Flexible Mutual Exclusion for Information Appliances in Multi-user Context-aware Service Environment

Takuro Yonezawa[1], Naohiko Kohtake[1], Kazunori Takashio[1], and Hideyuki Tokuda[1]

Graduate School of Media and Governance, Keio University
5322 Endo, Fujisawa, Kanagawa, 252 8520, Japan

Abstract. With context-aware computing, a user can control appliances automatically based on his/her context. Automatic control generates conflict of demands. Thus, mutual exclusion is needed for the appliance which cannot fill the demands simultaneously. We show the technique of flexible mutual exclusion based on the relationship among users and the place where the appliance exists.

1 Introduction

With context-aware computing [1] [2], a user can define automatic control of the appliances based on his/her context. However, if multi users define automatic control, conflict of demands will happen. Conflict resolution and mutual exclusion are needed to manage appliance which cannot treat multiple demands simultaneously (e.g. light, television or CD player). It is a great burden for users to decide which demand to execute whenever conflict occurs. Moreover, the number of users and appliances get larger, user cannot figure out whose demand is conflicting on which device. Thus, the appliances should execute mutual exclusion automatically. We show flexible mutual exclusion framework based on relationship among users and place which appliance exists.

2 Mutual Exclusion Framework

Mutual exclusion framework we propose is composed of *MPDP:Multi-level Priority Determination Policy* and *LPDM:Location-based Policy Distribution Mechanism*. Appliance executes mutual exclusion with MPDP which distributed by LPDM. MPDP is written by the administrator of the appliances because it should reflect the administrator's will.

MPDP: MPDP describes relationship among users as a group hierarchically and apply the flexible rule of mutual exclusion to each group. The structure of MPDP is shown in Figure 2. MPDP is composed of 3 elements(*User*, *Group* and *Rule*) basically. *Group* is a logical set composed of *User*s. *Group* can include other *Group*s, and it is possible to describe its' relation hierarchically. *Rule* is the way to determine demand which should be executed. *Rule* is deployed to each group. The administrator can use various flexible *Rule*s (e.g. the rule which uses context). If a conflict occurs, appliance refers MPDP and searches the deepest *Group* which includes all *User*s who require automatic control. And appliance executes a demand following the *Rule* of the *Group*. The reason of using *Rule* of the deepest *Group* is that the deepest *Group* indicates relationship among users the most specifically.

LPDM: To configure MPDP to many appliances manually costs very high. Therefore the effective way of distribution is needed. The place which appliance exists influences the policy greatly. For example, appliances in kitchen are expected for mama to give priority. LPDM distributes MPDP to the appliance based on the place where the appliance exists. The administrator defines the connection between the policy and the place on LPDM server. LPDM server recognizes the place where the appliance exists by using location sensor (e.g. with RF-Tag or ultrasonic sensor). The administrator can also configure a static policy to appliance as well as DHCP system.

The structure of mutual exclusion system with MPDP and LPDM is shown in Figure 2. LPDM server recognizes the place of appliances and distributes MPDP. Mutual exclusion system on appliances resolves the conflicts and realizes mutual exclusion based on MPDP. We implemented a prototype with JAVA. MPDP is written in XML. Various *Rules* of MPDP are implemented as classes of JAVA. We used RF-Tags [3] to recognize where the application exists.

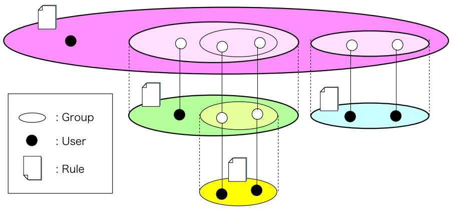

Fig. 1. Structure of MPDP

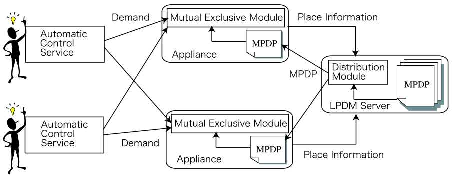

Fig. 2. Structure of mutual exclusion system

3 Discussion and Conclusion

The most flexible way is to write all cases as policy such as ”If user A and B conflict, B has a priority. If B and C conflict, C has a priority.”. However, this is not a practical way because to write all cases is a great burden. Another way is to write a group-based policy such as ”Group *Students* includes A, B and C. If conflicts occur in ’Students’, user who requires first has a priority.”. Though this way can describe relationship among users abstractively, if users belong to several group, it can’t be decided which group’s rule to use. As well as the group-based approach, a role-based approach has the same problem. MPDP can decide the rule to use based on the group which reflects the relationship among users the most specifically. Moreover, this framework is also useful for management because LPDM realizes easy and effective deployment of MPDP.

We showed the mutual exclusion framework for information appliances which be controlled automatically in this paper. With this framework, flexible mutual exclusion based on the relationship among users and the place where appliances exists will be realized. As next step, we will evaluate effectiveness of this framework through operation of the prototype in our laboratory and use feedback of evaluation into the next design.

Acknowledgment: This research has been conducted as part of Ubila Project supported by Japanese Ministry Internal Affairs and Communications.

References

1. Dey, A.K. et al. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications, *HCI Journal 16(2-4)*, 2001, 97-166.
2. Zhang, T. An Architecture for Building Customizable Context-Aware Applications by End-Users, *Pervasive 2004*, April 18-23, Linz / Vienna, Austria, 2004.
3. RF Code - Real-Time Asset and Personnel Identification, Tracking and Location Solutions Utilizing Active RFID Technology:`http://www.rfcode.com/`