances. With such a system we can easily realize an elderly people watching service, a healthcare support system, or an energy management system etc. However, there are some technical challenges.

(P1) It is difficult to cooperate with heterogeneous sensors following the application requirements.

(P2) It is difficult to deal with improvisatorial addition or removal of sensor devices.

We propose the following two methods.

(R1) Agentification of entity for a cooperation method between heterogeneous sensors.

(R2) A cooperative communication protocol for addition and removal of sensors.

Our contribution is to give a sensor application the capability to adapt to change of sensors' existence.

The rest of the paper is organized into the following sections. In section 2, we discuss related works. Section 3 introduces a cooperative mechanism to heterogeneous sensors. In Section 4, we present an implementation and usage scenario of its application. Finally, we give conclusions and highlight future works in section 5.

2: Related Works

Some research has already been researched about cooperative framework and technology for utilizing heterogeneous sensor networks. The Context-Toolkit is a framework for supported of construction of a context-aware service[1]. The framework has five components: a context widget, an aggregator, an interpreter, a service component and a discover component. Also, a framework and a toolkit are provided to the developer.

An event-driven processing framework based on the Context-Toolkit provides an event subscribe mechanism[2]. The event subscribe mechanism defines a sensor detection method as a sequence of receiving of requirements, judgment, and notification. For the mechanism, sensors are regarded as a programming interface and an application developer can easily construct a context-aware application.

Further, there is a study of a mobile agent-based framework for configurable sensor networks, where a middleware for sensor networks is proposed[3]. In the study, all sensors in the network are software agents, so the process of addition and removal of a sensor is regarded as an agent behavior.

In another study of agentification in a sensor network, a data collection system using mobile agents for sensor networks integrating multiple stations is proposed[4]. The sensor network has a data gathering node that is called a SinkNode. The mechanism aims at reduction of the communication burden with agentification of the SinkNode.

The Wearable Toolkit supports an incremental development for wearable computing environments[5]. The toolkit generates an activity from the users' context that is extracted from sensor data and the developer of the application can easily develop and debug the context-aware application. Sensors in the study are not only physical devices but also software components such as mailers and chat software, etc. Sensors can be added or removed from the toolkit at runtime.

We organize technical challenges of flexibly mechanism of heterogeneous sensors. We call sensors a device that measures physical amount of an entity in the real environment. An example of a physical amount is temperature, humidity or illumination. If a sensor outputs two or more signals (such as temperature and humidity or temperature and illumination), we also call it a sensor. We call the sensor composition heterogeneous sensors. In an application using the sensors, users can add, remove or rearrange the sensors. However, two technical problems arise when trying to realize these features.

(P1) It is difficult to cooperate with heterogeneous sensors following the application requirements.

(P2) It is difficult to deal with improvisatorial addition or removal of sensor devices.

In the articles[3, 4], the authors employ an agent architecture but there is little information of a concrete usage of the heterogeneous sensors. In addition, in all methods in studies[1, 2, 3, 4, 5],

application requirements for the sensors are pre-defined by an application designer. Therefore, the application cannot deal with improvisatorial addition or removal of sensor devices.

To cope with the above problems, the heterogeneous sensor network has to be able to deal with different output formats of the sensors. It should also be made possible to add, remove and rearrange sensors. We propose an agent-based cooperative framework for heterogeneous sensors. The framework has the following features.

(R1) Agentification of entity for a cooperation method between heterogeneous sensors.

(R2) A cooperative communication protocol for addition and removal of sensors.

Based on R1, we can solve P1 by making sensor devices and applications act as agents. Thus, application requirements can be stored as knowledge in the agents as sensor requirements. Based on R2, the cooperative communication protocol solves P2 by managing status of sensors. If there is improvisatorial addition or removal of sensors, the application can deal with the situation.

## 3: A Cooperative Mechanism to Heterogeneous Sensors

Our mechanism consists of four groups of agents which are called Sensor Device Agents, Real World Agents, Sensor Application Agents and Manager Agent (Figure 1). Sensor Device Agents normalize the physical amount, which is observed from a sensor device, and transmit the data to a corresponding Real World Agent. Real World Agents have the role to retain the data coming from Sensor Device Agents, the arrival time and the place and to publish data. Sensor Application Agents are applications such as energy saving services and monitoring services. The agent transmits a request that consists of a spatial constraint and a kind of physical amount to the Real World Agent. If the Real World Agent has no data at the exact requested place, the agent interpolates data. For this reason, our mechanism solves technical challenges. A Manager Agent is a mediator agent that has a role to manage functional information of each agent.

3.1: Agentification of Entity for A Cooperation Method Between Heterogeneous Sensors

We agentify all types of components in the real environment such as sensor devices and applications, based on the AMUSE[6]. The agentification is to apply knowledge into entities and to make the entities act as agents. Agents can treat status of entities as a context. We add a context management function, an inter-agent cooperation mechanism, and an adaptive communication mechanism which flexibly can select communication schemes for the agents[7]. Then the agent can compose services by satisfying various service requirements. A Sensor Device Agent has the role to normalization process and to acquire the location of the observation point. Sensor data coming from Sensor Device Agents to a Real World Agent consists of two elements: a normalized physical amount and a location of an observation point. Examples of a location are shown in Table 1 and of physical amounts in Table 2. The Sensor Device Agent transmits a key-value pair shown in Table 3 to the corresponding Real World Agent. In the table, "<>" are filled up with actual sensor data.

3.2: A Cooperative Communication Protocol for Addition and Removal
     of Sensors

The cooperative communication protocol is depictured in Figure 2. It shows four Sensor Device Agents and one Real World Agent and a special agent called Manager Agent. The latter manages
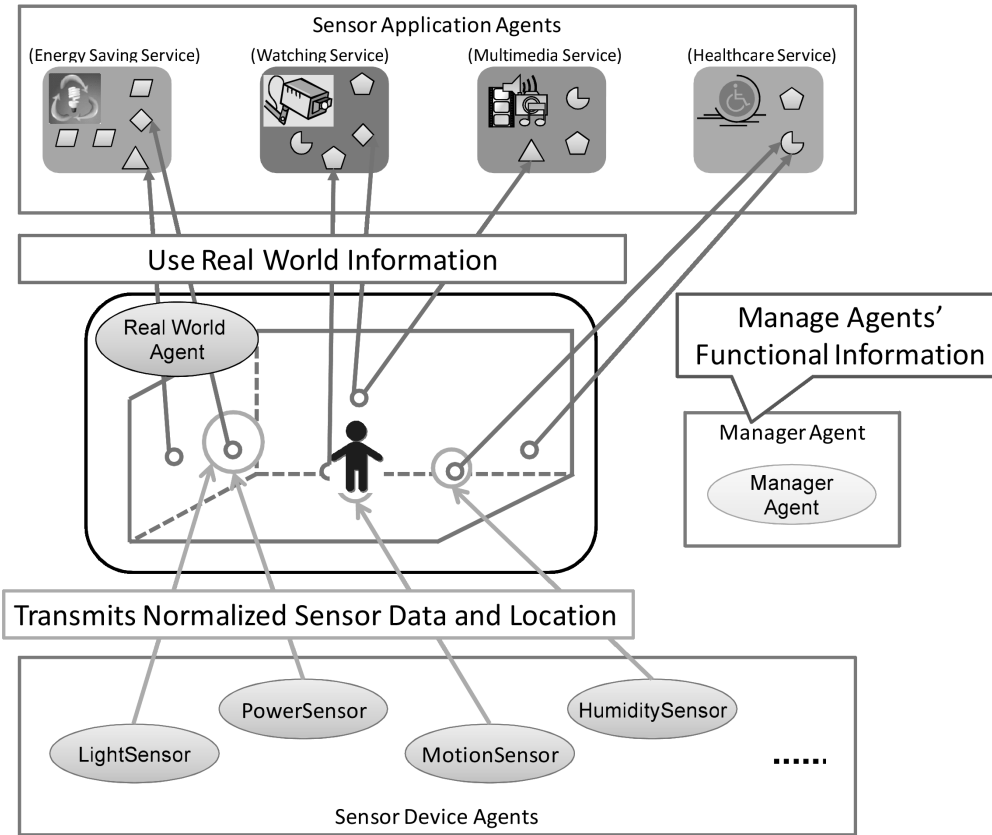
Figure 1. An architecture of the proposed method

Table 1. Examples of a real environment locations

Location	Explanation
W302	Room W302
W303	Room W303
(30.0, 15.0, 100.0)	A relative position of real environment

agents' functional information and stores their existence via the cooperative communication protocol. The Manager Agent firstly broadcasts a lookup message to other agents (Figure 2(1)) and the other agents reply with their functional information to the Manager Agent (Figure 2(2)). When Sensor Device Agents try to transmit sensor data to the Real World Agent, the Sensor Device Agent inquires the Real World Agent's location to the Manager Agent (Figure 2(3), (4)).

In addition, to handle improvisatorial addition or removal of sensor devices, the Manager Agent fuses sensor data and periodically checks for existence of a recipient agent. We show an application example of our mechanism in Figure 3. In rooms W302 and W303, there are several sensors and corresponding Sensor Device Agents. There are two Real World Agents in the environment and two types of Sensor Application Agents that use the Real World Agent. The Real World Agent processes addition and removal of sensor device agents according to the protocol.

The process of the improvisatorial removal of sensor device agents is a reverse operation. A Manager Agent deletes the functional information of an agent which was detected being removed

Table 2. Examples of normalized physical amounts

Type	Explanation	Range of value
Force	Pressure	0 to 1000
Light	Illumination	0 to 1000
Touch	Contactless or contacting	0 or 1000
Rotation	Turning angle	0 to 1000
Slider	A position of slider device	0 to 1000
Motion	Motion sensor value	0 to 1000
RFID	RFID Tag	Character String
Power	Electricity consumption	0 to ..
...	...	...

Table 3. A data structure to inform physical information to a virtual real-environmental agent

| Key	Value
Location	<A location in real environment>
<Type of physical amount>	<Data of physical amount>
...	...
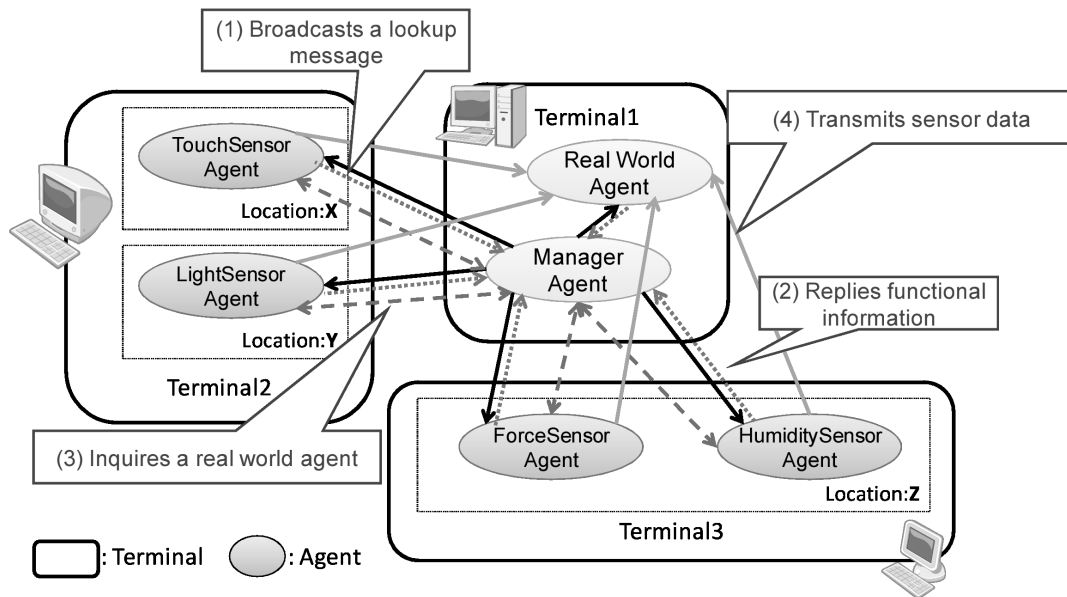
Figure 2. Cooperative communication protocol for addition and removal of sensors

from the environment. And the Manager Agent broadcasts a message which consists of the agent's name and the type of the function. Thus, agents which received the message can recover an internal process to a former state. With this procedure, a system can operate stably even if a sensor device or an agent is replaced or a sensor is damaged.

In addition, the combination of states does not depend on the number of sensors, but depends on the number of the types of sensors. Therefore, because the types of sensors are limited, the combi-
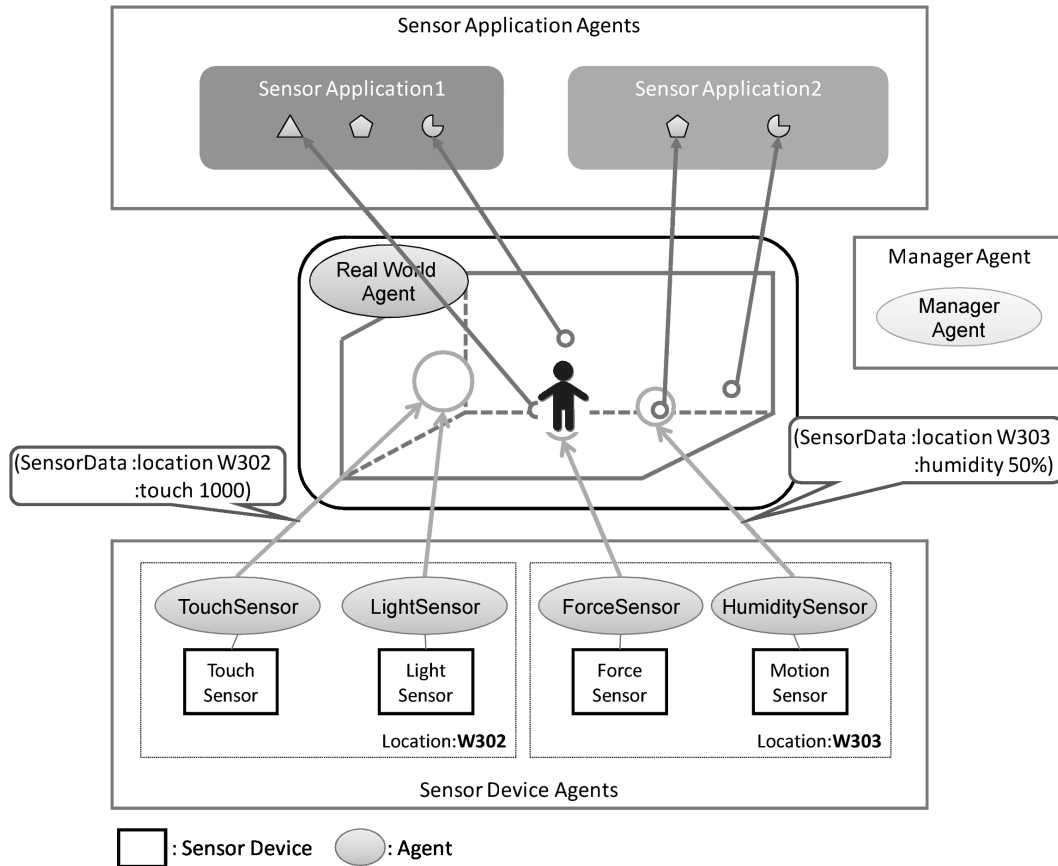
Figure 3. Application example of proposal mechanism

nation of states is limited in our method. Thus, it is possible to program all those combinations and pre-assign the actions.

4: Experiment and Evaluation

We have developed a prototype system to confirm the effectiveness of our mechanism. We used Phidgets[8], e-NODE[9], MOTE[10], and IDEA[11]. The latter is the agent framework developed by Java 6 SE. Phidgets has a light sensor, a servo motor, a touch sensor, a pressure sensor, a slider and a RFID receiver/tags. e-NODE has an electric power sensor, a humidity sensor and a Solid-State-Relay (SSR) unit for the control of power supply. MOTE has a humidity sensor and a light sensor. In addition, Phidgets' sensors are connected by a wired communication link to PC, e-NODE and MOTE by wireless (IEEE 802.15.4). Agents in the system transmit the following structured sensor data by an inter-agent communication language.

$$(< object > :< attribute1 > < value1 > \ ...)$$

The structure, called OAV (Object-Attribute-Value), is used in the IDEA framework for communication between agents. For instance, when a sensor agent in room W302 observes "illuminance is

Phidgets Motion Sensor
(Detects Infrared)

Figure 4. Experimental scenario of the gas alert application

300, pressure is 200, and RFID Tag is XYZ, ...", then the following OAV will be transmitted.

$$(SensorData \ : location \ W302 \ : light \ 300 : force \ 200 \ : RFID \ XYZ \ ...)$$

4.1: Experimental Method

We have run experiments on the following two applications.

(Expr. 1) Gas Alert Application

(Expr. 2) Humidifier Control Application

First, in Expr. 1 we confirm the cooperation method using a motion sensor and a Sensor Application Agent. Second, in Expr. 2 we confirm the cooperative mechanism and addition and removal of a sensor using a Phidgets light sensor, a humidity sensor, an RFID receiver/tag, an electric power sensor and an SSR unit.

4.2: Expr. 1: Gas Alert Application

The Gas Alert Application signals by a warning message and a sound if the gas stove is kept on for some period of time. The experiment scenario includes the following four steps.

1. The gas stove is turned on.
2. The gas stove is kept on for some period of time.
3. A motion sensor agent detects that the situation is unsafe.
4. The application alerts by a warning message and a sound.

We show the agent composition of the application in Figure 5. The motion sensor can observe infrared and detects the fire. The Manager Agent decides whether the gas stove has been kept
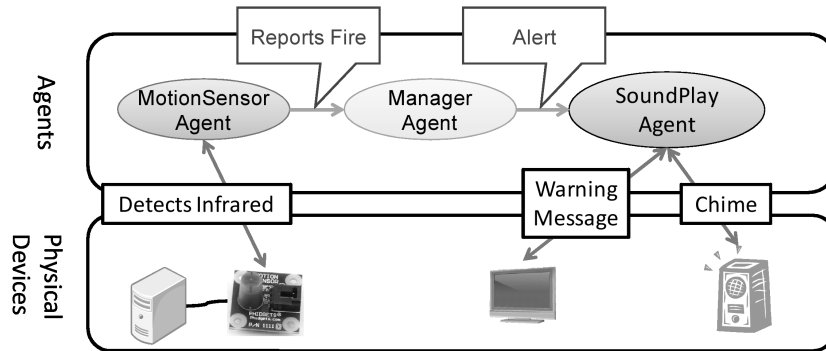
**Figure 5. Agent composition of the gas alert application**

on for some period of time or not, and if so the SoundPlay Agent alerts someone the situation. To detect the situation of the gas stove, we use sensor data of the Phidgets motion sensor. The sensor produces a value with a range between 0 and 1000 and with a median of 500. Therefore, we smooth the output with the moving average method. We have a preliminary experiment that decides the window of time of the moving average method and a threshold for the fire detection. We show the experimental result in Figure 6. The x-axis shows elapsed time in seconds, and the y-axis shows the smoothed output. From the figure, we can conclude the window of time to 5 seconds, with a threshold of 500. We show the experimental results of Expr. 1 in Figure 7. At (A) the initial state
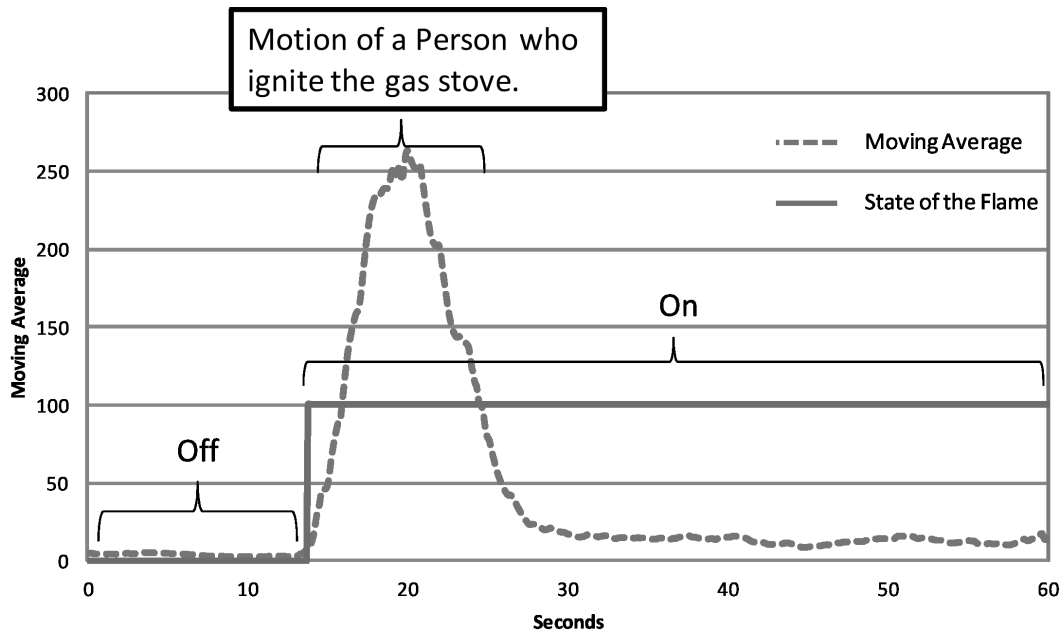
Figure 6. Knowledges of the agent

of the application is shown, at (B) a person turns the gas stove on, at (C) the stove is kept on during some period of time and at (D) the application alerts by a warning message and a sound. From the

(A)

(B)

Ignite the Gas Stove

Burning Continuously

Chime & Warning

(C)

(D)
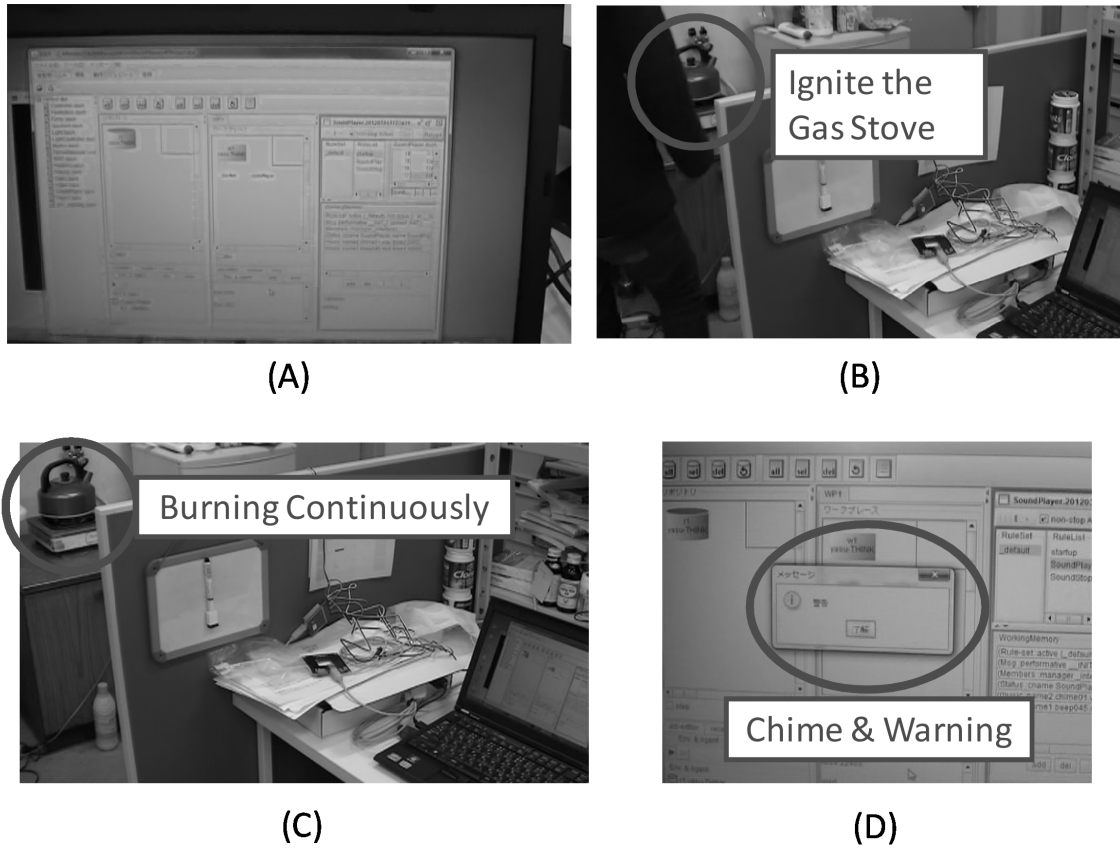
Figure 7. A snapshot of the gas alarm application

experimental result, we can confirm that the cooperation method is effective.

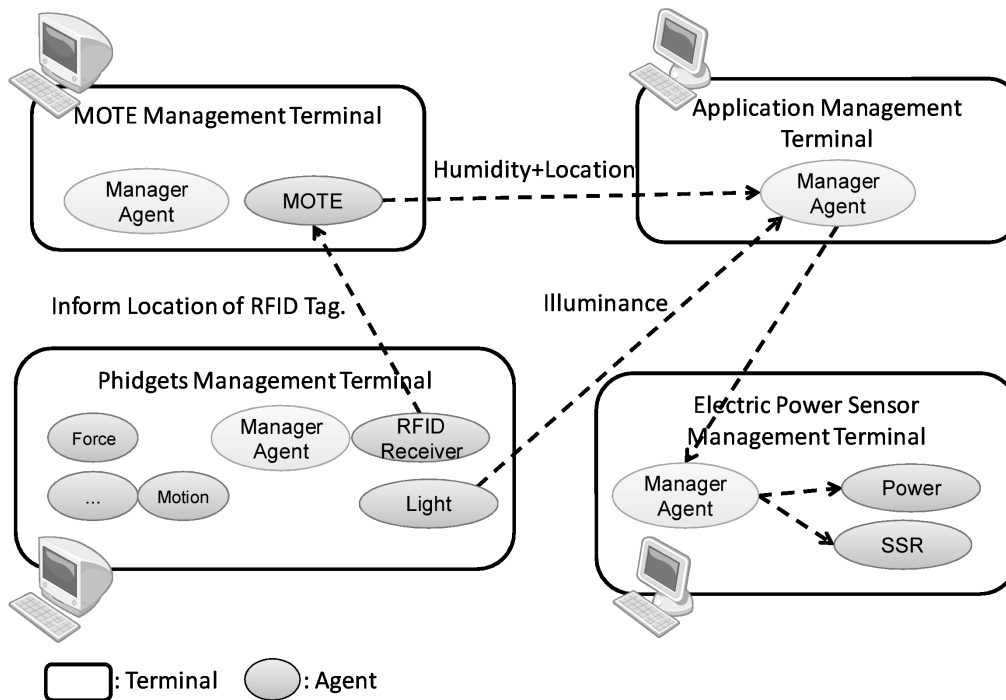### 4.3: Expr. 2: Humidifier Control Application

With the Humidifier Control Application, the experimental objective is to confirm whether it is possible to control heterogeneous sensors or not. We show the devices composition of the application in Figure 8. The experimental devices are a humidifier which only has a simple heating function (the humidifier cannot control itself according to humidity), a Phidgets light sensor, a MOTE humidity sensor, an RFID receiver/tag and an e-NODE SSR unit. To introduce location of MOTE into the application, we patched the RFID tag to MOTE.

We show the agents composition in Figure 9. The Light agent transmits information of illuminance to the Manager Agent and then the Manager Agent infers human existence and controls the humidifier. The Manager Agent controls the humidifier by switching the power supply on and off by the SSR agent based on information of illuminance which is transmitted from the Light agent. In addition, if a RFID tag is set to the RFID receiver, the Mote agent will transmit information of humidity and its location to the Manager Agent and the Manager Agent will try to use the information and control the humidifier based on information of illuminance and humidity.

We have a preliminary experiment that confirms whether the control of the humidifier is possible by using the SSR unit. If humidity is less than a threshold (50%) the SSR unit turns the humidifier on, otherwise off. The result is shown in Figure 10. The x-axis shows elapsed time in seconds,
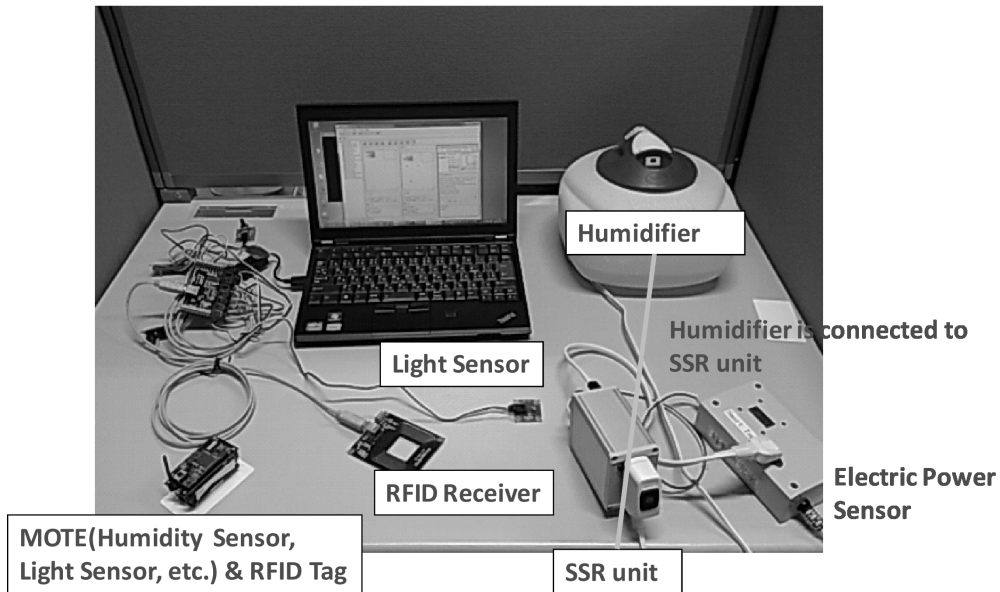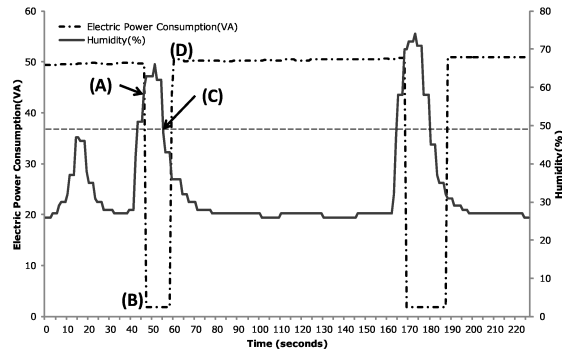
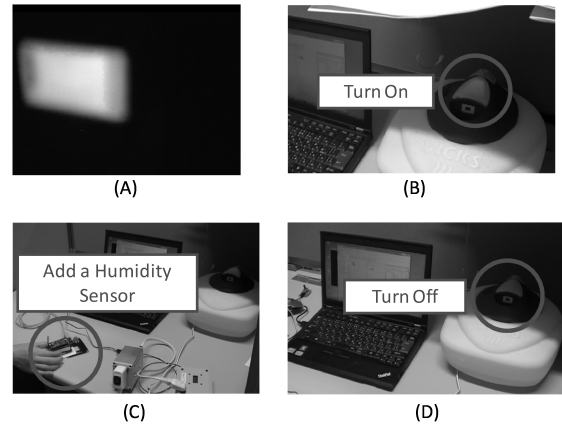**Figure 8. Composition of the humidifier control application**

Figure 9. The agent composition of the humidifier control application

the left y-axis shows humidity and right shows electricity consumption in Watts. The dotted line shows the threshold for the switching of the humidifier's power supply. At (A) humidity is over the threshold and then at (B) the SSR unit turns the humidifier off. At (C) humidity is under the

Figure 10. An experimental result of the relationship between humidity and power consumption of the humidifier

Figure 11. A snapshot of the humidifier controller application

threshold and then the SSR unit turns the humidifier on at (D). From the experimental result, we can confirm it is possible to control the humidifier by the SSR unit.

The experimental scenario of Expr. 2 includes the following two steps.

1. The light is turned on when someone comes into the room.

2. MOTE is set to RFID receiver.

We show the result in Figure 11. At (A) the humidifier and light is off, at (B) a person comes into the room and the humidifier and light is turned on, at (C) a person sets the MOTE to the RFID receiver and the application detects a humidity sensor and at (D) humidity is increased and the Manager Agent switches the humidifier off. From the result, we can confirm that the cooperative mechanism works effectively with the addition of a sensor.

In addition, we confirmed the removal of the sensor device from removing the MOTE from the system. The Humidifier Control Application returns its behaviour to the former.

4.4: Evaluation

From experimental results, we confirmed cooperation depending on the application among heterogeneous sensors. We confirmed the cooperative mechanism. In addition, the delay of the prototype system was up to around three seconds. Therefore, we consider that our system can endure practical use.

5: Conclusion

Our goal was to provide a flexible cooperative system for heterogeneous sensors and home appliances, and we proposed a sensor network infrastructure system based on a concept of multi-agents. The system empowers cooperative work between various sensors and appliances. With the system we can easily realize an elderly people watching service, a healthcare support system, or an energy management system etc.

Our experiments confirm effectiveness if the proposed mechanism by a prototype system. From experimental results we can confirm that the cooperation method and cooperative mechanism works

effectively. We will refine the cooperative mechanism and the cooperation method, design a sensor data collection mechanism and an analysis algorithm for modeling context-aware service.

6: ACKNOWLEDGEMENTS

References

[1] G.D. Abowd, A.K. Dey, and D. Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," HCI, Vol.16, pp. 97-166, 2001.

[2] T. Matsuura, S. Tagashira, T. Kitasuka, T. Nakanishi, and A. Fukuda, "Event-Driven Processing Framework Supporting Sensor Cooperation for Home Networks," B - Abstracts of IEICE TRANSACTIONS on Communications, Vol. J92-B, No. 7, pp. 1050-1060, 2009.

[3] T. Umezawa, I. Satoh, and Y. Anzai, "A Mobile Agent-based Framework for Configurable Sensor Networks(Special Issue on High Speed Networks and Multimedia Applications)," IPSJ Journal, Vol. 44, No. 3, pp. 779-788, 2003.

[4] T. Yoshihisa, A. Kanzaki, T. Hara, Y. Ishi, Y. Teranishi, and S. Shimojo, "A Data Collection System using Mobile Agents for Sensor Networks Integrating Multiple Stations," The Institute of Electronics, Information and Communication Engineers, Vol. 109, No. 131, pp. 147-151, 2009.

[5] T. Terada, M. Miyamae, M. Yamashita, "Wearable Toolkit: an Event-driven Rule Processing Engine and Tools for On-site Programming," IPSJ Journal, Vol. 50, No. 6, pp. 1587-1597, 2009.

[6] H. Takahashi, Y. Tokairin, T. Suganuma, and N. Shiratori, "Design and Implementation of An Agent-based middleware for Context-aware Ubiquitous Services," Frontiers in Artificial Intelligence and Applications, New Trends in Software Methodologies, Tools and Techniques, Vol.129, pp. 330 – 350, 2005.

[7] T. Ito, H. Takahashi, T. Suganuma, T. Kinoshita, and N. Shiratori, "Design of Adaptive Communication Mechanism for Ubiquitous Multiagent System," Journal of Information Processing, Vol. 18, No. 18, pp. 175 – 189, IPSJ, 2010.

[8] Phidgets Inc. - Unique and Easy to Use USB Interfaces: http://www.phidgets.com/ available at Jul. 2012

[9] e-NODE: http://www.embedded-sys.co.jp/products/enode.html available at Jul. 2012

[10] Crossbow MOTE: http://www.xbow.jp/01products/index.html available at Jul. 2012

[11] T. Uchiya, T. Maemura, H. Hara, K. Sugawara, and T. Kinoshita, "Interactive Design Method of Agent System for Symbiotic Computing," International Journal of Cognitive Informatics and Natural Intelligence, Vol. 3, No. 1, pp. 57–74, 2008.

Authors

Taishi Ito received M.S. degree in 2009 from Tohoku University, Japan. Currently, he is pursuing his doctoral degree in Graduate School of Information Sciences (GSIS), Tohoku University. His research interests include agent-based framework and its application. He is a student member of IPSJ.

Yasuhiro Kurita received B.A. degree in 2012 from Tohoku University, Japan. Currently, he is pursuing his M.S degree in Graduate School of Information Sciences (GSIS), Tohoku University. His research interests include sensor framework and its application.

Hideyuki Takahashi is an assistant professor of Research Institute of Electrical Communication of Tohoku University, Japan. He received his doctoral degree in Information Sciences from Tohoku University in 2008. His research interests include ubiquitous computing, green computing and agent-based computing. He is a member of IPSJ and IEICE.

Kazuto Sasai is an assistant professor of Research Institute of Electrical Communication, Tohoku University, Japan. He received his Dr.Sci. in Earth and Planetary Science from Kobe University. His research interests include network management, network analysis, complex systems and agent-based systems. Dr. Sasai is a member of IEICE, IPSJ, JSAI, and BSJ.

Gen Kitagata is an associate professor of Research Institute of Electrical Communication of Tohoku University, Japan. He received a doctoral degree from Graduate School of Information Sciences, Tohoku University in 2002. His research interests include agent-based computing, network middleware design, and symbiotic computing. He is a member of IEICE and IPSJ.

Tetsuo Kinoshita is a professor of Research Institute of Electrical Communication of Tohoku University. He received his B.E. degree in electronic engineering from Ibaraki University, Japan, in 1977, and M.E. and Dr.Eng. degrees in information engineering from Tohoku University, Japan, in 1979 and 1993, respectively. His research interests include agent engineering, knowledge engineering, knowledge-based and agent-based systems. He received the IPSJ Research Award, the IPSJ Best Paper Award and the IEICE Achievement Award in 1989, 1997 and 2001, respectively. Dr. Kinoshita is a member of IEEE, ACM, AAAI, IEICE, IPSJ, and JSAI.