

Experiences with Building Sentient Materials using Various Sensors

Tetsuo Yamabe, Kaori Fujinami, Tatsuo Nakajima
Department of Computer Science
Waseda University
{yamabe, fujinami, tatsuo}@dcl.info.waseda.ac.jp

Abstract

In the future, every material will contain a computer and become smart. We call these materials *sentient materials*. Sentient materials can detect their state, and enable an application to change their behaviors according to its surrounding environments in a context-aware way. However, there are many difficulties about building a sentient material, which is especially inherent in using various sensors. This paper describes some issues for building sentient materials from experiences with building simple prototype implementations, and discusses how to solve the problems by using a concept of the world model.

1 Introduction

Ubiquitous computing environments create a seamless boundary between atoms and bits. Especially, in the near future, embedding various types of sensors in our daily lives enables an application to customize their behavior according to its user's situation. Also, physical objects in our daily lives can be used to manipulate information in cyber spaces.

One of the most important issues to build ubiquitous computing environments is how to offer context information for customizing services. The customization of services according to a user's preference and situation needs various context information about our daily lives. The first step to retrieve the information requires to use various sensors. By analyzing raw data acquired by the sensors, an application retrieves context information from the low level information.

However, current sensor technologies are not suitable to retrieve our context information. We believe that future materials will contain many small computers and sensors, and they will give various information to us [1, 2]. For example, they will calculate geographic relationship among materials, or provide unique identifiers that are useful to know what they are. We call the intelligent materials *sentient materials*. As it is in some research examples studied now, context information acquisition which is based on a sentient material is one useful method to detect human action. Figure 1 shows context acquisition process using various sensors. A general process is shown in the

left-hand side of the figure, and a process which uses a sentient material is shown in the right-hand side.

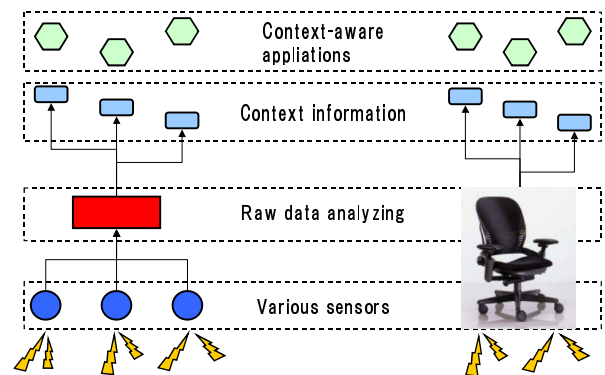


Figure 1: Context acquisition process using various sensors

However, there are many difficulties in building a sentient material, because of incorporating sensors into daily objects and using them in an appropriate way to analyze raw data and share context information among sentient materials. Moreover, a developer has to design a sentient material by considering issues which are inherent in a sensor, such as an error rate. This paper describes two prototype implementations of sentient materials, and describes some experiences with building them. Finally, we discuss how to solve the current issues by using a concept of the world model.

The structure of the paper is as follows. We express the nature of sensors in ubiquitous computing environments and the concept of a sentient material in Section 2. Section 3 describes two sentient materials we have implemented. Next, Section 4 examines difficulties in building sentient materials from our experiences. At last, Section 5 concludes the paper.

2 Sensors in Ubiquitous Computing Environments

A sensor is a device to detect all types of information and changes of energy. A sensor detects them by a phys-

ical way, chemical way and biological way. Furthermore, their output is generated as electronic signal in general. The range of monitoring subjects is very wide and complicated. Give a few example roughly, subjects to monitor are light, magnet, ultrasound, force, temperature, gas, and so on.

In ubiquitous computing environments, sensors are used to extract context information that models our real world. For example, a force sensor is used for detecting a chair's state, "How the chair is being used?". By incorporating sensors into daily materials, they can detect the state of their surrounding environments. We call the intelligent materials sentient materials. The real sentient materials may not be available soon, but we would like to consider the impact of the sentient materials to ubiquitous computing while building context-aware applications.

2.1 Related Work

Recently, several researches about sentient materials are conducted. TecO has built MediaCup, which is an ordinary coffee cup augmented with sensing, processing and communication capabilities to collect and communicate context information [3]. Sensors are incorporated to detect the movement and temperature of the cup. For detecting the movement, three metal ball-switches are incorporated to measure its orientation, and a switch in the base to detect when a cup is placed on a surface as opposed to be held or carried.

Load Sensing Lab in Lancaster University has augmented a table and floor with sensing the weight to detect the state of a surface [4]. They identified three context primitives that can be extracted from sensing the weight of surfaces. These are weight, position, and type of interaction. The experiments have demonstrated high accuracy object positioning, and the classification of interactions such as objects being placed, removed, or knocked over.

Shown in these cases, building sentient materials is one of the useful ways to acquire context information. Using sentient materials, an application can customize their behavior in a context-aware way. However, developing a sentient material and an application using sensors has many difficulties in integrating the physical world and virtual world. Contrary to a traditional application architecture, a context-aware application has to implement some stages to cut across the boundary between atoms and bits. Using sensors is the first step to acquire physical information, and the most important part of adding context-awareness to the application.

An application developer who has to acquire physical information from sensors, builds a sentient material with using self produced sensor systems or existing sensors. Especially in building a new sensor system, a developer has to learn basic electronics, and microprocessor programming for building devices. These low-level device construction and implementation make the development of sensor-based applications hard. Moreover, even if a device were constructed, there is no high-level development

platform or framework for developing sensor-based software. This means sharing information among services, appliances, and sentient materials, which is acquired by various sensors embedded in sentient material, is very difficult. We believe that a high-level development framework for software and system architecture are required. In this paper, we clarify problems for building a sentient material from experiences with building two prototype implementations.

2.2 Sensor Systems

In this section, we describe sensor systems which are used for building our sentient materials.

MICA is a platform for ad-hoc sensor networks, developed at University of California at Berkeley [5]. MICA Mote is a core module of the sensor, and it works as a node or base station in a sensor network. MICA Mote implements a CPU, a memory and a capability of radio transmission. Moreover, combining with a sensor board, MICA Mote can sense various data, such as noise, light, temperature, magnet and acceleration. Figure 2 shows the system. In a), MICA Mote modules are shown. The leftmost module is a programming board. The center module is a MICA Mote, and the rightmost module is a sensor board. In b), a MICA Mote combined with a programming board is shown.

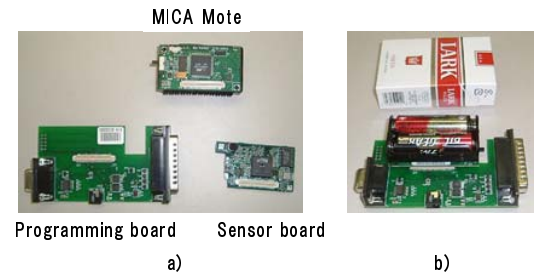


Figure 2: MICA Mote

Phidgets are devices developed at University of Calgary, that help a developer to construct physical user interfaces easily [6]. Phidgets contain physical devices (e.g. sensors, actuator) and their software so that a developer can concentrate on the overall use instead of the low-level device construction and implementation. Sensors provided by Phidgets are light sensors, force sensors, touch sensors, accelerometer, RFID tag readers and so on. Figure 3 shows some of phidgets sensors. Starting from the left, a slide sensor, a force sensor, a touch sensor and a light sensor are shown.

Phidgets are simple to program, and they are controlled through a simple API on the Windows operating system. No knowledge about hardware, microprocessors, USB, communication protocols, is necessary.

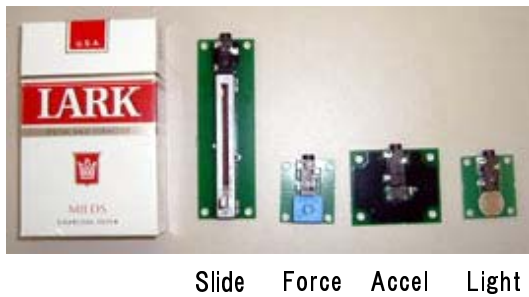


Figure 3: Phidgets sensors

3 Our Sentient Materials Implementations

We have built two sentient materials, and we describe them in this section.

3.1 SentientDoor

The first sentient material is an augmented door, which detects the motion of door closing. A MICA Mote is attached on a side, and it measures the acceleration of closing door by its accelerometer. Therefore, a door's closing motion is detected by analyzing the change of acceleration. Figure 4 shows a photo of the SentientDoor.

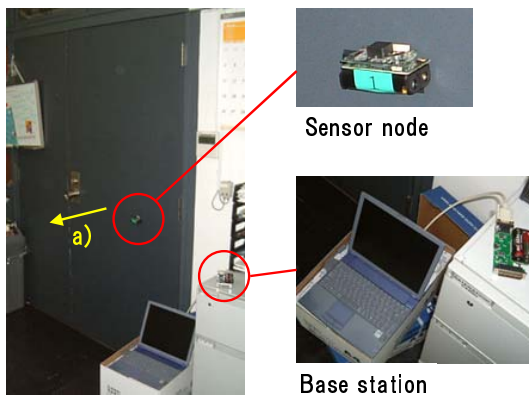


Figure 4: SentientDoor

As explaining the sentient material, a Mote combined with a sensorboard is attached on the door as a sensor node, and Mote combined with a programming board as a base station is attached to PC via a serial line (RS232C). An accelerometer on the sensorboard senses vertical acceleration to the closing direction every 250 msec. A broad arrow a) in the Figure 4 shows the direction. Five continuously sensed data are collected, and it transmits to the base station on radio. When receiving the packet, the

base station forwards data to an analyze program on PC via a serial line. Consequently, opening and closing of the door is detected.

Defining the maximum acceleration and the minimum one at the time of door opening and closing, SentientDoor detects the threshold of the maximum and minimum value. The algorithm for correcting errors caused by the vibration has been also implemented. Consequently, closing the door is detectable in the accuracy of 80%. The error is caused due to the accuracy of the sensor itself, the packet loss in radio communication, and the blur at the time of door opening and closing.

3.2 SentientCradle

The second sentient material is an augmented cradle for a cellular phone, which detects the arrival of e-mails. The pressure sensor of Phidgets is attached at the back holder, and the existence of a cellular phone and the vibration at the time of the arrival of e-mails is detected. The change of detected pressure is analyzed, and the state of the arrival of e-mails is detected. The touch sensor was attached in the front of a holder. Figure 5 shows a photo of the SentientCradle.



Figure 5: SentientCradle

As explaining the sentient material, the pressure sensor and the touch sensor are attached in the holder via Phidgets Interface Kit which is connected to PC via USB. It offers the value acquired by each sensor, which is encapsulated into the COM object. An analysis program

executed in PC gets data every 100 msec, and the existence of a cellular phone and the state of the arrival of e-mails are detected based on the acquired data. In the sentient material, events generated by the touch sensor are handled by an event-driven framework supported by Phidgets.

We have implemented the algorithm which is used for the analysis of the arrival of e-mails. It uses the amount of changes, not only the difference with the average value at the time of no vibrating, but also the amount of respective changes. Moreover, the vibration at the time of the arrival of e-mails must detect several vibrations as one arrival in a collective way, in order to consider several repeated vibrations as a defined pattern, and the pattern represents the arrival of e-mails. Consequently, the existence of the arrival of e-mails was detectable in the accuracy of 90 %. The error is caused due to the accuracy of the sensor itself and the blur of the holder.

4 Discussions

4.1 Problems with Building Sentient Material

In consequence of the two implementations, we found three problems about creating sentient materials as follows.

- Selecting appropriate sensors
- Sharing information through networks
- Extracting context about human action

In this section, we explain respective problems.

4.1.1 Selecting Appropriate Sensors

Up to now, various sensors have been developed for the purpose of a particular use (e.g. industry, medical use). They have been chosen for appropriate purposes carefully. When building sensor systems, we must use sensors whose performance is just appropriate for the purpose. If we use too highly sensitive sensors, the detection becomes incorrect. So, sensor sensitivity should not be too higher or lower. Therefore, various sensors are developed to adapt many cases. Whenever building sentient materials, we have to consider this point.

However, in fact, sensors that we can obtain are limited. The procurement of an expedient sensors may be difficult, and creating a brand new sensor for a particular purpose is expensive. So, we have to choose the most appropriate sensor from limited options, and recover the decline of performance by using ways as described later.

The relationship between sentient materials and errors changes due to the process of building them. When developing sentient materials from scratch, we can reduce this problem by designing an appropriate hardware model.

However, in the case of adding sensors to ready-made materials, errors detected from hardware will be increased and it means that the errors should be recovered by software. In particular, when the importance of one sensor is higher, the treatment of errors is more serious.

4.1.2 Sharing Information through Networks

Raw data acquired by sensors are gathered to an application through several types of networks. Sensors always monitor continuous changes of physical values, and the polling interval is differed from the granularity of data required by applications. However, many context-aware applications require milli-second interval. This means that the heavy network load between sensors and applications may cause problems. Therefore, sharing raw data acquired by sensors among applications through networks in a real-time fashion is very difficult.

On the other hand, the change of context information generated by analyzing raw data requires to be acquired within every few seconds, because considering the speed of human action. From these discussions, we should define event types fundamentally, and share highly abstract data among applications. For example, the context information which means "He is in the room" acquired by an RFID tag reader is generated from continuous raw events which means "IDXXXXX is in the range". In this case, what we should share is not raw data but context information so that it is possible to reduce the network load. Additionally, the logical/physical distance between a sensor and a data analyzing part is connected via unreliable networks, and the loss and delay of data through networks will be increased.

4.1.3 Monitoring Human Action

It is difficult to guess human action from information acquired by sensors. There are two reasons for this. The first reason is that abstraction level of context information acquired from one sentient material is too low. For example, monitoring a door's closing motion cannot be detected whether he just opens the door or he comes out of the room. To detect the door's motion, we need to combine several sensors. This fact means the limitation of context information which can be extracted from one sentient material. The other reason is the lack of techniques to analyze and use raw data. The difficulty of choosing appropriate sensors and analyzing raw data became big problems when we extracting context information.

4.2 Discussion about Using Sentient Materials

After taking into consideration these problems, we discuss about how to build a sentient material and how to share context information among applications. From the result of the current implementations, we found three important aspects of sentient materials. These are, "HW : a physical element e.g. sensors", "SW : a logical element e.g.

data analyzing parts”, “NW : an element to implement information sharing”. These elements are closely related to each other. Even if the performance of one element is enough, the entire system may not work well. Therefore, we should consider these elements with sufficient balance to build sentient materials.

As mentioned in Section 4.1, SW should implement an algorithm specialized in each sentient material, because of correcting errors underlying in HW (e.g. the accuracy of sensor, physical properties of material). Therefore, each sentient material implements SW specialized for HW, and generates particular context information in its system.

However, as mentioned in Section 4.1.3, the abstraction level of context information generated in each one sentient material is low. Additionally, as mentioned in Section 4.1.1, some critical errors which SW cannot correct well, because the accuracy of sentient materials depends heavily on one sensor. Therefore, we propose integrating multiple sentient materials using the world model which is a subset of the real world created in a virtual world as a platform for context-aware applications. Hereby, it becomes possible to generate another high level abstract context information by combining raw data collected by multiple sentient materials, and by correcting context information with multiple sentient materials collectively.

4.2.1 World Model : Platform for Context-aware Applications

The world model is a platform for a context-aware application in ubiquitous environments. It is built in the virtual world by modeling our real world. For example, NEXUS which is developed at Stuttgart University [7] models physical entities such as rooms and buildings as software objects. A set of the objects constructs a real world model based on location information acquired by GPS. Using an object-oriented model enables a developer to extend the original schema to special one for each use.

As one of the advantage of building the world model, it plays a role of the common platform for various applications. The world model facilitates an application which combines the real world and the virtual world making the states of the real world available to programs. Moreover, synchronizing information coherently facilitates the cooperation among applications. From the viewpoint of the application development, a context-aware application makes services simpler.

With the concept of the world model, integrating multiple sentient materials as an object is effective to coordinate them. The approach makes it easy to share context information. Therefore, it is possible to correct and extend context information acquired by each sentient material. In the next section, we describe the design of sentient materials as objects in the world model.

4.2.2 Sentient Material on the World Model

To design the relationship between sentient materials and the world model, we should consider about the difficulty

of sharing raw data directly. In this step, it is important to make clear the relationship between sensors and sensed object i.e. sentient materials. As mentioned before, it is desirable to share information at the highly abstract level, and the information needs not be updated within a short period. Therefore, the information can be encapsulated into objects that model physical entities. An application can share and use the information about sentient materials only to make a query to inquire an object’s state. Figure 6 shows an overview of integrating sentient materials with the world model.

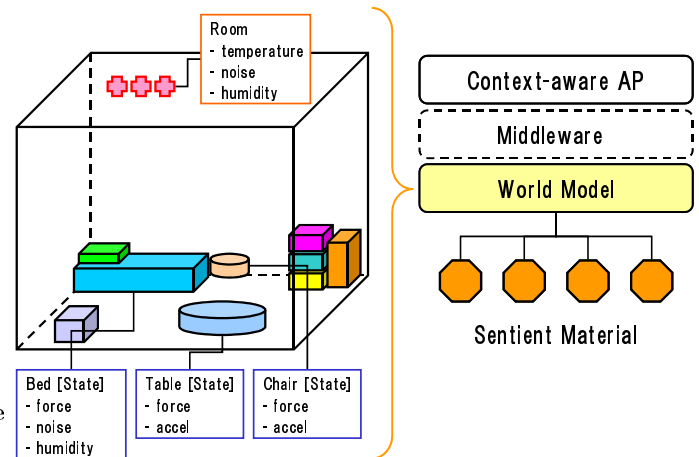


Figure 6: Sentient Material on World Model

However, it is also useful to share raw data. Applications do not use an object-specific state e.g. a chair’s “sitting” or “free”, but states that can be shared by several objects like “temperature” or “humidity”. In this case, what is acquired by applications is raw data, and generated context information (e.g. warm or cold) is differed in respective applications.

At this point, the following list summarizes the relationship between sensors and objects.

- Object-dependent sensors
 - a) provide raw data to generate low level abstract context information.
 - b) provide raw data to generate high level abstract context information.
- Object-independent sensors
 - c) provide raw data which present the implicit state of a space.

Then, we explain how to process raw data for each case. Figure 7 shows raw data processing in object-dependent sensors.

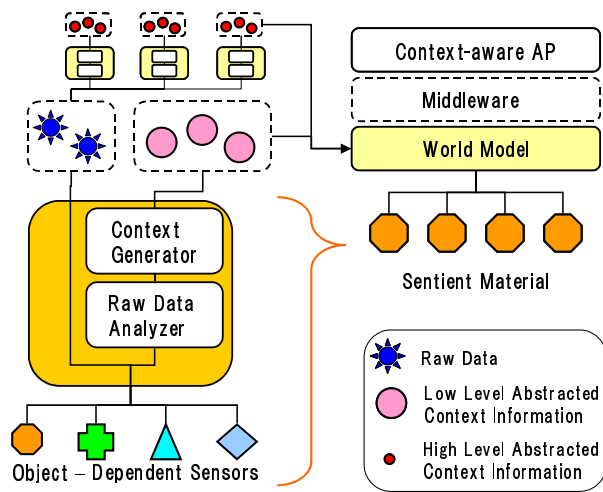


Figure 7: Raw data processing of object-dependent sensors

a) Raw data acquired by object-dependent sensors are processed directly in the raw data analyzer. Next step, the context generator extracts context information. In this case, raw data are acquired only by the raw data analyzer belonging to the object. Thus, it decreases data losing and delay caused by passing through networks. Also, raw data are processed in a distributed way, and the approach makes a system simple.

b) In another case, applications use not only context information but also raw data to generate context information. In this case, it is required to share raw data across networks.

c) Then, raw data of object-independent sensors are required by various raw data analyzers. It is also necessary to share raw data through networks.

Subsequently, we should consider the coordination of the sentient materials. As described above, we can correct errors by gathering information from multiple sentient materials in the world model. To facilitate the mechanism, we have to implement rules which define relationship among sentient materials by the physical law. For example, a sentient chair which has only one force sensor and detects its state by using a threshold value, will think "Someone is sitting me.", although there is only a bag on it. In this case, if other sentient materials (e.g, desk, cup...) show "No one is using me.", the world model would correct the chair's state by the rules.

The rules can be used not only to correct errors, but also to extend the capacity of sentient materials. As described in 4.1.3, context information acquired from one sentient material is too low. However, we can acquire higher abstracted context information by the rules which associate multiple sentient materials. From this point of view, our project also investigates context information inference [9].

This bottom-up approach enables us to build sentient materials for a specific domain and to minimize the size of the infrastructure. Moreover, considering them as a module and associating them by the rules on the world model, we can reuse them and reconstruct the world model created for a specific application domain. The world model makes it easy to deal with context information, and allows applications to access the information at an abstract level. This makes us to develop software for sentient materials dramatically easy.

5 Conclusion

In the paper, we have described sentient materials, which are a new form of materials in ubiquitous computing environments. Sentient materials can detect their states by sensing their surrounding environments. Then, they offer the new usage of a sensor in our daily lives, and have described various difficulties about context acquisition.

We have built two sentient materials and clarified problems of building them. These are a difficulty in selecting appropriate sensors, a difficulty in sharing information through a network, and a difficulty in extracting context about human action. When considering solutions based on these problems, we have observed three important aspects of sentient materials. These are SW, HW and NW. These elements are closely related each other. Therefore, we need to consider these elements with a sufficient balance to build sentient materials. We have also introduced the concept of the world model, and have discussed about building sentient materials as an object on the world model.

References

- [1] W. Butera, "Programming a Paintable Computer", MIT Ph.D Thesis, 2002.
- [2] L.H.Lifton, "Pushpin Computing: A Platform for Distributed Sensor Networks", MIT Master Thesis, 2002.
- [3] Michael Beigl, "MediaCups: Experience with Design and Use of Computer-Augmented Everyday Artefacts", Computer Networks, Special Issue on Pervasive Computing, Elsevier, 2001.
- [4] Albrecht Schmidt, "Context Acquisition Based on Load Sensing", Ubicomp 2002
- [5] "Berkeley WEBS", <http://webs.cs.berkeley.edu/>
- [6] "Phidgets.com", <http://www.phidgets.com/>
- [7] Nicklas, "The Nexus Augmented World Model: An Extensible Approach for Mobile, Spatially-Aware Applications", Proceedings of the 7th International Conference on Object-Oriented Information Systems: OOIS '01
- [8] T.Shoji, N.Nakamura, K.Fujinami, and T.Nakajima, "Software Infrastructure for Sentient Materials", In Proceedings of the IEEE Workshop on Software Technologies for Future Embedded Systems, 2003.
- [9] Kaori Fujinami, Tetsuo Yamabe, Tatsuo Nakajima; "Take me with you!": A Case Study of Context-aware Application integrating Cyber and Physical Spaces, In Proceeding of ACM Symposium on Applied Computing(SAC) 2004, Nicosia, Cyprus, March, 2004.