MAPLE: Multi-Agent Programming with Letter Exchanges on Sensor Networks

(Extended Abstract)

Tiffany Yi-Ting Tsao
Department of Computer Science and Information Engineering
Nation Taiwan University
yiting.tsao@gmail.com

Wan-rong Jih
Department of Computer Science and Information Engineering
Nation Taiwan University
wanrong.jih@gmail.com

Jane Yung-jen Hsu
Department of Computer Science and Information Engineering
Nation Taiwan University
yjhsu@csie.ntu.edu.tw

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent system*

General Terms

Design

Keywords

multi-agent system, sensor network, service composition

1. INTRODUCTION

Developing reusable and flexible sensor networks is an important task for context-aware applications. Previous researchers have deployed several applications on sensor networks [3, 5]. These sensor networks are consisted of various sensor nodes to monitor and control the environment. Since these sensor networks are consisted of various sensor nodes, developers need to take sensor deployment and application design into consideration at the same time to achieve their goals. It is a great challenge for developers to deal with low-level sensor controls while designing policies to interact with users. The flexibility of sensor networks is inherently limited by its design. Therefore, to make developers focus on application design without concern for any hardware issues is very important.

Some researchers have proposed the middleware solution to improve the flexibility of sensor network [4, 7]. By building a virtual machine on top of each sensor node, it is easy for developers to program the nodes using a predefined instruction set. The provided instruction set is an assembly-like language which brings limited assistance to developers. OASiS is a programming framework for middleware solution [6]. It treats the unit of application functionality as a service so that developers can compose the services to achieve their goals. However, the services in this framework still need to be redesigned when the application changes.

In order to improve the reusability of sensor networks, this paper proposes *MAPLE (Multi-Agent Programming with Letter Exchanges)*, a multi-agent architecture, to manage dynamic sensor networks. Each agent in MAPLE provides specific service, such as sensor network maintenance, environment perception, device control, service discovery, service composition, and context reasoning. Each service corresponds to an instruction. Then, the agents can easily use the services by wrapping the instructions as letters and sending them to others.

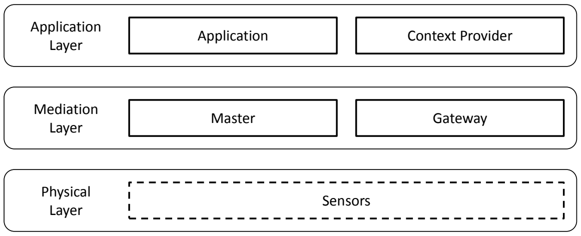

Figure 1: MAPLE: a multi-agent sensor network architecture. The rectangles with solid borders are agents.

2. DESIGN

The MAPLE architecture includes three layers (see Figure 1). Rectangles with solid borders represent the agents. All practical sensors with different abilities are used in the physical layer. The mediation layer is a communication interface to simplify the procedure for fetching sensor data and controlling actuators. In the application layer, the agents interact with end users and provide high-level services.

Gateway

Gateways are the managers that supervises sensors with the following instructions.

INSERT adds a sensor profile to an unused port when a new sensor is attached. For example, a user plugs in a motion sensor to the analog port 3, places it on the desk, and sets its sensitivity level to 10.

```
INSERT PORT="A.3" TYPE="MOTION" SENSITIVITY="10"
LOCATION="DESK"
```

UPDATE refreshes the profiles when the physical deploy-

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

ment changes. For example, a user replaces the motion sensor on the analog port 3 with a sound sensor and all other settings remain the same.

```
UPDATE PORT=”A.3” TYPE=”SOUND” SENSITIVITY=”10”
LOCATION=”DESK”
```

DELETE removes the stale profiles when sensors are broken accidentally. For example, a dog bites the sound sensor on the analog port 3 and then the broken sensor will be removed.

```
DELETE PORT=”A.3” TYPE=”SOUND” SENSITIVITY=”10”
LOCATION=”DESK”
```

READ acquires electrical signals from sensors. For example, a user retrieves the data from the analog port 3 every 5 seconds.

```
READ PORT=”A.3” INTERVAL=”5000” RECEIVER=”USER”
```

WRITE controls the interaction with actuators in the real world. For example, a user sends a signal to the digital port 0, and then the corresponding device will respond immediately.

```
WRITE PORT=”D.0” INTERVAL=”-1” RECEIVER=”Tester”
```

Master

Master provides two services, SELECT and SET, to search and compose the other services on the same sensor network.

SELECT searches available services that fit the given constraints. For example, retrieving the agents with a motion sensor in the living room.

```
SELECT NAME=”MOTION” LOCATION=”LIVING ROOM”
```

SET encapsulates the steps for achieving a given context or gathering context information. For example, gathering the luminance state on the desk every 5 seconds.

```
SET GOAL=”LUMINANCE” INTERVAL=”5000” LOCATION=”DESK”
```

Context Provider

Each context provider infers contexts from the sensor data. For example, the motion level provider infers the status from the given motion data.

```
INTERPRET MOTION=”400”
```

Application

Each application provides reminder or assistance to users. For instance, the auto lighting application controls the lamp on the ceiling according to the motion level at the desk.

```
SUBSCRIBE LOCATION=”CEILING,DESK”
```

3. IMPLEMENTATION

To demonstrate the usability of the MAPLE framework, we build an application to lighting a lamp when the motion level in the office is high. Several different sensors are deployed in a personal office. We integrate PL-PLAN [2] planner into the JADE [1] platform to control sensors and provide the instructions/services. In line 1- 2, the application requests the motion level and the lamp state. After the application makes a decision according to its build-in rules, it sends a request to control the lamp in line 7.

Algorithm 1 The behavior of the auto lighting application

```
1: SET GOAL=”MOTIONLEVEL” INTERVAL=”1000” LOCATION=”DESK”
2: SET GOAL=”LAMPSTATE” INTERVAL=”1000” LOCATION=”CEILING”
3: while true do
4:     // Receive the contexts
       ...
5:     // Make a decision
6:     if MOTIONLEVEL = HIGH and LAMPSTATE = OFF then
7:         SET GOAL=”LAMP” INTERVAL=”-1” LOCATION=”CEILING”
8:     end if
9: end while
```

4. CONCLUSION

Sensor network programming is a very challenging task because application developers need to design not only the sensor deployment but also their control strategies. In this paper, we propose the MAPLE framework to simplify the programming effort. It defines different agent roles to provide a variety of services in sensor networks, such as sensor management and context inference. Therefore, application developers can leverage the available services to design applications without worrying about the hardware. To summarize, MAPLE can not only manage dynamic sensor networks but also help application developers reuse the services in sensor networks.

5. ACKNOWLEDGMENTS

This work was also supported by National Science Council, National Taiwan University and Intel Corporation under Grants NSC99-2911-I-002-201, NSC 100-2911-I-002-001, 10R70500, and 10R70501.

6. REFERENCES

[1] JADE - Java Agent DEvelopment Framework. http://jade.tilab.com/.

[2] PL-PLAN. http://plplan.philippe-fournier-viger.com/.

[3] D. Christin, P. Mogre, and M. Hollick. Survey on Wireless Sensor Network Technologies for Industrial Automation: The Security and Quality of Service Perspectives. *Future Internet*, 2(2):96–125, 2010.

[4] C. Fok, G. Roman, and C. Lu. Agilla: A Mobile Agent Middleware for Self-Adaptive Wireless Sensor Networks. *ACM Transactions on Autonomous and Adaptive Systems*, 4(3):1–26, 2009.

[5] B. Kim, S. Hong, Y. Jeong, and D. Eom. The Study of Applying Sensor Networks to a Smart Home. In *Proceedings of the International Conference on Networked Computing and Advanced Information Management*, 2008.

[6] M. Kushwaha, I. Amundson, X. Koutsoukos, S. Neema, and J. Sztipanovits. OASiS: A programming framework for service-oriented sensor networks. In *Proceedings of the International Conference on Communication System Software and Middleware*, 2007.

[7] P. Levis and D. Culler. Maté: A Tiny Virtual Machine for Sensor Networks. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, 2002.