RM-012

Recommender System for Device Sharing in Ubiquitous Environments

Park Jong-Hyun †, ‡　　Sachio Hirokawa ‡

Abstract

Ubiquitous computing enhances computer use by making many computers available throughout the physical environment, while making them effectively invisible to the user. Invisible devices and software are connected to one another to provide convenient services to users [1, 2]. In this environment, users want to get a variety of services by using only an individual mobile device which has limitations such as tiny display screens, limited input, and less powerful processors. In other words, users hope to obtain a personalized service composed of user-suitable devices among sharable devices in a ubiquitous smart space which is called USS in this paper. However, the situations of each user are different and user preferences also are various. Therefore, although users request the same service in the same USS, the most suitable devices for composing the service are different for all users. For satisfying these user requirements, this paper proposes a device recommender system which infers and recommends user-suitable devices for composing a user required service. The objective of this paper is the development of the systems for recommending devices through context-aware inference in peer-to-peer environments. For this goal, this paper considers the environmental context and user preference and uses two kinds of reasoning. One is a context-based reasoning for applying an environmental context and the other is a user-based reasoning for user preference. We also implement a prototype system and show the scenario using a real ubiquitous mobile object (UMO).

1. Introduction

In the ubiquitous computing environment, the needs for service composition and collaboration between heterogeneous devices are required and an infrastructure that supports the requirements will be the essential factor for a seamless service delivery [3]. In other words, users in the ubiquitous environments expect that get a variety of services. However user mobile objects are light-weight and have limited devices such as tiny display screens, limited input, and less powerful processors. Therefore the users want to get a customized service composed of sharable devices in the USS in which the users are. For example, a pointer in a meeting room hope that his (her) mobile object searches devices for composing "Presentation" service and provides the customized service through a context-aware.

For these requirements, intelligent mobile objects have to act as a mediator between users and services. Figure 1 shows an environment in presentation room in which some sharable devices are prepared for supporting a variety of presentation. In that environment, if a user who is a pointer wants to present something, then he has to select and connect some devices. However it is not easy to select customized devices for users' presentation because the user does not have any information about devices or presentation environment. Therefore the user hope to get a service composed of customized devices for his/her presentation like Figure 2.

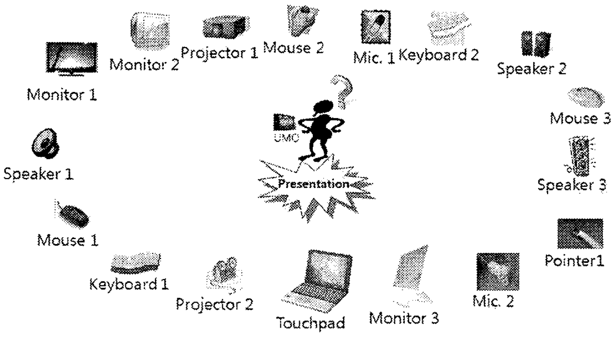

Figure 1. The environment in presentation room.

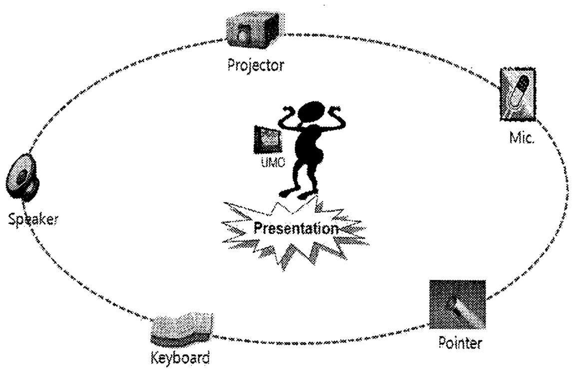

Figure 2. The "Presentation" Service Composed of Customized Devices.

The purpose of this paper is to develop the recommender system which automatically infers and recommends customized devices in a device sharing environment like Figure 1. Our recommender system is executed in a UMO (Ubiquitous Mobile Object) like a PDA and a mobile phone. The UMO has one or more sharable device and can share their devices with other UMOs by communication like Figure 3.

In the case of Figure 1, if the user requests a "Presentation" service into users' UMO (Ubiquitous Mobile Object), then the UMO searches devises by interacting with other UMOs and composes the service connected with selected devices. In order to provide these services by device sharing, however, the UMO have to infer which devices are needed for composing each service.

† Chungnam National University
‡ Kyushu University

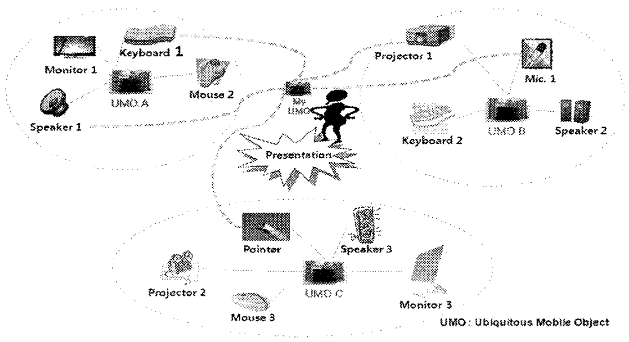

Figure. 3. Communication and Connection among UMOs

For this goal, we should consider the following issues;

(A) How to relate user-request services and devices needed for composing the requested services?

(B) How to reflect environmental context and user preference for the device recommendation?

(C) How to apply the dynamic change of user preference?

For each issue, this paper uses two step reasoning. One is context-based reasoning(CR) and the other is user-based reasoning(UR). The CR uses ontology to define the concept and relationship between service and devices. The CR and UR also use rule for reflecting environmental context and user preference, respectively. In the UR, dynamic change of user preference is applied by our estimation algorithm. This paper implements a device recommender system on the UMO which was developed by our co-research group [7]. We also show that the recommender system is reasonable for device sharing in a real environment through the "Presentation" scenario.

The remainder of this paper is organized as follows. Section 2 describes related work and Section 3 shows the architecture of middleware for device sharing and our system. Section 4 describes the methods for device reasoning. Section 5 and section 6 show a scenario and conclusion with future work, respectively.

2 . Related Work

There are some previous researches for these issues although an application domain is not the same as ours. In Ubiquitous computing, user wants to get a service composed by most suitable resources. UPnP and JINI are representative technologies that enable devices to interface with one another and its goal is similar to ours. These technologies can make home appliances, PCs and printers "plug and play" in a network environment [4, 5]. Universal Plug and Play (UPnP) [4] was created to provide an easy and convenient interface between a service and electronic devices dispersed in various locations. Also UPnP provides a common interface for any device. Therefore, a home service application can interface with devices without having to know the specifications of every device that supports UPnP. JINI [5] is a JAVA-based middleware developed by Sun Microsystems which allows network devices and software to dynamically interface with one another. JINI was proposed as a standard interface between devices or software in a distributed environment. From the resource sharing respective, the focus of UPnP and JINI is similar to ours. They just focus on the device sharing without consideration about context and user preference. However, our other goal is to recommend the user-suitable device. Especially we focus on the personalized recommendation.

Celadon Project [3, 6] was proposed by IBM and its main objective is to enable seamless collaboration between a wide range of heterogeneous mobile and environmental devices facilitated by middleware. They propose collaborative environments to be organized into Celadon zones, which are public areas equipped with wireless access points for such technologies as Bluetooth or 802.11, and with environmental devices, such as displays, printers and servers. Their goal is similar to ours. However, they apply only environmental context for reasoning because their reasoning is executed in server system for every user in the zone, whereas we focus on peer-to-peer environments. Therefore we can use more personalized information for inference and then serve more customized recommendation. Of course, generally the reasoning engine needs a lot of resource power for applying many factors into reasoning. In our recommender system, however, reasoning factors like ontology instance is generated in real-time when a user requests a service. Therefore our recommender system can be executed in personal mobile object.

3 . Architecture of Device Recommender System

Figure 4 shows the middleware for device sharing and the device recommender system consists of Device Reasoner, Local Context Manager, Context Information Base (CIB), and Wrapper&Encoder modules

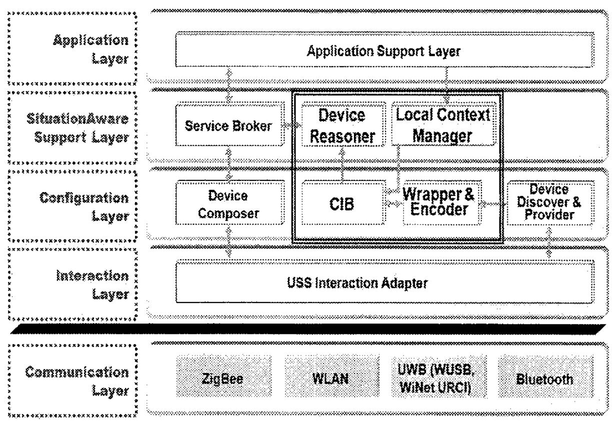

Figure 4. The Architecture of the Middleware for Device Sharing

We provide an overview of our device recommender system in Figure 5. Local Context Manager consists of Local Context Monitor and Local Device Manager. Local Context Monitor monitors and gathers environmental context information of local UMO such as current time, light, and position. Local Device Manager manages local devices owned by the UMO for sharing with other UMOs. Wrapper & Encoder obtains the information of current usable remote devices and sends the information of local devices by communicating with other UMOs. CIB manages device and context information in RDF, which is collected from Wrapper & Encoder and Local Context Manger, and processes SPARQL queries inputted from Device Reasoner. In order to manage context information in RDF, CIB uses a JENA which is one of the ontology-based inference engines [8]. Device Reasoner infers proper devices for composing a user requested service. For this goal, this paper chooses a rule and uses a JESS as rule-based reasoning engine [9].

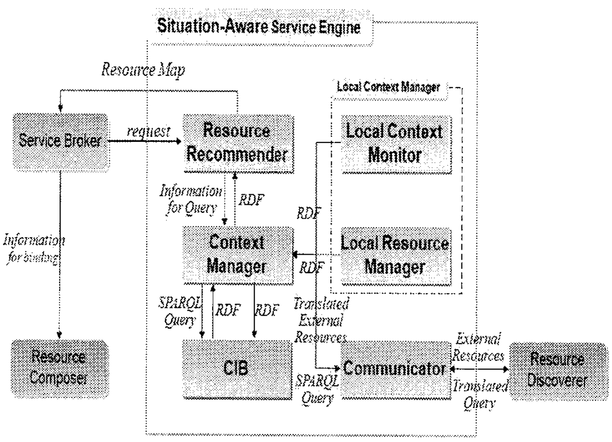

Figure 5. The Architecture of Device Recommender System

Figure 6 show sub-modules of Device Reasoner and a simple action flow. If a user requests "Presentation" service, then Context-Based Reasoner requires information needed for the "presentation" service to CIB through Context Manager. Context Manager manages the flow of request and information in Device Recommender System. The CIB infers devices needed for composing the "Presentation" service by using ontology-based reasoning then returns the information about usable devices, user preference, and environmental context. Context-Based Reasoner and User-Based Reasoner infer the devices based on environmental context and user preference by using rule-based reasoning, respectively.

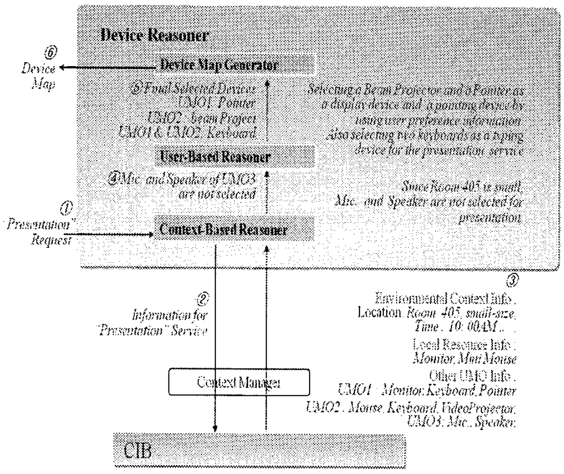

Figure 6. Architecture of Device Recommender System

In our example, Context-Based Reasoner is not selected a Mic. and a speaker because meeting room 405 in which user requests the "Presentation" service is small-size room. Also User-Based Reasoner selects the pointer and keyboard from UMO1 and the videoproject and keyboard from the UMO2 as a pointing, typing, and display device. Device Map Generator generates a map which consists of selected devices and then returns the device map to Service Broker.

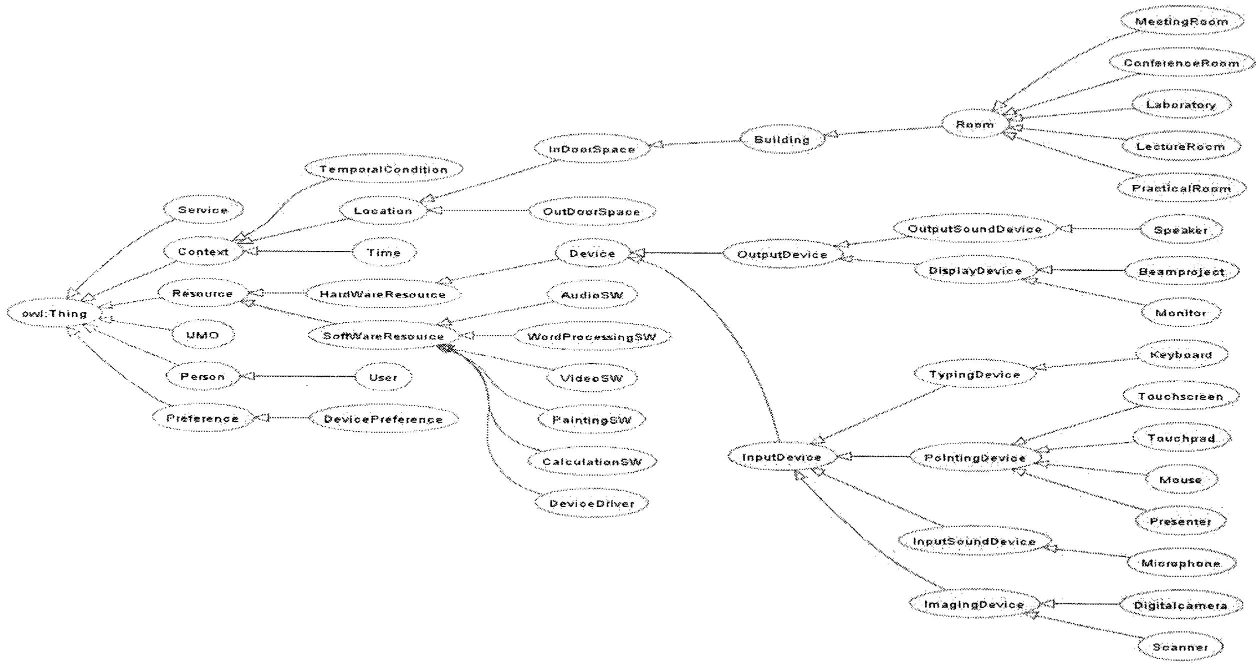

Figure 7. Partial Ontology structure

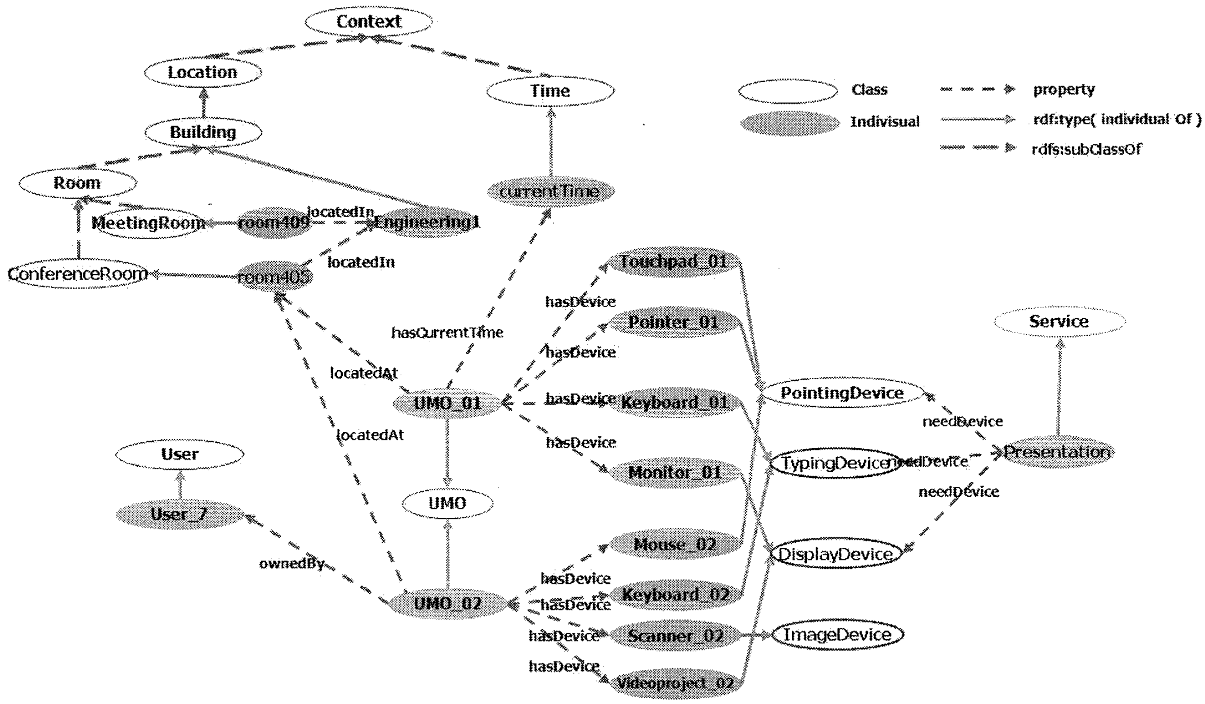

Figure 8. Sample Ontology Instance for ‘Presentation’ Service

4 . Device Reasoning

(A) Relation between services and devices

To define a relation between service and devices described in the introduction, we consider ontology. Ontology is a formal, explicit specification of a shared conceptualization of a domain [10]. Ontology includes the machine-interpretable definition of basic concepts in the domain and relationships among taxonomies. Ontology shares a common understanding of the structure of descriptive information and enables reuse of domain knowledge [11]. By ontology-based reasoning, the device recommender system infers which devices are needed for user requested service. Therefore this paper selects the ontology to generalize the concepts about devices, services, and environmental context and generate a relation among these concepts.

Figure7 and Figure 8 show the partial graph of our ontology and an example ontology instance for “presentation” service, respectively. Our ontology consists of six main classes, and each class has its own sub-classes. The service and device classes express the usable services and device resources in the USS, respectively. The service and device classes are connected by the needsDevice and usedForService properties. The UMO class is connected to the "User," "Location," "Time," and "Device" classes by the "OwnedBY," "LocatedAt," "HasCurrentTime," and "hasDevice" properties, respectively. There are two ubiquitous mobile objects (UMO) in the example ontology instance. UMO_01 has Touchpad_01, Pointer_01, Keyboard_01, and Monitor_01. UMO_02, its user is User_7, has Mouse_02, Keyboard_02, Scanner_02, and Videoproject_02. These UMOs are located in Room 405 which is one of conference rooms in engineering building. For example, , if a user requests 'Presentation' service, then our recommender system can infer the 'Presentation', service is composed of ‘Pointing Device’, ‘Typing Device, and ‘Display Device’. Therefore the system can recommend devices needed for the "Presentation" service among current usable devices. In our example, the system recommends the ‘Touchpad_01’, ‘Pointer_01’, ‘Keyboard_01’ and ‘Monitor_01’ related with ‘UMO_01’ and ‘Mouse_02’, ‘Keyboard_02’, and ‘Videoproject_02’ related with ‘UMO_02’. Also the system can infer that two UMOs are in the ‘room405’ and the room is a conference room in the engineering building 1. Our specific ontology and reasoning techniques are described in [12].

(B) Reflection of context and user preference

The result of the ontology-based reasoning is just a current usable device set for composing a user request service. Therefore, we have to consider about how to reflect a context and a user preference which is mentioned in the introduction as the second issue. To infer devices based on environmental context and user preference, this paper uses a rule. The reasoning based on the environmental context is executed in Context-Based Reasoner. Although Context-Based Reasoner infers devices based on environmental context, these devices are not personalized devices. In order words, the result of context-based reasoning is the same for all users who are in the same environment and requests the same service. However, satisfactions for each user are different because the user preferences are not the same.

Therefore User-Based Reasoner infers personalized devices by using user history and profile such as age, sex, weight, eyesight and so forth. Table 1 is an example rule for applying a user profile into device reasoning. The example rule is used to infer a mouse for left hander which is located in distance '5'. The detailed techniques for our rule-based reasoning are described in [13].

Table 1. An Example JESS rule for a User Profile

```
((defrule preferedMouse
    (User (is_lefty true) ) (Mouse (id ?mouseID)
    (shape left_handed) {distanceLevel <= 5} )
        => (assert (RecommendedMouse ?mouseID)))
```

(C) Application of dynamic changing of user preference

Although the rule-based reasoning reflects a current user preference, the preference can be dynamically changed by a verity of condition. That problem is related with the third issue in the introduction. For this issue, we should consider the user history about the usage of devices for customized service. To extract user preference from user history about device selection, this paper proposes a method for estimating a preference score of each device by Equation 1.

$$s(i) = \begin{Bmatrix} 1, for\ property(i) = property(pv) \\ 0, for\ property(i) \neq property(pv) \end{Bmatrix}$$

$$w(i) = \frac{N - R_i + 1}{\sum_{i=1}^{N}(N - R_i + 1)}$$

$$ps(device) = \sum_{i=1}^{N} s(i) \cdot w(i) \qquad (1)$$

Table 2. Estimated Preference Score of Devices

Monitor	Monitor_01	Monitor_02	Monitor_03
Brand	1 * 0.5	0 * 0.5	0 * 0.5
Brightness	1 * 0.333	1 * 0.333	0 * 0.333
Size	0 * 0.166	1 * 0.166	1 * 0.166
Total Score	0.833	0.499	0.166

In the Equation 1, *s(i)* denotes whether the value of the current device property *i* is the same as the value of user preference properties (*pv*). If the user prefer to that property, the *s(i)* is 1. *w(i)* calculates the weight of current property *i*. *N* and R_i denote the total number of device property and the ranking of property *i*. The ranking of each property is predefined by the user or calculated from property values of previous selected devices. Table 2 shows an example when three monitors are selected by Context-Based Reasoner and the ranking of user preference property is that the Brand is 1st, Brightness is 2nd, and the Size property is 3rd. the Brand and Brightness property values of the Monitor_01, The Brightness and Size property values of the Monitor_02, and The Size property value of Monitor_03 are the same as user preference property values. Finally, the first recommended monitor is the Monitor_01.

5 . Scenario

For our test scenario, we consider a "Presentation" service. Figure 9 shows an experimental testing environment for the "Presentation" service. There are some sharable devices in the meeting room.

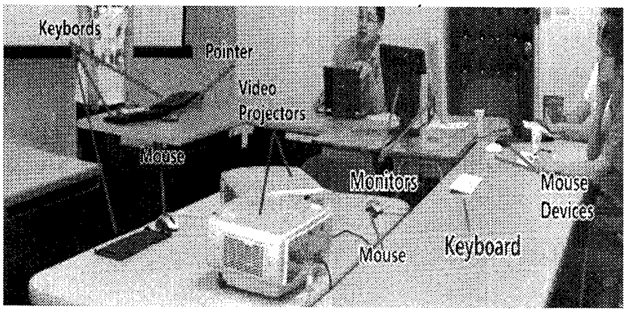

Figure 9. Experimental environment for the "presentation" service

In figure 10, the pointer requests "presentation" service to her UMO. And then the UMO automatically searches and recommends devices for composing the "presentation" service. In our test scenario, videoproject, monitor, mouse, keyboard, pointer devices are searched. For display device, the UMO recommends two videoproject among searched four display devices, which consist of two kinds of videoproject and two kinds of monitors, because the monitors are so small for "Presentation" service. Also the videoproject with high resolution has higher ranking than the other one because of user preference.

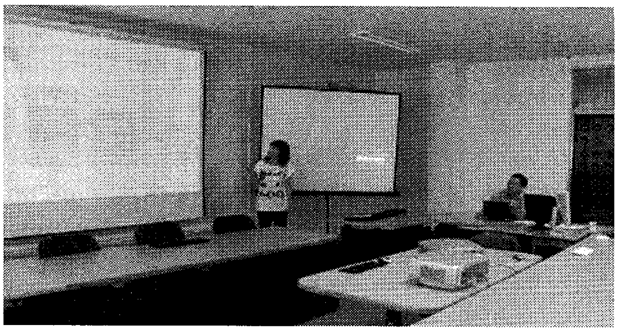

Figure 10. Service request and result

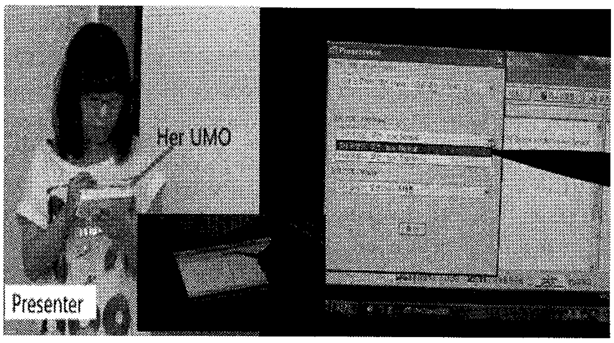

Figure 11. The presentation using devices recommended by the UMO

In the case of typing device, two keyboards are recommended among searched four typing devices by the environmental context because a distance between the user and selected two keyboards is closer than the others. Also the normal type keyboard takes a higher ranking than the natural keyboard by user history. For pointing device, the UMO recommends the unique pointer in the meeting room ruling out mouse devices because of user history.

Acknowledgement This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government[NRF-2009-352-D00252]

6 . Conclusion

In order to offer customized services to users by device sharing in ubiquitous environments, this paper has proposed the device recommender system which uses ontology and rule for applying environmental context and user preference. In addition, we proposed and evaluated the method for estimating user preference which is dynamically changed. We also has implemented its prototype system on the UMO and tested in real environment. Our recommender system can solve not only the limitation problem of mobile devices but also offer the user-oriented service through context-awareness.

Current our system supports just ten services. In the future, however, we will develop an emulator for performance evaluation and evaluate user satisfaction about the result of devices recommendation by our system.

Reference

[1] K. Romer, T. Schoch, F. Mattern & T. Dubendorfer, "Smart Identification Frameworks for Ubiquitous Computing Application," Proceedings of PERCOM 2003, Fort Worth, USA, March, 2003.

[2] M. Weiser, "Some computer science issues in ubiquitous computing," Communications of the ACM, Vol. 36, Issue 7, pp.75-84, July, 2003.

[3] M. C. Lee, H. K. Jang, S. Y. Kim, Y. S. Paik, S. E. Jin, S. Lee, C. Narayanaswami, M.T.Raghunath & M.C.Rosu, "Celadon: Infrastructure for Device Symbiosis", Proceedings of The Seventh International Conference on Ubiquitous Computing 2005, Tokyo, Japan , September, 2005.

[4] G. Bhatti, Z. Sahinoglu, K. A. Peker, J. Guo, & F. Matsubara, "A TV-Centric Home Network to Provide a Unified Access to UPnP and PLC Domains," Proceedings of International Workshop on Networked Applications, pp.234-242, January, 2002.

[5] JINI, http://www.jini.org.

[6] M. C. Lee, H. K. Jang, Y.S.Paik, S.E. Jin & S. Lee, "Ubiquitous device collaboration infrastructure: Celadon," Proceedings of SEUS-WCCIA.2006, Korea, April 2006.

[7] Y. M. Jeong, J. H. Kim, J. J. Ko, S. W. Lee, M. H. Kim, & J. C. Choi, ?Resource Collaboration Framework based on Context Awareness in Ubiquitous Computing Environment," Proceedings of ICUCT 2008, Khabarovsk, Russia, August 2008.

[8] JENA2: A Semantic Web Framework, http://jena.sourceforge.net/.

[9] JESS, the Rule Engine for the JavaTM Platform, http://herzberg.ca.sandia.gov/jess/.

[10] Gruber, T. "A Translation Approach to Portable Ontology Specification," Knowledge Acquisition, Vol. 5, No. 2, p.199-200 1993.

[11] N.F, Noy and D.L McGuiness, "Ontology Development 101: A Guide to Creating Your First Ontology," Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SML-2001-0880, March 2001.

[12] W. I. Park, J. H. Park, S. J. Kang, Y. D. Lee, H. S. Choi, S. H. Kang, M. S. Choi, K. R. Sohn, Y. K. Kim, & J. H. Kang, "A Context-Based Collaboration System in Ubiquitous Environments", Proc. ICCIT 07, Gyeongju,Korea, November 2007.

[13] J.-H. Park, W.-I. Park, Y.-K. Kim, & J.-H. Kang, "Situation-Based Resource Recommender system in Ubiquitous Environments", Proceedings of ICPCA 2008, Vol 1, pp.74~77, Alexandria, Egypt, October, 2008.