Modeling of Context-based Interaction Pattern

Sangchul Ahn, Donghoon Kang, and Heedong Ko

Abstract—Providing proactive service requires careful treatment because it has many risks that may force the user to be insufferable with the inadequate services. The context-based interaction pattern is an important factor to personalize the services because it is the practical foundation to adapt to the user. It can be extracted based on the user's activity relative to time, space, and object. In this paper, we describe the methodology to extract user interaction pattern with an example, smart home.

***Index Terms*—Personalization, context-aware systems, proactive service, smart spaces, ubiquitous computing**

I. Introduction

THE purpose of building context-aware system is for providing the proactive services to the user. The system may recognize the situation of the user with some assumptions, and then it provides the related service to user by itself. However, providing proactive service needs to be careful because it has many risks that may force the user to be insufferable with the inadequate services [1]. Actually, the automated services bring both conveniences and inconveniences to the user. The inadequate service gives the stress to the user however more serious problem is that they may occur repeatedly. If an inadequate service is occurred, they may be occurred again and again until the setting is changed. Then, the user may give up using the application.

In this reason, the proactive services may need two elements to judge the accurate service; the user context and the user's preference. The user context has been issued by many researchers of context-awareness [2][3][4]. On the other hand, the user preference hasn't been tackled relatively. We believe the user preference is a key as the methodology of personalization to provide the proactive services. The traditional applications have been applied the user's preference manually. The user has to set the optional preferences, and the application operates according to the specified options. However, the applications in the ubiquitous computing environment need to provide more natural interface gathering the user's preferences because it is not easy to set each preference for the numerous and invisible applications manually [5].

In this paper, we propose the methodology for modeling context-based interaction pattern from the system's experience with the user. In Section 2, we describe the entities of context-based personalization; the user context and system experience. The user context model is explained more detail in Section 3. In Section 4, we describe the user interaction model to extract the context-based user interaction pattern. Context Cube as a visualization tool is proposed in Section 5. Finally, the conclusion and future works are discussed in Section 6.

II. Entities of Context-based Personalization

A. User Context

The user context is the user-centered situational information. It is the basic information to trigger the related services according to the current user's situation. The user context is generally categorized by who the user is, where the user is, what the user is doing, when the user does it, and on which the user focuses. They are the elements to represent the current situation of the user in the interaction space. When the user context indicates a certain situation, the related service may be invoked. Therefore, it is the trigger to call the related services [6].

B. System Experience

The system experience is information that the system can gather from the interaction with the user (e.g., the repeated actions in a certain situation, or the interrupt against the proactive service that the system provides). It means that the system experience is not the directly defined options by the user, but the information which is gathered seamlessly through the interaction with the user. It may be reused as context to decide the service method. It is independent on the pre-designed scenarios, and it can be adapted to the user according to the activity patterns of the user dynamically.

III. Context Model

We distinguish system context (lower-level) from user context (upper-level) as the whole device states available for a context-aware system [7]. The system context is the collection of states for the available resources in the interaction space. The user context is the user-centered information as the trigger to call the related services. We designed ontology to represent the

This research is supported by the ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

Sangchul Ahn is with the Imaging Media Research Center, Korea Institute of Science and Technology, Seoul, 136-751 KOREA. He is also with the Yonsei University, Seoul, 120-749 KOREA. (corresponding author to provide phone: +82-2-958-5637; fax: +82-2-958-5769; e-mail: prime@kist.re.kr).

Donghoon Kang is with the Imaging Media Research Center, Korea Institute of Science and Technology, Seoul, 136-751 KOREA (e-mail: chocopie@kist.re.kr).

Heedong Ko is with the Imaging Media Research Center, Korea Institute of Science and Technology, Seoul, 136-751 KOREA. (e-mail: ko@kist.re.kr).

user context which is classified by activity, space, object and time. In this paper, we describe the methodology to extract the user interaction model with an example, smart home (Fig. 1).

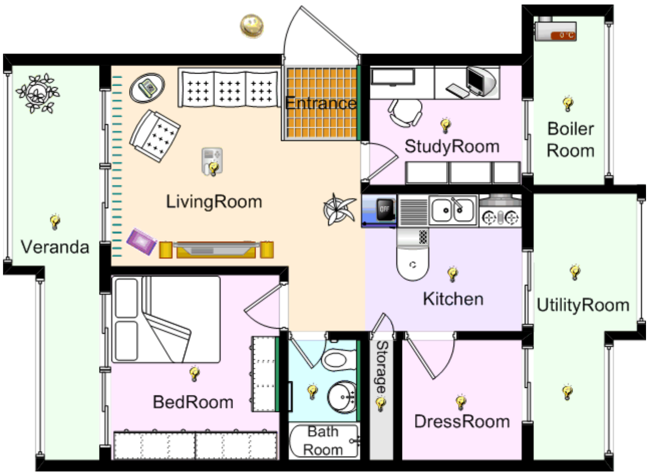

Fig. 1. The world model of smart home example - We are implemented this smart home ontology to validate the proposed methodology. Each device is simulated based on the universal plug and play. The user who is outside the gate can give the events to the system.

A. Space

Space class means the physical domain in the environment (e.g., home). It may have some districts (e.g., living room, kitchen, and bedroom). Each space consists of zones which are the square area in plane coordination. Zone has four properties defining the bounds of each corner (e.g., lower left, lower right, upper left, and upper right). Using the space ontology, the system can map the semantic domain from the physical space. Each object in this space also may have the semantic domain property.

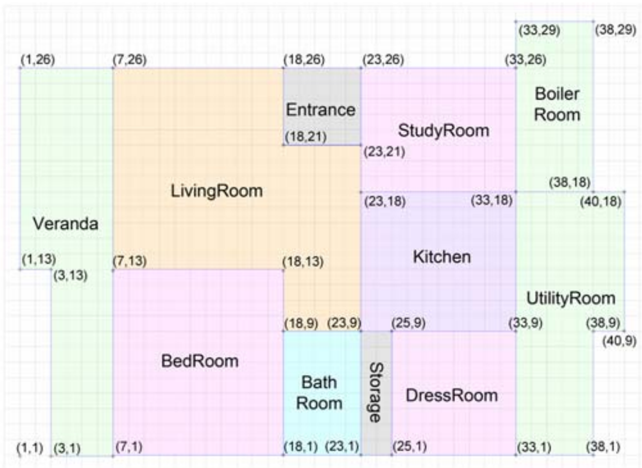

Fig. 2. The space model of smart home – The home has 11 districts. And living room, veranda, and utility room have the 2 zones respectively. The origin of grid (0,0) is at bottom-left of the map. Table I shows the defined zones and their bounds.

B. Object

The system identifies the objects by their own UUID (Universal Unique Identifier). In the user context, the objects may be abstracted and classified according to their own properties

TABLE I
AN EXAMPLE OF SPACE MODEL - APARTMENT

Space	Zone	Bound			
		LL	LR	UL	UR
Entrance	z001	(18,21)	(23,21)	(18,26)	(23,26)
Living Room	z002	(7,13)	(18,13)	(7,26)	(18,26)
	z003	(18,9)	(23,9)	(18,21)	(23,21)
Kitchen	z004	(23,9)	(33,9)	(23,18)	(33,18)
Bed Room	z005	(7,1)	(18,1)	(7,13)	(18,13)
Dress Room	z006	(25,1)	(33,1)	(25,9)	(33,9)
Study	z007	(23,18)	(33,18)	(23,26)	(33,26)
Bath Room	z008	(18,1)	(23,1)	(18,9)	(23,9)
Boiler Room	z009	(33,18)	(38,18)	(33,29)	(38,29)
Utility Room	z010	(33,1)	(38,1)	(33,18)	(38,18)
	z011	(38,9)	(40,9)	(38,18)	(40,18)
Veranda	z012	(1,13)	(3,13)	(1,26)	(3,26)
	z013	(3,1)	(7,1)	(3,26)	(7,26)

A home has the districts; entrance, living room, kitchen, bed room, dress room, study, bath room, boiler room, utility room, and veranda. Each space is composed of some zones which have bound property for lower left, lower right, upper left, and upper right.

and services. Therefore, defining the object ontology requires the common vocabulary which depends on the domain because each object may be manufactured by various vendors and their properties may be heterogeneous. When a new object involves in the environment, the user context should make an instance according to the common vocabulary from the system context. Table II shows the example of light object model based on UPnP device specification [8].

TABLE II
OBJECT PROPERTIES OF DIMMABLE LIGHT

Service Type	Variables	Type	Allowed Value
SwitchPower	PowerTarget	Boolean	true, false
	PowerStatus	Boolean	true, false
Dimming	LoadLevelTarget	ui1	0-100
	LoadLevelStatus	ui1	0-100
	OnEffectLevel	ui1	0-100
	OnEffect	String	OnEffectLevel, LastSetting, Default
	StepDelta	ui1	1-100
	RampRate	ui1	0-100
	IsRamping	Boolean	true, false
	RampPaused	Boolean	true, false
	RampTime	ui4	0-4294967295

This upnp-enabled dimmable light has two available services; switch power and dimming. It is enabled to control these services over the network.

C. Time

Time is the basis to model the user interaction pattern as it increases infinitely, but the conceptual time loops recursively (e.g., daytime, night, day, day of the week, week, month, season, and year). The user's repetitive actions may have the relations with the conceptual time. The time ontology defines this conceptual time, and when the user activity occurred, the user context may be represented with it [9]. Table III shows the time model of our work.

D. Activity

The activity is the model of characterizing the status of the user. It is used as a trigger to invoke the related services. That is, a basis to judge when and how the system reacts. The possible

TABLE III
CONCEPTUAL TIME MODEL

Type	Value
Year	B.C., A.D.
Season	Spring, Summer, Autumn, and Winter
Month	January, February, March, April, May, June, July, August, September, October, November, and December
Week	7 Days
Day of Week	Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday
Day	1~31
A.M.	00:00 ~ 11:59
P.M.	12:00 ~ 23:59
Daytime	06:00~17:59
Night	18:00~05:59
Hour	0 ~ 24
Minute	0 ~ 60
Second	0 ~ 60

The conceptual time model is based on the system timestamp. It can be inferred from the occurring periods of activities. This time model is based on time ontology.

activities in an environment are dependent on the domain. Therefore, the activity ontology defines the whole possible activities. Each activity may be inferred from the system context by its own rules.

TABLE IV
THE EXAMPLES OF ACTIVITIES AND RULES

Activity	Rule		
	Object	Condition	Value
Enter home	Door	IsOpened	true
	Floor Tracker	Weight	80
Taking a shower	User	IsIn	Bath Room
	Shower	IsShowering	true
Watching the Movie	Television	IsPowered	true
		Channel	5 (Movie)
	User	IsIn	Living Room
Sleeping	User	IsIn	Bed Room
	Bed	Weight	80Kg
Shaving	User	IsIn	Bath Room
	Shave	IsPowerd	true
Cooking	User	IsIn	Kitchen
	Oven	IsPowered	true

The activity is a trigger to activate the related services. Each activity has its own rules.

IV. USER INTERACTION MODEL

The user interaction model is a pattern of occurring the user's activities. When the user is on a specific activity, the chained actions may be occurred according to the relativities for the task. It is possible modeling a pattern from the repetitive and chained operations by the user and it would be the useful context which can be reused at the next similar situations. To provide the proactive service which satisfies the user, it is important that the system uses the user's personalized interaction model as a context. There are several elements for modeling user interaction pattern as following:

A. *Time dependency*

This is composed by the repeated activities which are dependent on conceptual time (e.g., waking up, eating, going to church, sleeping). If an activity is occurred recurrently at a certain conceptual time, the activity has a relation with time.

B. *Space dependency*

The whole activities are occurred in a space. However, the activities may have dependencies for the specific spaces (e.g., taking a shower, sleeping, cooking) or not (e.g., listening the music, talking). If an activity is occurred recursively at a certain space, the activity has a relation with space.

C. *Object dependency*

Although the whole activities are not dependent on the specific objects, many activities are dependent on the objects (e.g., sleeping, shaving, listening to music). If an activity is occurred recursively with a specific object, the activity has a relation with the object.

The core dependencies of activity are independent on each other. Therefore, the activity may depend on the whole dependencies or not. Fig. 3 shows that an activity has the relations among time, space, and object dependency.

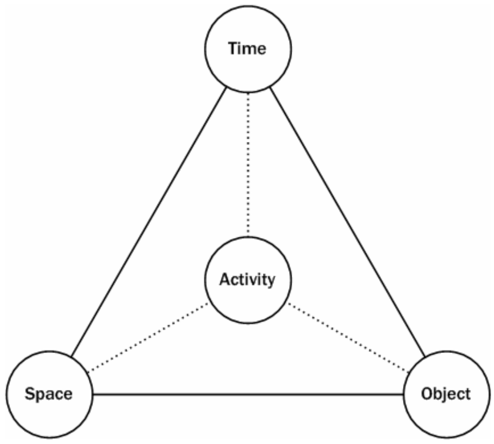

Fig. 3. The core dependencies related with activity. Activity has relations with time, space, and object properties. The degree of relations is different by activity. And the activity may depends on the whole dependencies or not.

There are the additional dependencies for user activity.

D. *Schedule*

The schedule is a reserved activity. If the user has a schedule in a little time, his current activities may relate to the schedule (e.g., morning call, meeting appointment).

E. *Feedback from manual command*

If the proactive service is not triggered, the user may command the task manually. Then, it should be reflected in the next similar situations. And also when the user interrupts and rollback the provided service, the system should be reflected.

V. CONTEXT CUBE

We think visualizing interaction patterns is useful because it makes easier for grasping and analyzing by end-user. Context Cube is our approach to the visualization tool for modeling the personalized interaction patterns in terms of domain activity, space, object, and time properties. Bauer *et al*. denoted their sensor device as 'ContextCube', to serve as an information source [10]. However, it is not the visualization tool like our system, but a physical device. Using Context Cube, the user interaction pattern can be visualized in the three dimensional space. By default, the unit of pattern is the activity of user. And as we have explained in the previous sections, each activity has relations with time, space and object. Therefore, the activity of user can be represented to:

Activity (Time, Space, Object)

It may show just consecutive activity logs at early stage. However, as the specific activity is occurred repeatedly, it grows the possibility that can be occurred at next similar situation. Context Cube has five properties defining the possible elements depends on the domain as following:

A. Domain Activity Property

Domain activity property defines the whole possible user activities in the interaction space. As we mentioned in section three, the activity is defined with its own rules. When the environment situation satisfies the rules, the activity is occurred. Each activity has the weight variable and it grows when the same activity that has the similar relations with space, object, or time properties.

B. Space Property

Space property defines the boundary and districts of the physical domain. A space may contain objects or users; therefore it also groups the objects surrounding the user.

C. Object Property

Object property contains the current object resources available in the interaction space. The whole objects have the location attribute which is related with space property.

D. Time Property

Time property consists of the interpretation factors classified by the level of detail. The conceptual abstraction of time can be the period of user activity (e.g., Monday, July, and evening).

E. Personalized Interaction Pattern

Personalized interaction pattern is a set of the activities. As the specific activity is occurred repeatedly, its weight value will be greater by counting.

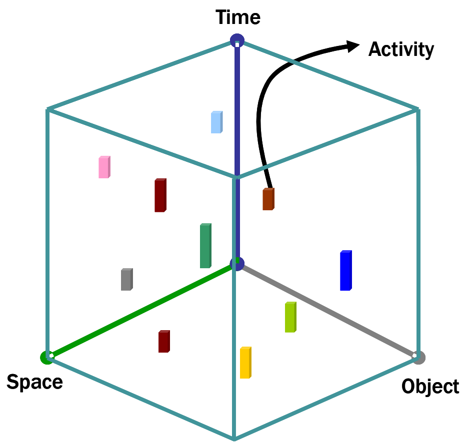

Fig. 4. The Context Cube – The activities is shown in the cube. The different color means the various activities. Time is recursive in terms of classes by level of detail. If the activity is occurred frequently, the weight of the activity grows.

VI. Conclusion and Future Work

There are several benefits when we model context-based interaction pattern using Context Cube. First, we can model multiple personalized interaction patterns for each user. As the each end-user may set their own interaction patterns, the user would fine it very useful, and as it is stated earlier, some of those patterns would be found automatically and it would be very user-friendly.

Second, the 3D visualized model of Context Cube is easy for understanding and customizing the services by end-user. Because each interaction patterns are related with something they are used to (i.e., time, space, and object) the user would find it easy to understand new patterns or setting new patterns for the future use.

Third, this generalized model can be applied to the various domains. The three factors of the Context Cube are universal so it should be able to apply to the other areas of the fields. Although it is not proof of the rules of natural world, we believe this generalized context-based personalization model assists to develop the personalized and proactive services. In future work, we will apply the probabilistic model to extracting the interaction pattern.

Acknowledgment

We thank to Hyun-Jhin Lee. The working of Context Cube is inspired from her research.

References

[1] Owen Conlan, Ruaidhri Power, Steffen higel, Declan o'Sullivan, and Keara Barrett, "Next Generation Context Aware Adaptive Services," in 2003, Proceedings of the 1st international symposium on Information and communication technologies, pp. 205-212.

[2] William Noah Schilit. "A System Architecture for Context-Aware Mobile Computing", Ph.D. dissertation, Columbia University. 1995.

[3] Anind K. Dey, "Providing Architectural Support for Building Context-Aware Applications", Ph.D. dissertation, Georgia Institute of Technology. 2000.

[4] Albrecht Schmidt, "Ubiquitous Computing – Computing in Context" Ph.D. thesis, Dept. Computer Science., Lancaster University, 2002.

[5] Sven Meyer and Andry Rakotonirainy, "A Survey of Research on Context-Aware Homes", Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003 - Volume 21, 2003, pp 159-168.

[6] Joyce Ho and Stephen S. Intille, "Using context-aware computing to reduce the perceived burden of interruptions from mobile devices", Proceedings of the SIGCHI conference on Human factors in computing systems, 2005, pp 909-918.

[7] Donghoon Kang, Sangchul Ahn, Heedong Ko, Weduke Cho, and Youngtack Park, "Context Awareness for ubiquitous Computing System", Journal of Korea Intelligent Information Systems Society 2004-Vol.1, 2004.

[8] Universal Plug and Play Device Architecture Version 1.0 Available: http://www.upnp.org/download/UPnPDA10_20000613.htm

[9] Jerry R. Hobbs and Feng Pan, "An ontology of time for the semantic web", ACM Transactions on Asian Language Information Processing (TALIP) Volume 3, 2004, pp. 66-85.

[10] Martin Bauer, Christian Becker, Jorg Hahner, and Gregor Schiele, "ContextCube - Providing Context Information Ubiquitously," Proc. on Distributed Computing Systems, 2003, pp.308-313