

# Finding Small Changes using Sensor Networks

Kaoru Hiramatsu, Takashi Hattori, Tatsumi Yamada, and Takeshi Okadome

NTT Communication Science Laboratories, Japan

{hiramatsu,takashi\_hattori,tatsumi,houmi}@cslab.kecl.ntt.co.jp

<http://www.kecl.ntt.co.jp/>

**Abstract.** A sensor network makes environments smarter by sensing small changes of physical quantities using heterogeneous networked sensors, such as small sensor nodes and fixed type sensors. This paper illustrates our experimental setting developed in an ordinary office using prototype sensor nodes and reports how a sensor network detects small changes in a smart environment.

## 1 Introduction

Since Weiser published an article on ubiquitous computing [1], we have witnessed many projects that build smart environments to provide humans with reactive and proactive services that support daily activities. A key feature in such environments is an unobtrusive sensor network that consists of in-situ type small sensor nodes and fixed type sensors connected to the network that enable a system to capture small changes of physical quantities in the environment, such as changes of temperature, human movements, and the opening and closing of doors.

Many researchers and organizations (e.g., Smart Dust project [2] and Sensor Web project by NASA JPL [3]) are concentrating on sensor networks and have published various technologies. TinyOS [4] is an operating system designed for wireless embedded sensor networks. TinyDB [5] is a query processing system for extracting information from a network of TinyOS sensors. Regiment [6] is a functional macroprogramming language based on region streams that represent spatially distributed, time varying collections of sensor node states. Some of the main research topics are routing algorithms that enable a system to efficiently monitor sensor data streams and optimize communication and query processing for reducing battery usage.

These technologies enable us to program reactive services based on single, small changes easily sensed by one sensor. For example, it is easy to program a reaction that turns on a room light corresponding to a single sensor that detects the opening of a door. We simply connect a state of the door sensor to a switch of the room light. In contrast to this simple reaction, there are some difficulties in building services based on relevant changes sensed in the environment. For example, imagine a service that controls a surveillance camera according to unusual sensor states (e.g., window vibration and strange noises). The service might require conjunctive conditions carefully selected from the possible combinations of sensor states captured by a sensor network. In this programming, the developers have to select appropriate conditions from possible combinations. Also, additional efforts are required to reuse the program in other places if the

conjunctive conditions deeply depend on the characteristics of heterogeneous sensors in the environment.

To improve such inflexibility in developing smart environments, we tackle the following issues:

1. Encapsulation of low level controls of individual sensor nodes and communication protocols via networks.
2. Abstraction of contexts from the small changes captured by the sensor network.
3. Creating an upper layer program that finds appropriate conditions that systematically trigger services.

If these issues are tightly coupled with each other and the abstract contexts are described freely, the reusability of the developed program would be spoiled again. Thus the system should be layered into lower level controls of sensor devices and higher levels of application programs by importing data models based on functional differences and abstract contexts. Also, contexts are important for expressing what the system intends to detect in the environment, and a common vocabulary is indispensable to express and interpret contexts without ambiguity and misunderstanding.

The process to develop sensor networks based on the above issues includes some exploratory challenges, but here we focus on the second issue. In contrast to Antifakos's experiments [7] that observed the correlation of audio signals using Smart-Its sensor boards, we analyze the cooccurrences of small changes captured by various types of sensor devices in a sensor network. Cooccurrence is a feature extracted by a system from the sensor data even if the sensor devices have different characteristics and various sampling rates. Moreover, cooccurrences should be logically and systematically related to abstract concepts in hierarchical knowledge (e.g., WordNet [8] and ontologies [9]). This abstraction enables developers to concatenate distributed systems smoothly even if based on different vocabularies and granularity.

The remainder of this paper is organized as follows. Section 2 presents the details of our prototype sensor network that consists of prototype sensor nodes and fixed type sensors. Section 3 analyzes the cooccurrences of small changes from the recorded sensor data of the office and discusses how they correspond with abstract concepts. Finally we draw our conclusions in Section 4.

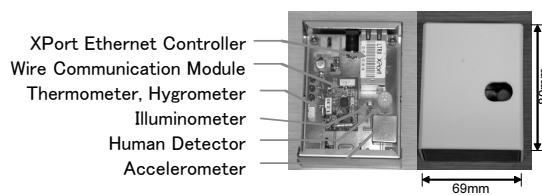
## 2 Building a Prototype Sensor Network

Imagine an office furnished and used by individuals. In the morning, somebody opens the office door, turns on the lights and the air conditioner, and places his chair in front of his desk. Soon the air conditioner comfortably controls the temperature, and then he may walk to drink a cup of coffee and take a break. The office environment is slightly transformed by these events, some of which are captured by a sensor network as small changes of physical quantities and sensor states.

To capture such small changes, we develop in-situ type prototype sensor nodes equipped with currently available sensor devices and set up a prototype sensor network by attaching sensor nodes to arbitrary objects in the office and configuring fixed type sensors.

## 2.1 Prototype Sensor Nodes

To capture the small changes that occur in the office, we develop in-situ type prototype sensor nodes. It is easier to attach small sensor nodes than big ones to such objects in the office as chairs, desks, doors, and so forth. Moreover, long life batteries, wireless communication, higher sampling rates of sensor devices, and related computational power are also indispensable in the early phase of our experiment. But it is hard to simultaneously satisfy all requirements (e.g., battery capacity and size are closely correlated). Satisfying the experimental requirements as much as possible, we develop prototype sensor nodes (Figure 1) using battery packs employed in mobile phones and currently available sensor devices. Figures 2 and 3 show sensor nodes attached to a door and a chair, respectively.



**Fig. 1.** Overview of prototype sensor node

Our sensor node is equipped with a thermometer, a hygrometer, an illuminometer, an accelerometer, and an infrared sensor (that detects human movements). It is also equipped with H8/36049 (an embedded CPU from Renesas Technology Corp.), XPort (an Ethernet interface module by Lantronix, Inc.), and iB-5234EK (a wireless communication module from Millennial Net, Inc.). Since the H8/36049 is not powerful enough to analyze all sensor data streams, the CPU in our prototype sensor nodes samples the analog data of the sensor devices, converts them into digital data, and packs them into a packet that it transmits via a wired or wireless network.

## 2.2 Fixed Type Sensors

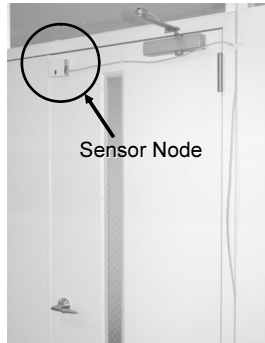
In addition to prototype sensor nodes, we utilize the following fixed type sensors by attaching them to installed objects such as doors, windows, and office floors.

### – Sensor floor system

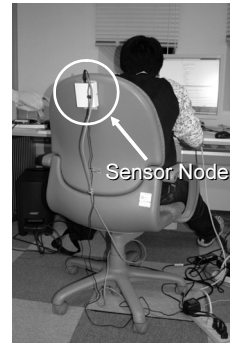
We set up a sensor floor system (VS-SF99 by Vstone Co., Ltd.) on the office floor divided into 18 cm squares with output existence of pressure within each square. By analyzing the distribution of the pressure, the system can detect the positions of fixed and moving objects in the office.

### – Reed switches

We also installed reed switches at the doors and the windows to detect whether they were open or closed. They were connected to XPEVA (by Device Drivers Limited) with XPort for reporting the states of the switches via the network.



**Fig. 2.** Prototype sensor node attached to a door



**Fig. 3.** Prototype sensor node attached to a chair

### 2.3 Experimental Setting

The top left of Figure 4 shows an office layout in which we established a sensor network. To enable people to continue working without interruption, we installed desks, chairs, bookshelves and put many objects around the room, such as books, PCs, cups, and so forth. We then attached the prototype sensor nodes to the objects.

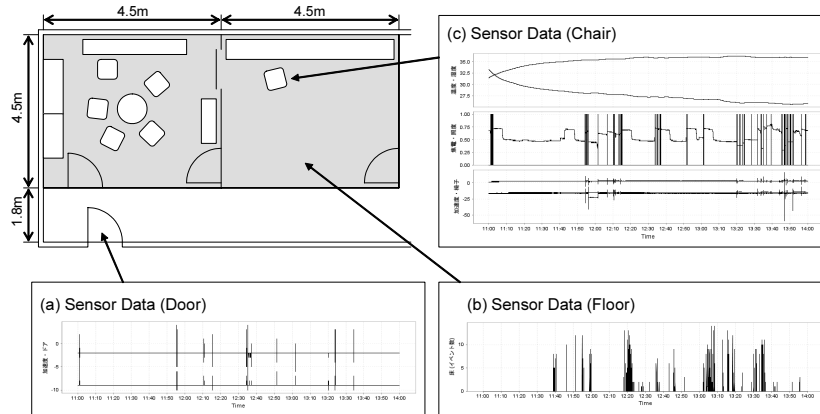
We also set up fixed type sensors to complement the prototype sensor nodes. The sensor floor system was fixed to the floor, the grayish area in Figure 4. The coverage area of the sensor floor system is about 9 m by 4.5 m. Reed switches were installed at the doors, the windows, and the bookshelves in the office to report their states (e.g., open or closed) via XPEVAs.

All sensors belong to a sensor network that utilizes a sensor data management system [10] such as middleware that simplifies dealings with sensor data streams by encapsulating low level controls of respective sensor devices and communication protocols on the network.

## 3 Finding Small Changes

Using the experimental setting, we recorded small changes including temperature fluctuations, human movements, and such events as the opening and closing of doors. Figure 4 shows the practical data captured by the sensor network from 11:00 a.m. to 14:00 p.m. on May 2nd, 2005. In this experiment, the sampling rate of each sensor was about 100 ms.

During this period, a person was using the office to develop a software module of the sensor network. As shown in the bottom left graph ((a) Sensor Data (Door)) and the bottom right graph ((b) Sensor Data (Floor)) of Figure 4, we can observe that he might have opened and closed the door several times and walked around the office. The other graph ((c) Sensor Data (Chair) in Figure 4) was recorded by the prototype sensor node attached to the chair and regarded the person's behavior around the chair (e.g., when



**Fig. 4.** Small changes sensed at chair, door, and floor in an office

he approached it and how often he reclined against it). However, these descriptions of behavior greatly depend on the human knowledge of the office and a worker's usual behavior. Without such knowledge, it is difficult for a system to abstract events and contexts from the raw sensor data and hard to isolate important and informative parts in the sensor data streams.

To avoid this problem, we propose another approach that enables us to logically and systematically abstract events and contexts from raw data. It is divided into two phases: (i) analyzing the raw sensor data and (ii) relating them to predefined knowledge. In the following, we focus on phase (i) and extract cooccurrences of the small changes of the sensor data.

### 3.1 Approach

There are many ways to analyze sensor data captured by sensor networks. According to the characteristics of sensor devices, some sensor data changes rapidly and some slowly (e.g., acceleration observed at the chair increases and decreases more rapidly than the temperature at the same place). It is also hard for a sensor network to sample the sensor data with high frequencies as in a directly connected monitoring systems. These factors make it impossible to deal uniformly and ideally with the sensor data captured by heterogeneous sensor devices.

To avoid such drawbacks, we extract small changes in the sensor data as simple features of the sensor data and find cooccurrences among them. To consider the differences of the change rates of the sensor data, we utilize various lengths of time windows, as shown in Figure 5. While receiving the sensor data, the system memorizes the maximum and minimum data within the time windows. If the differential between the maximum and minimum data within one time window exceeds a threshold, the system determines that the sensor data have changed within the time window. The system scans small changes in the sensor data and calculates probabilities of their occurrences and cooccurrences (e.g., how frequently do sensors A and B change simultaneously?).

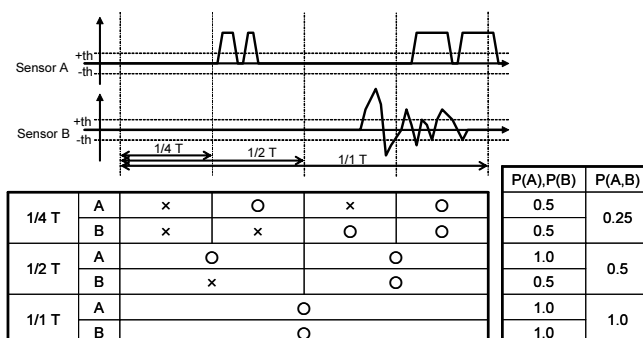


Fig. 5. Definition of small change cooccurrence

### 3.2 Cooccurrences of Small Changes in One Sensor Node

As shown in graph (c) of Figure 4, small changes of the sensor data of the illuminometer, the accelerometer, and the infrared sensor show skewed distribution and tend to cooccur within short times. First, we applied the approach to the sensor data gathered by the sensor node attached to the chair.

To calculate the probabilities of the small changes of the sensor data and their cooccurrences, we set up time windows at 1, 5, 10, 15, and 30 seconds and utilized standard deviation of each sensor data as a threshold of changes. The results are shown in Table 1. Note that the thermometer and hygrometer data are omitted in Table 1 because no change was detected in relatively short time windows.

Table 1. Probabilities of cooccurrences of small changes at chair

small changes	time window (sec.)				
	1	5	10	15	30
accelerometer	.346	.533	.591	.618	.669
infrared sensor	.023	.050	.070	.092	.139
illuminometer	.002	.014	.030	.044	.086
accelerometer, infrared	.013	.041	.062	.085	.128
illuminometer, accelerometer	.002	.012	.025	.036	.072
infrared, illuminometer	.001	.014	.029	.038	.069
accelerometer, infrared, illuminometer	.001	.012	.025	.033	.064

The chair acceleration data had a higher probability of small changes than the other sensors because it was probably moved most frequently during the experimental period. From such probability we can deduce the length of time that a person was sitting in the chair.

In cooccurrence probabilities, the illuminometer and the infrared sensor have a higher cooccurrence probability than the illuminometer and the accelerometer. Since infrared sensors detect human movements, the cooccurrences of small changes between

the illuminometer and the infrared sensor are regarded as such events as “someone approached the chair” and “someone left the chair.”

Small changes of different physical quantities detected by the sensor devices of the sensor nodes show the skewed distribution of cooccurrences; therefore, the probabilities of the cooccurrences of small changes increase by increasing the length of the time window. According to this observation, we expect to extract the sequences of the small changes of the sensor data and summarize events (e.g., someone starts typing and takes a break) by more detailed analysis.

### 3.3 Cooccurrences of Small Changes among Sensor Nodes

Table 2 shows the probabilities of the small changes of the chair accelerometer, the door accelerometer, and data of the sensor floor system and their cooccurrences in the sensor network. They are calculated in the same manner as Section 3.2: by using time windows at 1, 5, 10, 15, and 30 seconds and a standard deviation of each sensor data as a threshold of changes. The probability of small changes of the chair accelerometer was highest and the door accelerometer was lowest. These results reflect usual office activities.

**Table 2.** Probabilities of cooccurrences of small changes among sensor nodes

small changes	time window (sec.)				
	1	5	10	15	30
chair accelerometer	.346	.533	.591	.618	.669
door accelerometer	.008	.017	.026	.029	.053
sensor floor	.063	.107	.134	.165	.217
chair, door	.002	.013	.020	.024	.044
door, floor	.001	.004	.006	.010	.025
floor, chair	.026	.073	.098	.122	.169
chair, door, floor	.001	.002	.005	.007	.022

The probabilities of their cooccurrences increase by extending the length of the time window in the same way as the chair. First, notice that small change cooccurrence probabilities between the door and the floor and between the door and the chair showed a tendency to increase more than between the chair and the door by widening time windows from 15 to 30 seconds. Second, small change cooccurrence probability among them showed the same tendency as small change cooccurrence probabilities that are calculated, including the door.

These points can be explained by distances among the objects. As shown in Figure 4, the door did not touch with the floor and the chair, and it takes some steps to go from the door to the chair. Accordingly, it is hard for people to move between them in several seconds but they could get to the chair in dozens of seconds. By focusing on such peculiar increases of cooccurrence probabilities, we can measure relative distances among sensor nodes in sensor networks and create object maps based on ordinary office activities.

Of course, we may explain events that happened in the office (e.g., how the individual walked to the chair in the office) based on cooccurrence probabilities. In this sense, the analysis of sequences in small changes sensed by spatially distributed sensor nodes and consideration of the length of the time window are important parts of data processing in sensor networks.

## 4 Conclusion

We implemented prototype sensor nodes, set up a sensor network in an office, and investigated the cooccurrences of small changes captured in an office.

Since our project is in the early stages, the experiment described in Section 3 is still in progress. Analysis of the sequences of small changes is the next step. We are planning to investigate the correspondence of sensor data with abstract concepts of knowledge and context in an office. Since this is a key to develop sensible services based on heterogeneous sensors in smart environments, we need to discuss what kinds of knowledge are necessary and how to deal with them.

## Acknowledgment

The authors would like to thank Takashi Mitsuoka of Boid Inc. for his great contributions to the experiment.

## References

1. Weiser, M.: The computer for the twenty-first century. *Scientific American* **265** (1991) 115–124.
2. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., Pister, K.: System architecture directions for networked sensors. *ACM SIGOPS Operating Systems Review archive*, Vol. 34, No. 5, **34** (2000) 93–104.
3. Delin, K.: The sensor web: A macro-instrument for coordinated sensing. *Sensors* **2** (2002) 270–285.
4. TinyOS Project: TinyOS (2004). <http://www.tinyos.net/>
5. Madden, S., Franklin, M., Hellerstein, J., Hong, W.: The design of an acquisitional query processor for sensor networks. In: *Proceedings of SIGMOD Conference*. (2003) 491–502.
6. Newton, R., Welsh, M.: Region streams: Functional macroprogramming for sensor networks. In: *Proceedings of the First Workshop on Data Management for Sensor Network*. (2004) 78–87.
7. Antifakos, S., Schiele, B.: Beyond position awareness. *Journal of Personal and Ubiquitous Computing* **6** (2002) 313–317.
8. Cognitive Science Laboratory, Princeton University: Wordnet (2005). <http://wordnet.princeton.edu/>
9. W3C: Web Ontology Working Group (2004). <http://www.w3.org/2001/sw/WebOnt/>
10. Hiramatsu, K., Ogura, K., Okadome, T.: A sensor data management system based on p2p communication adaptors. In: *Proceedings of the 67th Annual Convention IPS Japan*. Volume 3. (2005) 367–368 (in Japanese).