

Utilizing Descriptive Documents for Adaptive and Reconfigurable M2M Systems

I-lung Tsai, Wan-rong Jih, Yen-Ling Kuo and Jane Yung-jen Hsu
Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan
{r99944030, wrjih, r98922037, yjhsu}@csie.ntu.edu.tw

Abstract—Machine-to-Machine (M2M) has becomes an important research topic in the recent years. There are application choices for users, but these applications cannot exchange information with each other. Applications for different purposes cannot share the same devices, because the providers are different. Therefore, the users need an integrated framework that can easily configure devices and adapt to the user requirements. We propose three descriptive documents to facilitate the achievement of reconfiguration and adaptation. Results demonstrate our framework can seamlessly manage devices and promptly adapt to the environmental changes.

Keywords—M2M, reconfiguration, adaptation

I. INTRODUCTION

As an increasing number of various computing elements, researches in Machine-to-Machine (M2M) are growing popular. Many applications focus on enabling machines to equip with capabilities of automatic communication, coordination and configuration. These systems were developed independently without considering the integration of applications. Each application has different infrastructure and communication standard. Even for some applications using the same types of devices, they still cannot communicate with each other. As a result, it is hard to configure devices in order to adapt to different applications. Consequently, a system with capabilities of reconfiguration and adaptation is a very important issue for M2M application development.

Gilman Tolle [1] proposed a sensor network management system to manage the resource and query system. Chien-Liang Fok [2] et al. introduced Agilla to propagate the M2M applications without prior installation. Manish Kushwaha [3] et al. initiated OASiS for integrating the ambient contexts into services. In 2011, Stephen Dawson-Haggerty [4] et al. presented sMAP which used the idea of profiles to manage device information and design service applications. We utilize the advantages of previous researches and propose three descriptive documents to achieve a reconfigurable and adaptable system.

II. DESCRIPTIVE DOCUMENTS FOR USERS

The three descriptive documents as well as three different users are utilized for implementing a reconfigurable and

adaptive system. We present the three documents in the following subsections. The three users are installers, developers and end-users. An installer is a hardware installer who owns professional knowledge and installation skills in hardware. A developer is a software developer who has computer programming experience and can implement software applications. An end-user is the person who will use the system.

A. Profile

A profile describes attributes of hardware devices. These attributes are obtained from hardware specifications and are easy maintain by an installer. In this paper, there are three kinds of profile: *sensor profile*, *actuator profile*, and *node profile*.

The sensor profile describes the specifications of environmental sensors. Attributes of sensor profile are ID, name, output signal type, valid data range, allowed sample rate, and so on. Every sensor has unique identity and works under the specified environment for sending the sensing data. Output signal can be either digital or analog.

The actuator profile contains ID, name, input voltage, power in watts, and so on. Contents of sensor and actuator profile are similar; most of their attributes are the hardware requirements of devices, but the data and control access direction are different. Sensors provide environmental contextual data whereas actuators reflect the control of devices.

The node profile describes a special device that connects sensors and control actuators. Accordingly, attributes of node profile are ID, mobility, the connected sensors, the affected actuators, and so on. A node could be desktop, single-board microcontroller, smart phone, etc. Any computing-capable device, either stationary or portable, could play the role of a node. Therefore, the node profile has an attribute to describe the mobility of nodes. Moreover, contents of node profile declare the communication methods between the node and sensors/actuators, such as the communication protocol, channel or port, and so on. Communication protocols that we used are SPI, USB, UART, Wi-Fi, ZigBee, and Zwave.

B. Objective

Developers have to determine the objectives of a system. For example, in a meeting room, the objectives may include

climate, lighting and ventilation, and privacy, etc.

After defining the objective, developers have to declare the relationship between devices and objectives. Purpose of this relationship is to establish the connection between objective and sensors/actuators. For example, Table I shows temperature sensors and climate are related to each other. In addition, Table I presents that the curtain can affect the lighting and privacy; the fan has connection with climate and ventilation.

Table I
SEMINAR RELATION TABLE

Device	Type	Climate	Lighting	Ventilation	Privacy
temperature	sensor	✓			
humidity	sensor	✓			
light	sensor		✓		
curtain	actuator		✓		✓
fan	actuator	✓		✓	

C. Scenario

End-users create scenarios to describe their needs and preferences. A scenario is the situation an end-user may encounter. Table II denotes contents of scenarios that are the weight of each objective. For example, weights of the empty scenario are 15% climate, 0% lighting and 15% ventilation.

Table II
AN EXAMPLE OF SCENARIOS

Scenario	Climate	Lighting	Ventilation
Empty	15%	0%	15%
Quiet Work	60%	60%	60%
Presentation	95%	20%	95%
Discussion	90%	85%	65%

A scenario will switch to another scenario according to the changes of environmental contexts, which are detecting by sensors. For example, according to the scenarios in Table II, if a person entered an empty meeting room and turn on the projector, the scenario will switch from empty to presentation.

Figure 1 illustrates the integrated view of actuators, objectives and scenarios. The left-hand side represents the relationship between devices and objectives, each associate with a weight. The right-hand side of Figure 1 visualizes the associated relationship. As a result, if the scenario is changed, the different associated relationship will be applied.

III. RECONFIGURATION AND ADAPTATION

Figure 2 is a stack diagram that describes the components of M2M system architecture. In the hardware layer, profiles depict specifications of devices. In the middle layer, user interfaces and software agents utilize the three descriptive documents. Agents coordinate with each other to achieve

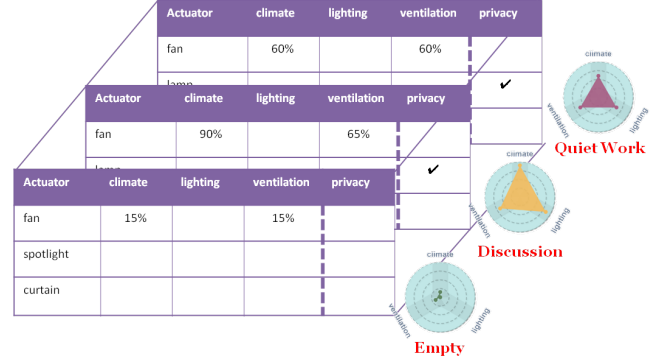


Figure 1. Profile, Objective and Scenario

the goals of current scenario and the associated objectives. The user layer shows that the user interface allows users interact with the system.

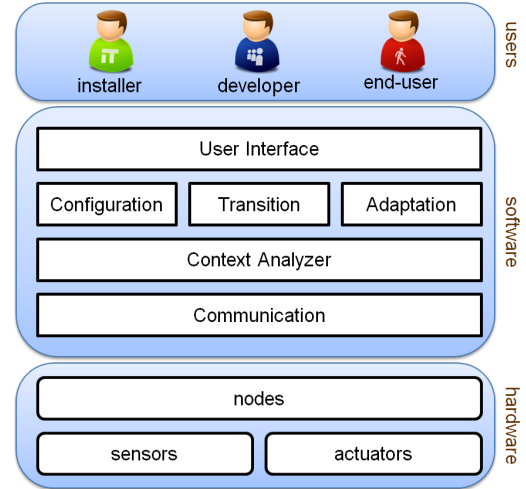


Figure 2. Stack Diagram

If the configuration agent and adaptation agent need to access devices, the agent will send request to the communication agent. Table III lists the allowed requests of the communication agent.

Communication agent sends the request to a node, which connects to sensors and controls the operation of actuators. The first three requests, *assign*, *delete*, and *reset*, are designed for maintaining nodes. The *insert* and *remove* requests maintain the connection of sensors/actuators on the node $\langle node \rangle$. Request *read* is to sense the data from a sensor and the sensor is attached to a port $\langle port \rangle$. Request *write* is to control the actuator to execute the specified operation. The last two requests explain that agents do not directly access the sensors and actuators, but access through the node devices. These communication requests provide flexibilities of accessing devices and are easy to configure a

Table III
REQUESTS FOR ACCESSING DEVICES

Command	Description
$\langle node \rangle$ assign	Create a new node
$\langle node \rangle$ delete	Delete a node
$\langle node \rangle$ reset	Reset a node
$\langle node \rangle$ insert $\langle port \rangle : \langle device \rangle$	Add a new sensor/actuator to the specified port of the node
$\langle node \rangle$ remove $\langle port \rangle$	Remove a device from the port
$\langle node \rangle$ read $\langle port \rangle$	Receive sensor data from the specified port
$\langle node \rangle$ write $\langle port \rangle : \langle op \rangle$	Control an actuator via the specified port to perform the operation

M2M system.

The context analyzer is an agent that collecting and analyzing the contextual sensing data. After computing the contexts from context analyzer, transition agent decides the switch between various scenarios. Given a scenario, adaptation agent use genetic algorithm to determine the control strategies of actuators. As the problem encoding, attributes of an actuator is the gene of a chromosome. Attributes include the operating state of the actuator, energy consumption, contents of actuator profile, etc. Given a set of actuators A and objectives O , Equation (1) is the fitness function of adaptation agent.

$$\min \sum_{a \in A} \left[\alpha \times \phi_a + (1 - \alpha) \sum_{o \in O} \beta \times \phi_o \right], \quad (1)$$

where ϕ_a is the disparity in energy consumption between current operating state of actuator $a \in A$ and the one defined by the scenario. The ϕ_o plays the similar role as ϕ_a , except the target is for the objective o . In this paper, objectives are climate, lighting, and ventilation, etc. Values of coefficients α and β are between 0 and 1. In Equation (1), the first term denotes the evaluation of energy consumption and the second term represents the effects of objectives. The goal of adaptation agent is to find a control strategy that satisfies preferences of the scenario and saving energy.

IV. EXPERIMENTS

A. Reconfiguration Demonstration

Figure 3 depicts the user interfaces of our system. We deploy sensors in a meeting room and use Arduino¹ to collect raw data of sensors and control actuators. The round dot represents the Arduino node. A capital letter in a small square stands for a sensor, includes light (L), temperature (T), humidity (H), motion (M), CO2 (C) and sound (S) sensors.

A rectangle box in Figure 3 describes the connected devices of a node. There are two types of devices, one is sensor and the other is actuator. The node in Figure 3 has ID 0003; it connects five sensors and one actuator. Sensor profile in Figure 4 shows that the sensor ID 0011 is a temperature sensor, ID 0012 is a humidity sensor, ID 0013 is a light sensor, and vice versa. The actuator profile in Figure 4

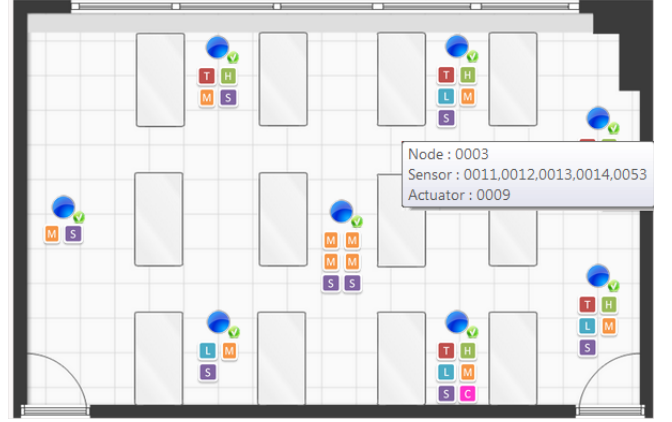


Figure 3. Sensors Deployment in a Meeting Room

indicates that actuator ID 0009 is a speaker. Both sensors and actuators are located at location 0003.

Sensor ID	Sensor Name	Type	Valid Range	Location
0011	temperature	digital	-40~125	0003
0012	humidity	digital	0~100	0003
0013	light	analog	0~1023	0003
0014	motion	digital	0~1	0003

Actuator ID	Actuator Name	Voltage	Power	Location
0007	lamp	110	15	0004
0008	lamp	110	15	0004
0009	speaker	110	12	0003
0010	projector	110	250~575	0003

Figure 4. Actuator and Sensor Profiles

If installers have to change devices, it is not necessary to stop the system. They just need to do is to modify the corresponding profile. In order to ensure the correct

¹Arduino <http://www.arduino.cc/>

connection of a node, installers also need to modify the associated contents of node profile, which declares the sensors and actuators are connecting to the node.

Figure 5 is the user interface for creating a new scenario. The three objectives, climate, lighting and ventilation, are

Figure 5. An Example of Creating Scenario

predefined by the developer. Which objective will be chosen will depend on the requirements of systems.

In the upper part of Figure 5, the end-user can assign a weight to each objective. For example, the weight of climate is 90%, lighting is 85%, and ventilation is 65%. As the lower part of Figure 5, it specifies the preferred target for each objective, values of the target is a small portion of the valid range.

B. Adaptation Evaluation

We identify four scenarios for the meeting room, including empty, quiet work, presentation, and discussion. Only the empty scenario states for non-human activities, the other three are human-related activities. It takes time to detect the scenario change and control the corresponding actuators. We present the response time for dealing with scenario changes in Figure 6.

The results are consistent with the normal human activities time. There are very few midnight activities and most activities are held during the daytime. It shows the average response time is no more than 2 seconds and even the worst one is within 10 seconds.

V. CONCLUSION

In order to solve reconfiguration and adaptation problems in M2M applications, we propose three descriptive documents to describe the devices information and user needs.

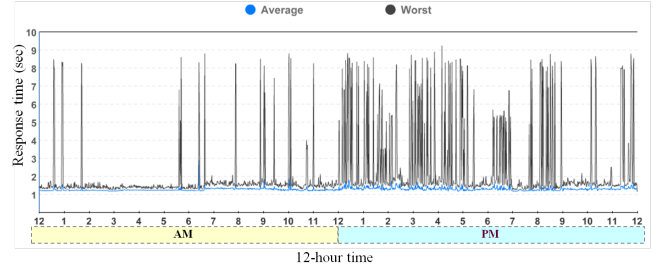


Figure 6. Response Time for Scenario Changes in One Day

Agents use these documents to reconfigure the parameters of devices and adapt to actuators to environment for different scenarios. We deploy sensors and actuators in a meeting room. Contents of the three documents are based on the need of students. Experimental results showed that we can find the appropriate control strategies in 10 seconds. However, it is a small-scale experiment, we need to explore more types of spaces and deploy more sensors and actuators in the space.

REFERENCES

- [1] G. Tolle and D. Culler, "Design of an application-cooperative management system for wireless sensor networks," in *Proceedings of the 2nd European Workshop on Wireless Sensor Networks (EWSN 2005)*, ser. EWSN '05, January 2005, pp. 121–132.
- [2] C.-L. Fok, G. C. Roman, and C. Lu, "Rapid development and flexible deployment of adaptive wireless sensor network applications," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS 2005)*, ser. ICDCS '05, June 2005, pp. 653–662.
- [3] M. Kushwaha, I. Amundson, X. Koutsoukos, S. Neema, and J. Sztipanovits, "Oasis: A programming framework for service-oriented sensor networks," in *Proceedings of the 2nd International Conference on Communication Systems Software and Middleware (COMSWARE 2007)*, ser. COMSWARE '07, January 2007, pp. 1–8.
- [4] S. Dawson-Haggerty, X. Jiang, G. Tolle, J. Ortiz, and D. Culler, "smap: A simple measurement and actuation profile for physical information," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys 2010)*, ser. SenSys '10, 2010, pp. 197–210.