A Recommender System for Device Sharing Based on Context-Aware and Personalization

Jong-Hyun Park
Computing and Communications Center, Kyushu University,
Hakozaki 6-10-1, Higashi-ku, Fukuoka, 812-8581, JAPAN
[e-mail: jonghyunpark@cnu.ac.kr]

Received January 13, 2010; revised February 26, 2010; accepted March 19, 2010; published April 29, 2010

Abstract

In ubiquitous computing, invisible devices and software are connected to one another to provide convenient services to users [1][2]. Users hope to obtain a personalized service which is composed of customized 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. Although users request the same service in the same USS, the most suitable devices for composing the service are different for each user. For these user requirements, this paper proposes a device recommender system which infers and recommends customized 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 context and user preference. Also I implement a prototype system and test performance on the real ubiquitous mobile object (UMO).

Keywords: Device sharing, recommender system, context reasoning, context-aware, ontology-based reasoning, rule-based reasoning, estimating user preference

A preliminary version of this paper appeared in IEEE ICC 2009, June 14-18, Dresden, Germany. This version includes a concrete analysis and supporting implementation results on MICAz sensor nodes. This research was supported by a research grant from the IT R&D program of MKE/IITA, the Korean government [2005-Y-001-04, Development of Next Generation Security Technology]. We express our thanks to Dr. Richard Berke who checked our manuscript.

DOI: 10.3837/tiis.2010.04.006

1. Introduction

In a ubiquitous computing environment, service composition and collaboration among heterogeneous devices are required thus an infrastructure that supports these require-ments is an essential factor in a seamless service delivery [3]. In other words, users in ubiquitous environments expect to obtain a variety of services using only a personalized mobile object such as a mobile phone or a personal digital assistant (PDA). However, these mobile objects are light-weight and equipped with limited devices such as tiny display screens, limited input, and less powerful processors.

Therefore, users want to get a service -specifically a customized service- by sharing devices in the USS where the users are located.

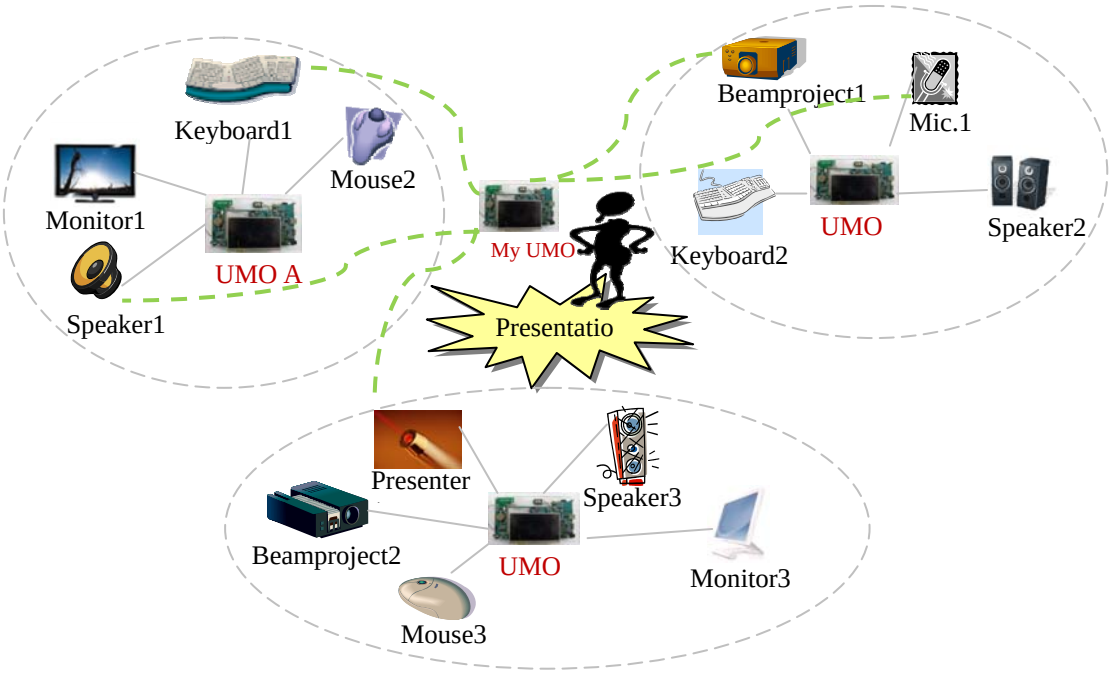

Fig. 1. The "Presentation" Service Composed by Device Sharing

For example, a presenter in a meeting room can use his/her mobile object to auto-matically search for devices that can prepare "presentations" and provide a customized service through context-aware. For requirements such as these, intelligent mobile objects have to act as the mediator between users and services. In **Fig. 1**, if the user requests for the "presentation" service in his/her Ubiquitous Mobile Object (UMO), then the UMO searches for sharable devices by interacting with other UMOs and recommends proper devices among the identified sharable devices. After the user selects the devices, the UMO then offers a customized service by connecting the devices. This paper focuses on the recommendation of a device for composing a service. Thus, the means of inferring a proper device among numerous sharable devices is this study's main topic. A device-sharing framework for service composition and collaboration among heterogeneous devices and a device recommender system, which is a core component in the framework, are proposed as well. This paper also puts forward methods for solving the following issues for the device recommender system.

- How can devices and services be generalized and connected?

There are a variety of devices and services for a user in a USS. Moreover, each device has various properties. Therefore, defining the concept of devices and services and the relationship between them is necessary for device recommendation. Although the system can find devices for composing a service, a device set for the service is not always the same in different context. Therefore this paper considers the environmental context and the user preference in order to recommend devices.

- How can the environmental context be applied to device recommendation?

To apply the environmental context, the device recommender system should generate from a low-level context to a high-level one. This task is considered as the most important for the environmental context-aware. I should also consider how the high-level context can be applied to device reasoning.

- How can user preference be applied?

Although users request the same service in the same environmental context, the most suitable devices for composing the service are not the same for every user because each user's preferences are different. Therefore, I should consider user preference and device usage history when recommending personalized devices. This paper designs and implements a device recommender system on the UMO, which was developed for device sharing by our co-research group [4]. Our system uses two kinds of reasoning techniques for the above issues. One is ontology-based reasoning and the other is rule-based reasoning. In addition, I propose a method for estimating user preference score from usage history. The performance of the two techniques is evaluated and determined in order to provide an acceptable performance for device recommendation. The remainder of this paper is organized as follows. Section 2 describes the Related Work. Section 3 shows a middleware for device recommendation and the architecture of our system. Section 4 and section 5 describe the ontology-based reasoning and the rule-based reasoning including the method for estimating user preference respectively. Section 6 shows a scenario and performance evaluation. Finally, Section 7 provides concluding remarks.

2. Related Work

There are some previous researches for these issues mentioned in introduction, 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 [5][6] . Universal Plug and Play (UPnP) [5] 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 [6] 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. However, our other goal is to recommend the user-suitable resource. Especially I focus on the personalized

recommendation. Celadon Project [4][7] was proposed by IBM since 2004 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 I do not consider a server system for device recommendation and focus on device reasoning in a personal mobile object in Peer-to-Peer environments. Therefore I can use more personalized information for inference and then serve more customized recommendation.

3. Device Recommender System

Fig. 2 shows a middleware for device sharing, which was proposed in our previous study. The device recommender system consists of the Device Reasoner, Local Context Manager, Context Information Base (CIB), and Wrapper & Encoder, all enclosed in a bold lined-box.

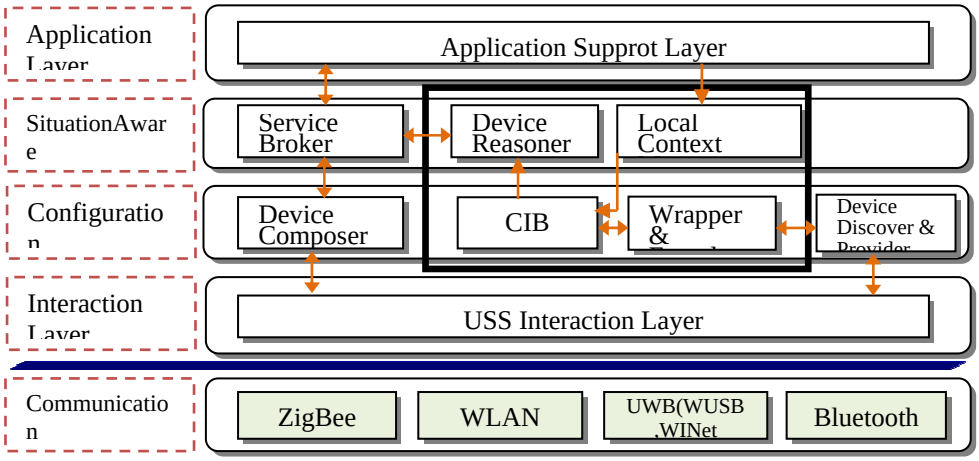

Fig. 2. The Middleware Architecture for Device Sharing

I provide an overview of our device recommender system in **Fig. 3**. The Communicator obtains information on current usable remote devices and then sends the information on local devices to other UMOs. The CIB manages the device and the context information in Resource Description Framework (RDF), which is collected from the Communicator and the Local Context Manager. In addition, the CIB processes SPARQL queries inputted from the Device Reasoner and returns the result in RDF. In our system, I select RDF to represent information [8] because I use ontology for reasoning, and RDF is one of the standard languages for describing ontology. For interoperability, therefore, the system's natural choice for searching is SPARQL because it is a standard query language for RDF data [9]. To manage context information in RDF, the CIB uses JENA, one of the most popular ontology-based inference engines [10]. The Device Reasoner infers the proper devices for composing a user-requested service. To achieve this goal, this paper chooses a rule and uses JESS as a rule-based reasoning engine [11]. The Local Context Manager in the middleware consists of the Local Context Monitor and the Local Device Manager.

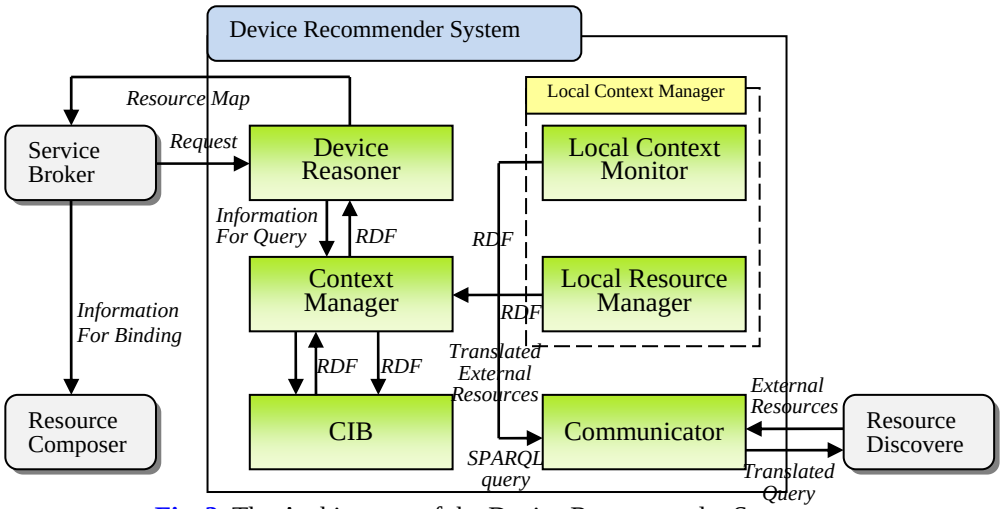

Fig. 3. The Architecture of the Device Recommender System

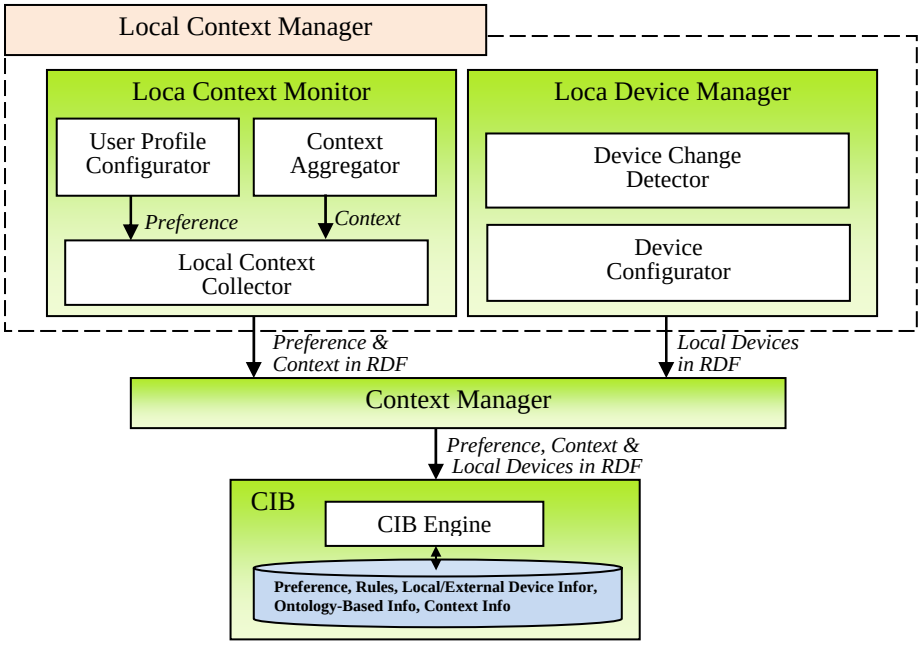

Fig. 4. The Architecture of the Local Context Manager

Fig. 4 describes the architecture of the Local Context Manager in detail. The Local Context Monitor monitors and manages user profiles and context information. The User Profile Configurator gathers and manages user profiles and preference information. The context aggregator dynamically collects environmental context information on the local UMO such as current time, light, and position. The Local Device Manager manages the local devices owned by the UMO for sharing with other UMOs. The Device Configurator aggregates the local device information and the Device Change Detector monitors, and manages the situation of local devices. If one of the local devices is not sharable anymore, then the Device Change

Detector requests reaggregation into the Device Configurator. The output of the Local Context Manager is written in RDF.

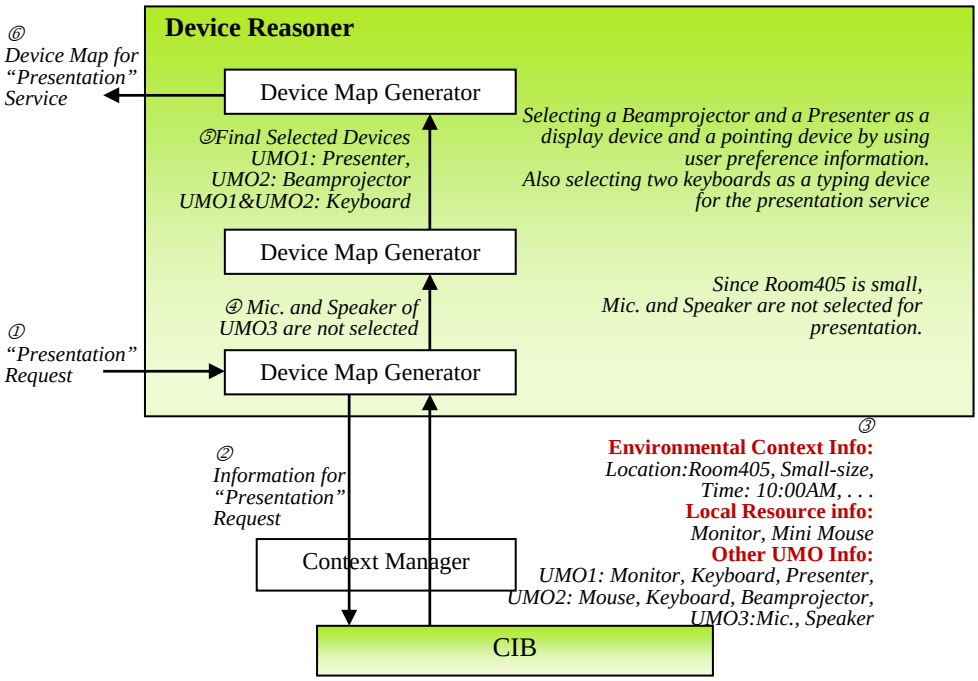

Fig. 5. The Architecture of the Device Reasoner

Fig. 5 shows the sub-modules of the Device Reasoner and the action flow between them. If a user requests a "presentation" service, the Context-Based Reasoner will demand for the information needed to compose the "presentation" service from the CIB through the Context Manager. The CIB infers and returns a current sharable device, a user preference, and environmental context information for composing the "presentation" service by referencing the ontology. The Context-Based Reasoner infers devices based on the environmental context. In our example, a microphone and speaker of the UMO3, which were selected by ontology-based reasoning, are filtered out for the "presentation" service because the meeting room (room 405) requested by the user is small. To select customized devices among the candidate devices that was filtered by the Context-Based Reasoner, the User-Based Reasoner employs a rule for reflecting a user preference and an estimating method for obtaining a user preference dynamically updated from user history. In **Fig. 5**, the User-Based Reasoner selects the "beam projector" of UMO2 as a display device, the "presenter" of the UMO1 as a pointing device, and two "keyboards" of the UMO1 and the UMO2 as a typing device. The Device Map Generator generates a map consisting of selected devices and then recommends the map to the Service Broker.

4. Ontology-Based Reasoning

To address the first issue among the three issues mentioned in the introduction, this paper considers ontology. According to Thomas [12], the ontology-based approach is very powerful and applicable in the ubiquitous environment. Ontology is a formal and an explicit

specification of a shared conceptualization of a domain [13]. 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 [14]. Ontology is used to make explicit assumptions and to separate domain knowledge from operational knowledge. Additionally, ontology has the advantage of sharing of knowledge, logic inference and the reuse of knowledge. If a system uses ontology, then the system can provide a general expressive concept and offer syntactic and semantic interoperability. By mapping concepts in different ontologies, structured information can be shared. Hence, ontology is a good candidate for expressing context and domain knowledge. Many ontology languages exist, including Resource Description Framework Schema (RDFS) [15], DAML+OIL [16], and OWL [17].

OWL was proposed by the Web Ontology Working Group of W3C and is regarded as a key to the Semantic Web. As a language for defining the Web, OWL is more expressive than other ontology languages such as RDFS. OWL is based on Resource Description Framework (RDF) [8], a language that embodies the idea of identifying objects using Web identifiers and representing resources in terms of simple properties and property values formed by triple. This approach could produce a new metadata-generating procedure using semantic relationships [14]. As mentioned in previous section 3, our system uses RDF to express information for device sharing. **Fig. 6** shows a class-property diagram of our ontology. Our ontology consists of six main classes, with each class having its own sub-classes. The service class and device class express usable services and device resources in the USS, respectively. The service class and the device class are connected by the needsDevice property and usedForService property. The UMO class is connected to the "User," "Location," "Time" and "Device" classes by the "OwnedBY," "LocatedAt," "HasCurrentTime," and "hasDevice" properties, respectively.

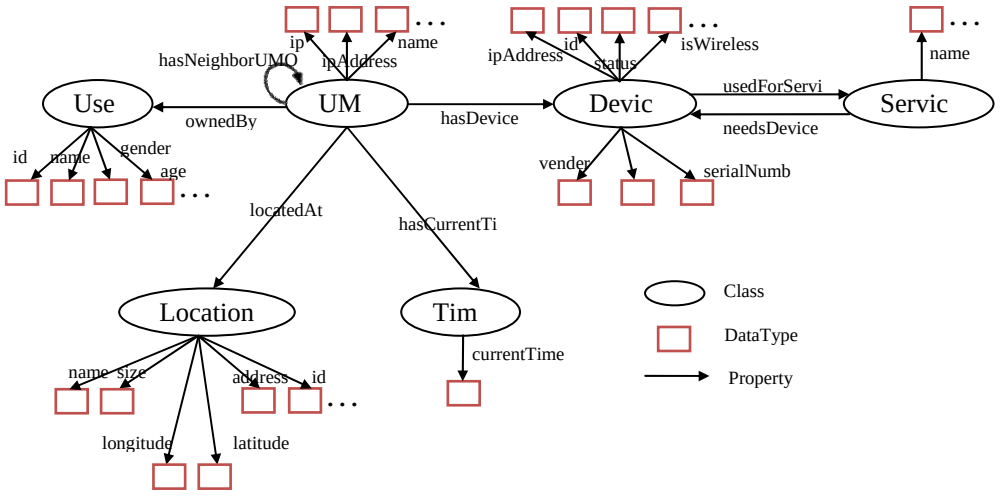

Fig. 6. Ontology Modeling

Fig. 7 is an example of the ontology-based reasoning for the "presentation" service. If a user requests his/her UMO for the "presentation" service, the UMO then infers that the user needs the "Pointing Device," "Typing Device" and "Display Device" resources for composing the "presentation" service through the ontology. The next step in the ontology-based reasoning is to find current sharable device resources. In this example, there are two UMOs. UMO_01 has a presenter, a keyboard, and a monitor, and UMO_02 has a mouse, another keyboard, and a

beam projector. The UMO also infers that these UMOs are located in room 405, which is a meeting room in the Engineering building.

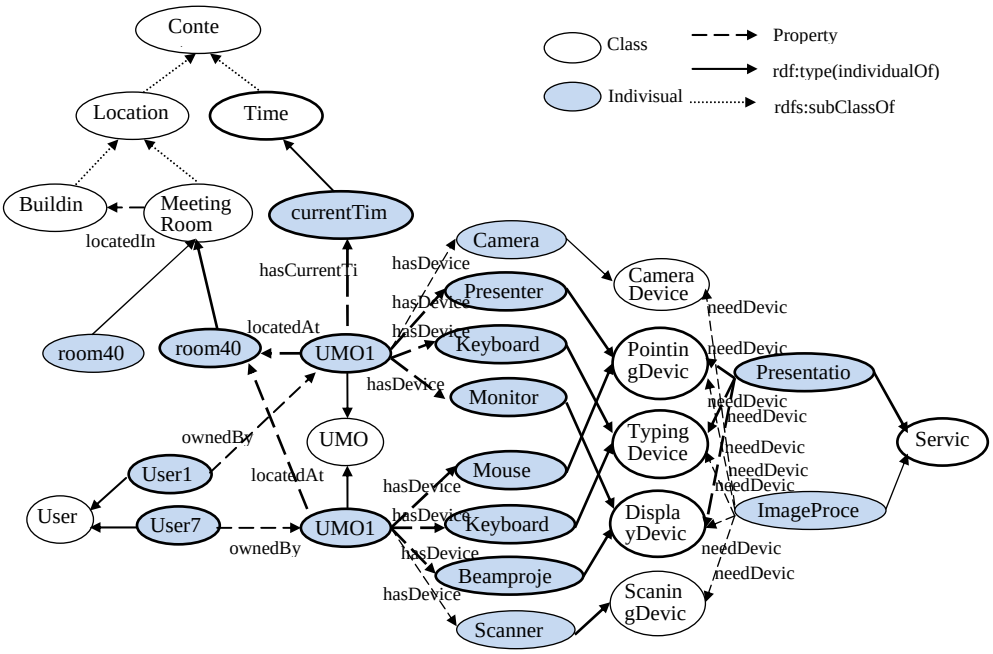

Fig. 7. Sample Ontology Instance

Table 1. Example SPARQL query and its result

SPARQL query

SELECT ?UMO ?Device ?Room_size ?Current_time
WHERE {
?Service :servicename "presentation".
?Service :needsdevice ?DeviceType.
?Device rdf:type ?DeviceType.
?UMO :hasdevice ?Device.
?UMO :locatedat ?Room.
?Room :size ?RoomSize.
?Time :currenttime ?currentTime
}

Result
UMO : UMO_01 UMO_02 Device : monitor, presenter, keyboard_01, : beam-project, mouse, keyboard_02 Room_size : small Current_time : 10:00AM

To query the RDF data, I use the SPARQL query language proposed by W3C because it is one of the standard query languages for RDF. Table 1 shows an example SPARQL query for obtaining the environmental context and device information needed for the "presentation" service and its result.

5. Rule-Based Reasoning and Estimating User Preference

The devices inferred by the ontology-based reasoning are just a set of usable devices in current environmental context for composing a userrequested service, not a personalized set.

In this paper, I use rules for applying an environmental context and a user preference to the device reasoning. The rule-based reasoning consists of two parts for device reasoning. The former is a reasoning based on the environmental context, and the latter is based on the user preference. **Fig. 8** shows various conditions for selecting a monitor device in different environmental contexts.

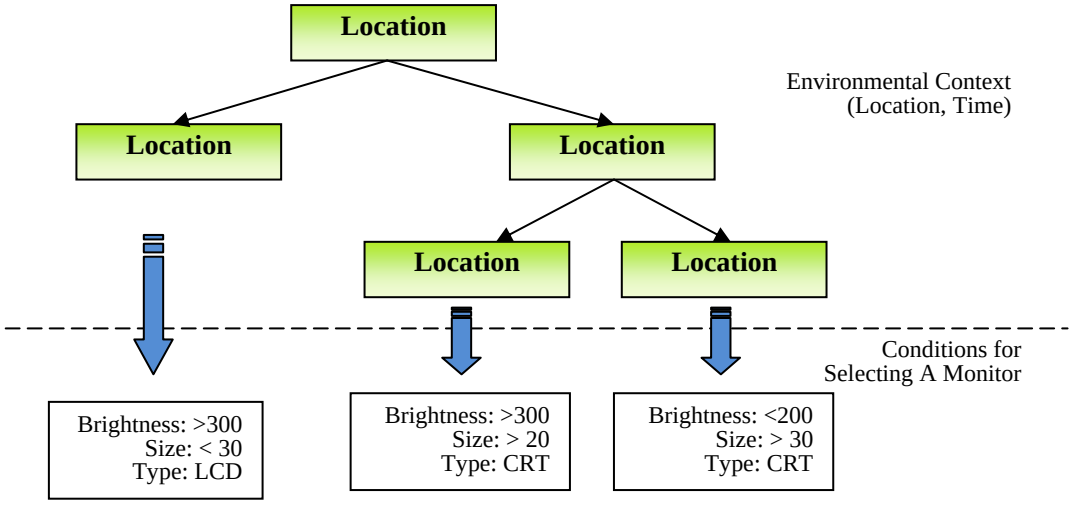

Fig. 8 Various conditions for selecting a monitor based on the environmental context.

Table 2 describes a rule in the JESS rule for selecting a monitor when the user is indoors. Although the Context-Based Reasoner infers devices based on the environmental context, these devices are not personalized devices. In order words, the result of context-based reasoning is the same for every user who requests the same service in the same environment. However, the satisfaction of each user is different because each user's preferences are not the same. Therefore, the User-Based Reasoner infers personalized devices using user preference. Our system uses a user profile and history as user preference information. User profile information such as age, gender, weight, height, and eyesight is defined as a rule similar to Table 3 and is used for personalized reasoning.

Table 2. Example JESS rule for selecting a monitor

JESS Rule
(defrule IndoorMonitorForPresentation (Service presentation) (Location Indoor) (Monitor (Monitor_ID ?id) (Monitor_Brightness ?brightness) (Monitor_Size ?size) (Monitor_Type ?type)) (test (> ?brightness 300) (> ?size 30) (> ?type LCD)) => (assert (RecommendedMonitor ?id)))

Table 3 shows an example rule for selecting a mouse for a lefty through the user profile.

Table 3. Example JESS rule for a user profile

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

Although the rule-based reasoning reflects a current user preference, the preference can be dynamically changed by a verity of condition. Therefore the User-Based Reasoner must consider the user history about the usage of devices. Of course, there are a variety of factors except the usage history. However, our recommender system runs in a personal mobile device which has limited computing power. Therefore, this paper can't consider a variety of factors and have to choose core factors for the reasoning. User history is one candidate factor and seems proper for our recommendation because our target environment is a personal mobile object. Using a usage history causes problems of private information frequently. Therefore, this paper focuses on the user profile and usage history for user-based reasoning. To infer a current user preference from the user's personal history, this paper proposes a method for estimating the preference score for each device from the history.

Table 4 shows an example user history which includes the start time, the end time and the attribute value of a device. The start time means the time when a user actually selects a device and the stop time is the disconnected time of the selected device. The attributes are matched with the properties of classes defined in the ontology like the vendor of a device.

Table 4. Usage history instance

2010/1/1 AM 09:10:20 2010/1/1 AM 10:10:50 Monitor vendor:A,brightness:300,size:20
2010/1/1 AM 09:10:20 2010/1/1 AM 10:10:50 Keyboard vendor:K,is_wireless:true,key:103
2010/1/1 AM 09:10:20 2010/1/1 AM 10:10:50 Mouse vendor:M,is_wireless:true,button:3
:
2010/1/3 PM 06:30:40 2010/1/3 PM 06:32:35 Keyboard vendor:E,is_wireless:false,key:108
2010/1/3 PM 06:30:45 2010/1/3 PM 07:25:20 Monitor vendor: A,brightness:300,size:20
2010/1/3 PM 06:30:50 2010/1/3 PM 07:26:40 Mouse vendor: M, is_wireless:false, button:7
2010/1/4 PM 11:20:20 2010/1/4 PM 11:55:20 Monitor vendor: C,brightness:250,size:19
:
2010/1/7 PM 07:55:30 2010/1/7 PM 09:10:20 Monitor vendor:B,brightness:300,size:19
2010/1/7 PM 07:55:30 2010/1/7 PM 07:57:30 Mouse vendor:O,is_wireless:false,button:5
2010/1/7 PM 07:58:00 2010/1/7 PM 09:10:20 Mouse vendor:U,is_wireless:true,button:3
2010/1/7 PM 07:55:30 2010/1/7 PM 09:10:20 Keyboard vendor:E,is_wireless:true,key:108
:

Table 5 shows an example transaction data including the usage history of monitor devices in the **Table 4**. To remove the meaningless transaction, this paper defines that the threshold value about the usage time of the device is 60000ms. This transaction data is used to estimate user preference score.

The *s(i)* in Equation (1) denotes whether or not the current property *i* of the device is user preference property(*pp*) . The preference property*(pp)* is automatically decided by the frequency of user selection, of course, can be manually predefined or changed by a user. In **Table 5**, I can infer that the user prefers the vender “A” monitor with "300" brightness and "19" inch.

Table 5. Transaction Data

Type	Property	Value	Frequency of device selection
Monitor	Vender	A	7 (1st)
		B	1
		C	2
	Brightness	300	6 (2nd)
		250	4
	size	19	3 (3rd)
		20	2
		21	2
		17	2

$$s(i) = \begin{cases} 1, & for\ property(i) = property(pp) \\ 0, & for\ property(i) = property(pp) \end{cases} \quad (1)$$

$$w(i) = \frac{N - R_i + 1}{\sum_{i=1}^{N} (N - R_i + 1)} \quad (2)$$

$$ps(device) = \sum_{i=1}^{N} s(i) \cdot w(i) \quad (3)$$

The *w(i)* in Equation (2) is the weight of property *i* for selecting a device. *N* and R_i denote the total number of device property and the ranking of property *i*, respectively. The R_i is the ranking of frequency of property *i* in the height frequency of all properties. For example, the ranking R for "vender" property in **Table 5** is first, "brightness" property is second, and "size" is third because vender "A," brightness "300" and size "19" have been selected 7, 6 and 3 times, respectively. The *preference score(ps)* of a device can be estimated by Equation (3).

Table 6. Candidate monitors

Monitor	Property	Value
Monitor_01	Vender	A
	Brightness	300
	Size	20
Monitor_02	Vender	B
	Brightness	300
	Size	19
Monitor_03	Vender	C
	Brightness	250
	Size	19

Table 7. Estimated preference score

Monitor	Monitor_01	Monitor_02	Monitor_03
Vender	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
Preference Score	*0.833*	*0.499*	*0.166*

If the three candidate monitors in **Table 6** are inferred by the previous reasoning, then the preference score of each monitor can be estimated as shown in **Table 7** by Equation (3). Finally, monitor_01 with the highest preference score will be recommended to the user.

6. Implementation and Evaluation

6.1 Scenario

For our test scenario, I consider a "presentation" service. **Fig. 9** shows an experimental testing environment for the "presentation" service. There are some sharable devices in the meeting room.

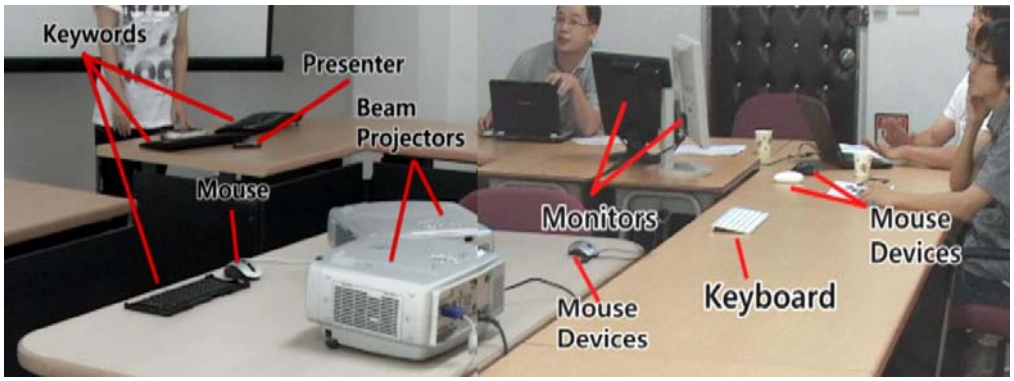

Fig. 9. Experimental environment for the "presentation" service

In **Fig. 10**, the presenter makes a request for a "presentation" service to his/her UMO. The UMO then automatically searches and recommends devices for composing the "presentation" service.

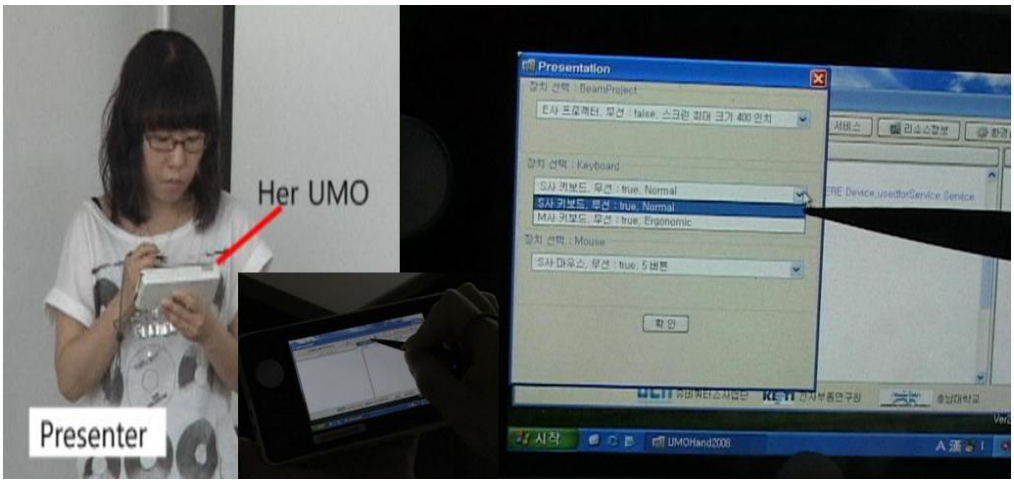

Fig. 10. Service request and results

In the test scenario, there are four display devices consisting of two kinds of beam projectors and two kinds of monitors. The UMO recommends two beam projectors for the display device

because the monitors are so small for the "presentation" service. Furthermore, the beam projector with a high resolution has a higher ranking because of user preference. In the case of the typing device, two keyboards are recommended among the four typing devices searched by the environmental context because the distance between the user and the two keyboards is closer than the others. The normal-type keyboard also takes a higher ranking than the natural keyboard based on user history. For the pointing device, the UMO recommends the unique presenter in the meeting room, ruling out mouse devices because of user history.

Fig. 11. The presentation using devices recommended by the UMO

If the user selects the devices among the recommended devices, then the UMO will connect the selected devices. The user can then start the presentation.

6.2 Performance Evaluation

The Korea Electronics Technology Institute (KETI), which is our co-working group, developed the hand-type UMO as the object for device sharing [3]. Our recommender system was developed and evaluated on the UMO. The experimental setup is as follows: the CPU is 499MHz, having 512MB of Memory. I used JENA [10] and JESS [11] as an inference engine for ontology-based and rule-based reasoning. The operating system is Windows XP.

Table 8. Services and Related Devices

Service	Devices for composing a service
Presentation	Beam-project, Monitor, Mouse, Keyboard, Speaker, Mic., Presenter, TouchScreen, Trackball
Wordprocess	Beam-project, Monitor, Mouse, Keyboard, TouchScreen, Trackball, Printer
Displaying	Beam-project, Monitor, Mouse, TouchScreen, Speaker
Scanning	Scanner, Monitor, Mouse
Printing	Printer

The goal of our evaluation is to show the availability of our recommender system on the real mobile object and the approximative threshold of the number of device. for this goal, I tested two kinds of reasoning techniques: the ontology-based reasoning and the rule-based reasoning.

I don't consider the estimating time of user preference because that is statically executed. For the ontology-based reasoning, I select five services in **Table 8**. Each service is related to the different numbers of devices, and these relations depend on the number of relations between device class and service class in our ontology. Thus, **Table 8** is used for evaluating how the number of relations affects the evaluation time for the ontology-based reasoning.

In real environment, it is difficult to prepare a lot of devices for evaluating. Moreover, to set various environmental context is impossible. Therefore I developed the simulator for generating the virtual experimental data. **Fig. 12** shows the ontology-based reasoning times for composing each service in **Table 8**. To measure a performance, I checked the start time before making the SPARQL queries and the finish time after obtaining the results sets. The minimum number of devices for evaluation is 10, while the maximum number is 200. Each device has 10 required properties and 8 optional properties. Likewise, the environmental context information for all services is set in the same manner. The reasoning time for each service is less than 1,000 ms, except for the "presentation" service when the number of devices was less than 50. The time also increases linearly as the size of the device increases.

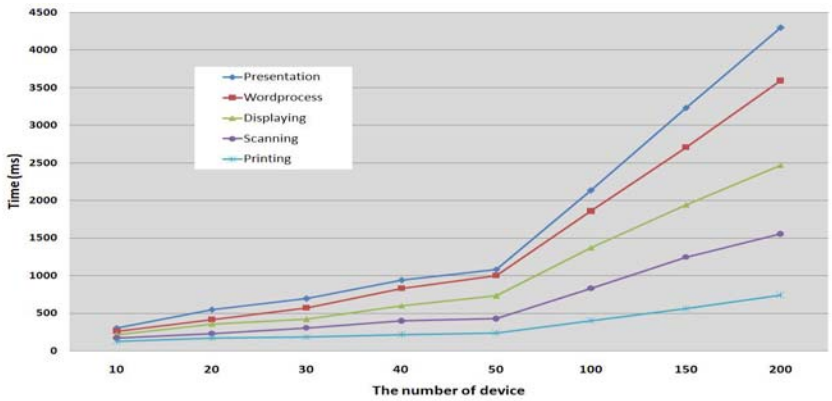

Fig. 12. Ontology-based reasoning time according to the number of devices

To evaluate our rule-based reasoning, I consider the number of facts and rules because these are the input of the rule-based reasoning. The number of facts depends on the number of devices selected by the previous ontology-based reasoning because a fact is composed by a property of devices. The rule is used for filtering devices by current environmental context and user preference. Therefore, these two factors affect the performance of rule-based reasoning. This paper chooses the "presentation" service for evaluating rule-based reasoning. The increasing rate of the fact is relative to the output of the ontology-based reasoning. Therefore, the number of facts is increased from 10 to 200. The number of rules for evaluating is initially 50 rules, which is increased until 300 by multiples of 50. The rule is not dependent on specific facts, and the filtering rate of devices by rules is regular. Of course, the environmental setting for the testing is fixed for all cases.

However, the result of this evaluation can be referred to for various environments for the device sharing. **Fig. 13** illustrates that the rule-based reasoning also linearly increases with the increase in size of the facts and rules.

Through the two kinds of results, I know that the device recommender system, especially in the UMO developed by our co-working group, is reasonable when the number of devices is about 50.

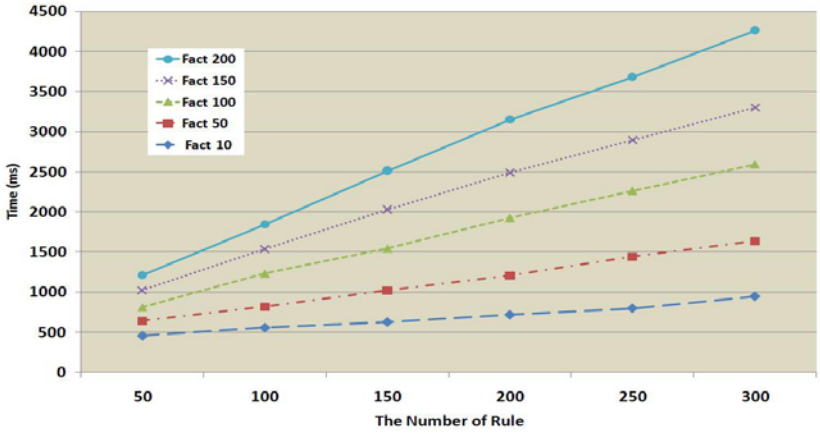

Fig. 13. Rule-based reasoning time according to the number of facts and rules

6.3 Satisfaction Evaluation

To evaluate the satisfaction of our recommender system in real environment, I give five different kinds of monitor, keyboard, and mouse to five users. And then they use the devices during ten weeks. Users execute our recommender system once a week in similar environmental context and select three devices among recommended devices. At this time, users are recommended by two kinds of recommendation method. One is applied with the environmental context and initial user preference information and the other is applied usage history in addition. I investigated the satisfaction of users with four rating scale like very satisfaction, satisfaction, dissatisfaction, and very dissatisfaction. For the convenience, I give the score from 1 to 4 for "very dissatisfaction", "dissatisfaction ", "satisfaction " and "very satisfaction ", respectability. **Table 9** and **Fig. 14** show the satisfaction score of users. The ER in **Table 9** means the recommendation technique applied with usage history and the BR is not applied.

Table 9. The Satisfaction Score of Users

BR: Basic Recommender ER: Extended Recommender			1 week		2 week		3 week		4 week		5 week		6 week		7 week		8 week		9 week		10week	
			BR	ER	BR	ER	BR	ER	BR	ER	BR	ER	BR	ER	BR	ER	BR	ER	BR	ER	BR	ER
Very Satisfaction(4), Satisfaction (3) Dissatisfaction (2), Very Dissatisfaction (1)	User_1	Monitor	3	3	3	3	3	3	2	2	2	2	1	3	1	4	1	4	1	4	1	4
		Keyboar	3	3	3	3	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3
		Mouse	1	1	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4
	User_2	Monitor	4	4	4	4	4	4	2	2	2	2	2	3	2	4	2	4	2	4	2	4
		Keyboar	4	4	4	4	4	4	4	4	4	4	3	3	3	3	4	4	4	4	4	4
		Mouse	3	3	3	3	3	3	3	3	2	2	4	4	4	4	4	4	4	4	4	4
	User_3	Monitor	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4
		Keyboar	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
		Mouse	3	3	3	2	2	2	2	2	2	2	2	2	2	1	3	3	3	3	3	3
	User_4	Monitor	3	3	3	3	3	3	2	2	2	2	2	2	2	3	2	4	2	4	2	4
		Keyboar	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4	3	4	3	4
		Mouse	4	4	4	4	4	4	2	2	2	2	2	3	4	3	4	3	4	4	4	4
	User_5	Monitor	3	3	3	3	2	2	2	2	2	3	2	4	2	2	2	2	2	4	3	4
		Keyboar	3	3	1	1	1	2	1	4	1	1	1	1	1	3	1	4	1	4	1	4
		Mouse	2	2	2	3	2	2	2	3	2	4	3	2	3	2	3	2	3	4	3	4
Total			45	45	43	46	40	46	36	44	35	41	36	44	38	46	40	52	40	57	41	57

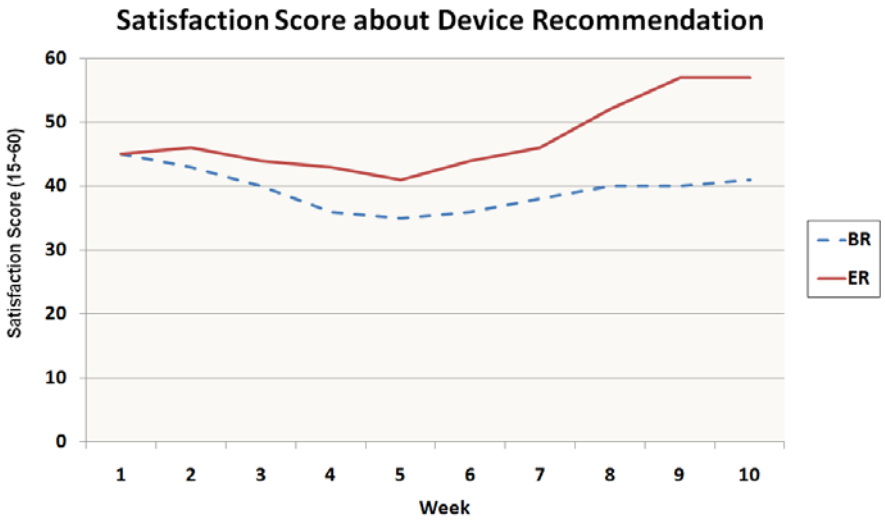

Fig. 14. The comparison of the Satisfaction score according to the time

Since the base of the ER is the BR, the initial satisfaction score of two recommenders is the same. However the satisfaction score of the ER is higher than the BR in general. As in the case of the mouse selection of the user_1, if the result of the initial recommendation is useless to a user, the next recommendation has high score by the previous user selection.

Generally, however, the satisfaction score gradually increases after the user preference was changed as in the monitor selection of the user_2. The score of the mouse selection of the user_3 doesn't increase because the user selects other mousses every week.

7. Conclusions

In this paper, I discussed the design of a middleware for device sharing using a device recommender system and developed a recommender system which is implemented on the UMO developed by our co-working group. I proposed the reasoning engine to infer devices based on context-aware and user preference. Also the paper proposed reasoning techniques based on ontology and rule for applying environmental context and user preference and evaluated each technique. In addition, I proposed and evaluated the method for estimating user preference which is dynamically changed. The reasoning performance ability of our approach shows that the reasoning techniques and recommender system are applicable for device sharing environments. This research proved that realization of the device sharing was possible. In addition, the reasoning performance of the system was demonstrated to be appropriate for the device sharing UMO in peer-to-peer environments. I expect that the result of this paper can be applied for device sharing environments directly, of course, this paper doesn't focus on a sensitive problem such as privacy issue and right management. In the future, I will optimize our system to embed on a watch-type UMO. In our current version, a user requests a service into a UMO directly, however I will deal with the service reasoning that infer a service for the user from a current user situation such as schedule information and real-time tracking information of the user like [18].

References

[1] K. Romer, T. Schoch, F. Mattern and T. Dubendorfer, “Smart Identification Frameworks for Ubiquitous Computing Application,” in *Proc. of PERCOM 2003*, 2003.
[2] M. Weiser, “Some computer science issues in ubiquitous computing,” *Communications of the ACM*, vo.36, no.7, pp.75-84, 2003.
[3] M. C. Lee, H. K. Jang, S. Y. Kim, Y. S. Paik, S. E. Jin, S. Lee, C. Narayanaswami, M.T.Raghunath and M.C.Rosu, “Celadon: Infrastructure for Device Symbiosis,” in *Proc. of UbiComp*, 2005.
[4] Y. M. Jeong, J. H. Kim, J. J. Ko, S. W. Lee, M. H. Kim, and J. C. Choi, “Resource Collaboration Framework based on Context Awareness in Ubiquitous Computing Environment,” in *Proc. of ICUCT 2008*, 2008.
[5] G. Bhatti, Z. Sahinoglu, K. A. Peker, J. Guo, and F. Matsubara, “A TV-Centric Home Network to Provide a Unified Access to UPnP and PLC Domains,” in *Proc.of IWNA*, 2002.
[6] JINI, http://www.jini.org.
[7] M.C. Lee, H.K. Jang, Y.S. Paik, S.E. Jin, and S. Lee, “Ubiquitous device collaboration infrastructure: Celadon,” in *Proc.of SEUS-WCCIA*, 2006.
[8] RDF Primer, W3C Recommendation, February 2004. http://www.w3.org/TR/2004/REC-rdf-primer-20040210.
[9] SPARQL Query Language for RDF, W3C Recommendation, Jan. 2008. http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115.
[10] JENA2: A Semantic Web Framework. http://jena.sourceforge.net
[11] JESS, the Rule Engine for the JavaTM Platform. http://herzberg.ca.sandia.gov/jess.
[12] T. Strang, C.L-Popien, “A Context Modeling Survey,” in *Proc. of UbiComp*, 2004.
[13] T. Gruber, “A Translation Approach to Portable Ontology Specification," *Journal of Knowledge Acquisition*, vo.5, no.2, pp.199-200, 1993.
[14] E. J. Ko, H. J. Lee and J. W. Lee, “Ontology-Based Context Modeling and Reasoning for U-HealthCare,” *IEICE Transactions on Information and Systems*, vol.E90-D, no.8, pp.1262-1270, 2007.
[15] RDFS (RDF Vocabulary Description Language 1.0: RDF Schema), W3C Recommendation, February 2004. http://www.w3.org/TR/2004/REC-rdf-schema-20040210.
[16] DAML. http://www.daml.org.
[17] OWL Web Ontology Language Overview, W3C Recommendation, February 2005. http://www.w3.org/TR/2004/REC-owl-features-20040210.
[18] D. K. Shin, D. G. Shin, Q. C. Nguyen, and S. Y. Park, “Real-Time Tracking of Human Location and Motion using Cameras in a Ubiquitous Smart Home,” *KIIS Transactions on Internet and Information Systems*, vo.3, no.1, pp.84 - 95, 2009.

Jong-Hyun Park received his Ph.D. and M.S. degrees in computer science from Chungnam National University, South Korea, in 2007 and 2002, respectively, and his B.S. degree in computer science from Woosong University, South Korea, in 1999. From 2007 to 2008, he was worked toward researcher in Software Research Center, Chungnam National University, Korea. He was a visiting professor of the Division of Information Technology & Shipbuilding, the Koje Collage for a year from 2009. He is currently a visiting researcher at the Computing and Communications Center of the Kyushu University in JAPAN. His research interests include recommender system, Context-Aware, Ontology, Reasoning, Personalization, Semantic Web, Web Information system, XQuery Processing, XML database, and Database systems.