A Framework for Designing Interaction as Learning Process

Supporting Users' Development of Domain Knowledge through Interaction

Jung-Min Choi* Keiichi Sato**

** Institute of Design, Illinois Institute of Technology*
Chicago, Illinois, USA, jungmin@id.iit.edu
Oriental Culture & Design Center, Kookmin University
Seoul, Korea, choijm@kookmin.ac.kr
*** Institute of Design, Illinois Institute of Technology*
Chicago, Illinois, USA, sato@id.iit.edu

Abstract: Technological aids provided by products and systems have reduced people's problem-solving efforts in everyday practices. It is crucial for users to learn how to operate a system, and yet their acquisition of knowledge about domain problem-solving is largely disregarded in the current design process. Without enough domain knowledge, users cannot creatively adjust their system use to meet their variable needs, and users' experiences with a system can be confined to step-by-step operational procedures. Therefore, systems should be designed to support users' development of domain knowledge through interaction. This research proposes the Learning-Based Approach, which emphasizes the incorporation of users' domain knowledge into the system design processes. Example products are selected to more effectively explain the difference between the Learning-Based Approach and the existing approaches. By supporting users' domain knowledge development, system designs will be able to help users produce higher-quality results and a higher-level of satisfaction.

Key words: product and system design, user interface design, design methodology, operation and domain knowledge, users' learning process

1. Introduction

Technological aids provided by interactive products and systems have reduced people's problem-solving efforts in everyday practices. In order to improve user-system interaction, designers are mainly concerned with how to help users solve problems and achieve goals more easily and efficiently. Clearly, it is crucial for users to learn how to efficiently operate a system, and yet their acquisition of problem-solving knowledge of the task domain is largely disregarded in the current design process. As a result, domain knowledge has not been appropriately incorporated into system design, and users have lost opportunities to learn about domain practices through interaction [3,4].

Take, for example, the use of a digital camera. As depicted in Figure 1, users try to take a satisfactory picture through the interaction with a digital camera via its user interface design. In the current practice of using a camera, user experience mainly consists of step-by-step machine operational procedures, such as pressing button

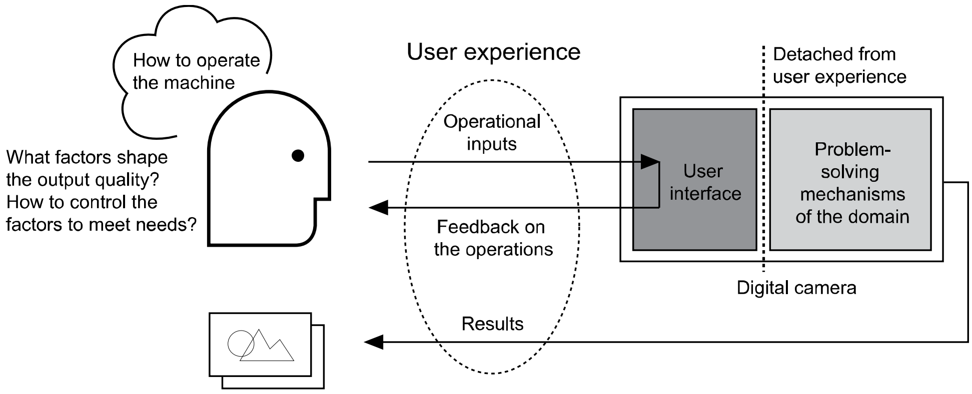

Figure 1. Users' operations of a system, detached from the reality of domain practice

A, and then choosing option 5, etc. Throughout these processes, users can receive the output or finished product, i.e. the photos. However, users may not be aware of the specific factors that shaped the quality of the photos, nor are they knowledgeable about the ways in which problem-solving mechanisms may be manipulated in order to achieve satisfactory photos. Those problem-solving mechanisms (i.e. photo-taking principles) are not attainable and acquirable through the user interface designs, but are, rather, hidden behind the "black box" type of user interface design, detached from user experience.

Without enough domain knowledge, users' problem-solving can be confined within system's functioning, and they will not be able to manipulate their system use in order to achieve satisfactory results. In order for users to fully achieve their goals, they should be able to creatively utilize tools to meet their variable needs. Therefore, it is not sufficient for designers to consider how to help user better operate the interface elements; rather, designers should also consider how to support users' ability to learn the "hidden" domain problem-solving mechanisms. In other words, domain knowledge should be properly embedded in products and systems and communicated through user interface designs so that users can obtain enough domain knowledge through interaction.

This new approach to product and system design is named the Learning-Based Approach. This research aims to propose a methodology by which designers can effectively embody the concept of Learning-Based Approach in design practices. In order to frame the approach, this particular paper first reviews existing theories and research on users' knowledge development in system use and knowledge manipulation methods in system design practice. Based on this review, basic mechanisms of the Learning-Based Approach are proposed, describing how domain knowledge can be incorporated into design processes and acquired by users through interaction. Finally, some case studies are conducted to explain the proposed approach using real design examples.

By emphasizing users' development of domain knowledge, this research is expected to contribute to user-system interaction in different ways. Developing domain knowledge will constantly increase users' problem-solving ability to produce higher-quality results when irregular or unexpected situations occur. In other words, with enough domain knowledge, users will be able to creatively utilize the system in order to meet their variable needs. As a result, users will have more convivial experiences and higher-levels of satisfaction overall. Also, the

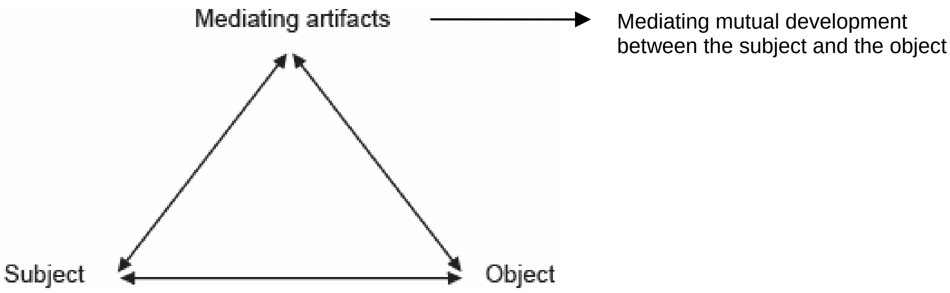

Figure 2. Artifact-mediated interaction and development (adapted from [5])

Learning-Based Approach will help designers to develop more effective methods of user support. By supporting users' domain knowledge development, a system can provide users with more robust support in achieving their goals.

2. Interaction as Learning Process

2.1 Learning process inherent in user-system interaction

Users' knowledge development (i.e. learning) is an inherent process in user-system interaction. When people use an interactive system or product to achieve a goal, they constantly utilize existing knowledge and obtain new knowledge. Throughout these processes, the users' knowledge is constantly modified and generated. Focusing on the constructive knowledge development in using a system, this research considers user's interaction with a system as their learning process.

Cultural historical activity theory provides a fundamental explanation about knowledge development phenomena in artifact-mediated human activities. Activity theory studies different forms of human practices focusing on the mutual developmental changes in the interaction between the subject and the object, and emphasizes the important role of artificial tools in this development [5,10]. According to Lektorsky [11], when an entity is combined with human needs, it becomes an object of activity. The subject constructs the object, "singling out those properties that prove to be essential for developing social practice (p. 137)," and interacts with the object, mediated and supported by artificial tools. Based on Lektorsky's theory, Engeström [5] illustrates human activities as the triadic representation, as shown in Figure 2. When adopted for system design, the activity theory effectively explains the developmental changes when computing artifacts are involved in human practices. From this viewpoint, the main concern in system design is how to create tool mediation to support users' knowledge development (i.e. learning) through system use.

2.2 Knowledge needed to achieve goals using a system

In the system design field, there have been several different approaches to designing which enhances learning, including considerations of learnability as a usability issue (e.g., [12]), and user support tools such as help systems and user's manuals (e.g., [2]). Most of these approaches are focused only on the knowledge process between the subject and mediating tool, i.e. how to help a user ("subject") gain enough knowledge for operating a system ("mediating artifact") so that he/she can become an expert user of the system as quickly as possible.

Focusing on users' acquisition of knowledge about how to operate a system cannot, however, fully address the knowledge needed to achieve the object in their system use. As Kaptelinin [9] points out, system designers and researchers should keep in mind that the main reason people use interactive systems/products is to "achieve some goals that are meaningful beyond system use (p. 106)." In other words, people need tools to achieve their "object of activity"; proficient operations in using a system ("mediating artifact") may not be the main purpose of their system use.

Some researchers argue that the role of a system should be a "means" rather than an "ends." For example, Illich [6] emphasizes people's autonomous and creative actions in their relationships with tools and their environment, using the term *conviviality* (p. 11; p. 20). In other words, people should have freedom to decide how to utilize tools, rather than passively adjusting their tool usage to the pre-determined, particular ways of use. The author uses the term *convivial tools* in order to refer to tools that allow users' flexible utilization of them to meet their own needs. In order to provide users with more *convivial* experiences, a system should allow users to have more power and a sense of efficacy in their interactions so that they can be encouraged to self-determine more meaningful methods of system use. Similarly, Norman [13] postulates that users tend to have lost control over interaction and have become too dependent upon by technologies. According to the author, although automatic technologies are rapidly being developed, it is impossible for designers to completely prefigure users' intentions, behaviors, and the environmental factors that can intervene in the pre-planned structure of system use. The difficulty in controlling such variations in real situations often cause systems' malfunctioning, and as a result, can cause users' frustration. In order to solve this problem, Norman suggests that a design should allow users to monitor their interaction with the system so that users can take control whenever the system cannot deal with the contextual complexity.

In order to be able to flexibly adjust system operations to meet various needs and to consequently achieve satisfactory results, it is necessary for users to have sufficient knowledge about problem-solving mechanisms in the domain. In current design practice, however, designers' efforts to support users' interaction with a system are focused on their efficient learning of system operational procedures; users' acquisition of knowledge about domain problem-solving is largely disregarded. When user interfaces are not designed to reflect the reality of domain practice, users may not have enough opportunities to obtain domain problem-solving knowledge. Therefore, system designers should be concerned with how to provide appropriate support for users' acquisition of domain knowledge.

3. A framework of Learning-Based Approach

3.1 Types of knowledge in system use

Knowledge involved in interaction can be classified into two types: knowledge about how to operate a system, and knowledge about problem-solving mechanisms in the domain of concern [3,4]. Domain knowledge can include the taxonomy, action sequences, and cause-effect relationships which comprise the processes and methods of problem-solving in a domain of concern. For example, when people bake bread using a bread machine, they may need knowledge to operate the machine itself (operation knowledge) as well as knowledge about the general principles of baking (domain knowledge). Figure 3 explains the involvement of these two types

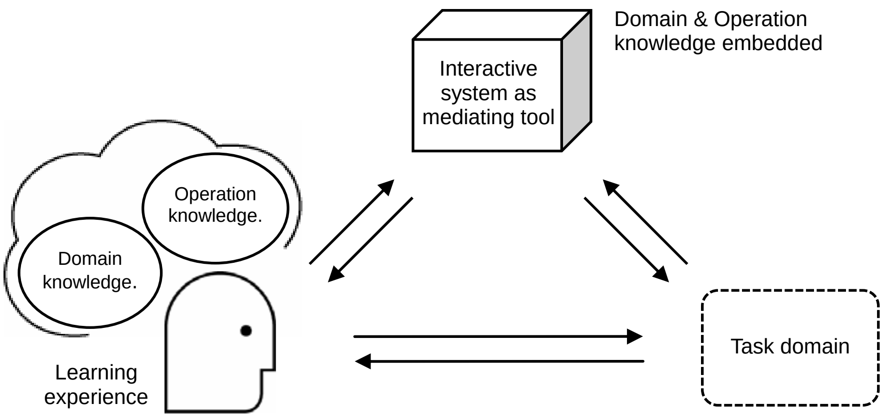

Figure 3. Types of knowledge involved in interaction

of knowledge in system use. This model is based on the tool-mediated subject-object relationship proposed by activity theory, which is mentioned earlier (refer to Figure 2). In the present research, the term mediating artifact is replaced with the term 'interactive product or system,' the term subject with 'user,' and the term object with 'task domain.'

The current focus of system design tends to be on the user's system operational knowledge. However, users' primary purpose of system use is to achieve goals in the domain of concern. For example, when a user is operating a bread machine (means/mediating tool), his/her goal may be to make satisfactory bread (ends/object), rather than to proficiently operate the machine. In order to help users achieve satisfactory results through system operations and have more control in the process, systems should be designed to support users' acquisition of task domain knowledge, rather than remaining limited to their acquisition of operation knowledge. Considering the significant roles of domain knowledge in system use, this research proposes a new approach to interactive system and product design, named the Learning-Based Approach. When users understand the mechanisms of the task domain that shape the quality of the results, they will be able to adjust system operations to meet their variable needs. In addition, the acquisition of domain knowledge will constantly provide users with more convivial experiences with system use and higher-level satisfaction overall.

3.2 Mechanisms of the Learning-Based Approach

This paper proposes a conceptual framework that can help designers implement the Learning-Based Approach in system/product design practice. The framework, illustrated in Figure 4, provides the explanations of how domain knowledge can be appropriately embedded in a product or system and can be obtained by users through interaction.

Phase 1. Designers' understanding of domain knowledge structure and users' tasks. First of all, designers should understand the structure of domain knowledge and should identify users' tasks in using a system. For example, when a designer is developing a bread maker, he/she needs to know the materials, methods, and processes of baking (i.e. domain knowledge structure) and should analyze what users have to do in using a bread maker (i.e. users' tasks). Task analysis is considered a typical method for developing a human-centered

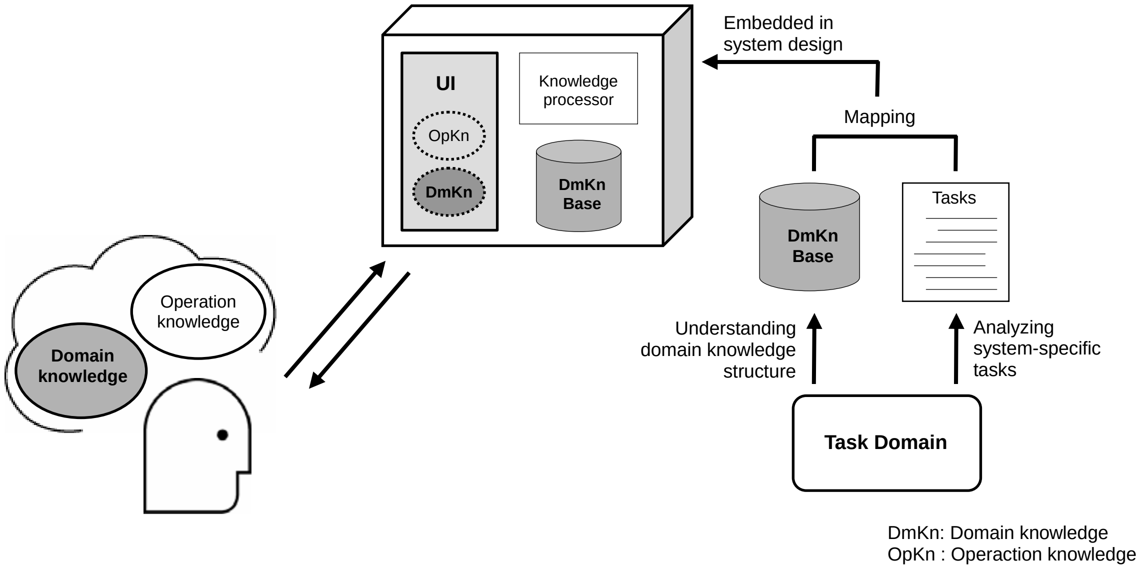

Figure 4. A conceptual framework of implementing the Learning-Based Approach

product/system development (e.g. [8]). The previous task analysis methods do not, however, care about *users'* understanding of the task domain. Rather, most of the methods are concerned only with *designers'* better understanding of users' goals and tasks in the domain so as to support users' goal achievement. However, a more effective and robust way of supporting users will be to help them develop their own knowledge about how to achieve goals. The critical argument of this research is that domain knowledge should be incorporated into system design processes for the purpose of users' knowledge generation, not limited to the designers' purpose of system development. In order to develop a system that provides users with domain knowledge, designers need to build up a domain knowledge base which contains the knowledge structures of domain problem-solving mechanisms. Then, they should consider the delivery of the domain knowledge structures to users through the user interface designs.

Phase 2. Mapping between domain knowledge structure and users' tasks. Once designers identify tasks and domain knowledge structure, they should appropriately link particular domain knowledge objects to particular tasks. In other words, it is important for designers to determine what kinds of domain knowledge objects can be helpful for users' better performance in executing particular operational tasks. For example, when a digital camera user needs to select an ISO number out of seven for taking a bright picture, he/she should have some knowledge about photography in order to select an appropriate value. In this case, the relevant domain knowledge to be provided for users can include the terminology related to the ISO control and the cause-effect relationships among the sensitivity to light (i.e. ISO), the amount of light inflow, photo brightness and clarity, etc. These domain knowledge objects should be mapped onto relevant operational tasks in the design process.

Phase 3. Delivery of domain knowledge via UI designs. Domain knowledge embedded in a system or product should be delivered to users through interaction. As previously depicted in Figure 1, most of the UI designs of the current systems do not provide enough opportunities for users to obtain domain knowledge embedded in the system. That is, domain knowledge is not appropriately communicated in user-system interaction. UIs of products and systems should be designed to deliver domain knowledge so as to support users' learning of the domain mechanisms.

Phase 4. Users' constant development of their domain knowledge. Simply displaying domain knowledge via the UI elements may not the ultimate goal of the Learning-Based Approach. Rather, a system needs to encourage users to actively obtain, accumulate, and generate knowledge through interaction. To this end, designers should also consider strategies for initiating and constantly motivating users' learning processes. By providing domain knowledge for users in an appropriate manner, a system will be able to help users not only effectively obtain domain knowledge, but to also maintain their domain knowledge acquisition for higher-level satisfaction.

4. Case studies

Some example products are selected to more effectively explain the distinctions between the Learning-Based Approach and other existing approaches. The examples show the importance of users' domain knowledge in interaction as well as how particular designs can support or hinder users' acquisition of domain knowledge.

4.1 UI design of interactive consumer products

The Learning-Based Approach can be adopted to improve the current UI designs of consumer products and systems. Let us take the example of digital cameras. When using a digital camera, people need to know how to operate the camera UI elements (operation knowledge) and how to manipulate various factors for photo quality, such as light inflow and shutter speed (domain knowledge). In the observational research on digital camera use, Choi and Sato [4] conclude that people's domain knowledge is critical to the effective and satisfactory use of digital cameras. For example, users who were familiar with camera operational procedures, but not with photographic principles, could not take quality pictures when an unusual situation occurred, and could not utilize the full capability of the camera.

Some existing digital camera designs offer user-supporting features by which users can more easily learn how to use the camera, including function descriptions of individual UI elements and guided tours. However, most existing digital cameras do not provide appropriate assistance for users' learning about photographic principles. The existing camera UI designs can be divided into three categories in terms of the levels and types of learning-supporting features, as shown in Figure 5. The function descriptions of individual UI elements (category B) may be more helpful for users' learning of operating a specific camera, compared with the simple presentation of the short name of each icon (category A). Also, some cameras more actively guide step-by-step operations in order for users to efficiently get accustomed to using the camera, by providing a separate guide option (category C). However, user support in all of these categories fails to support users' acquisition of enough domain knowledge. The function descriptions of category B are intended to simply inform users of the meaning of particular UI icons for easier camera operation. Although the guided tour of category C may allow users to indirectly discover domain mechanisms through the operations, such as the relationships between relevant photographic factors, their primary role is to guide the operational procedures. In order to support users' development of domain problem-solving capability, systems should be designed to enable users to learn about photography through interaction.

A. Low-level support for operation and domain knowledge

B. Medium-level support for operation knowledge, but low-level for domain

C. High-level support for operation knowledge, but medium-level for domain

Figure 5. Different levels and types of learning support in camera UI designs

Coffee makers can be taken as another example. In the design and use of coffee makers, domain knowledge can be coffee-making principles, such as the cause-effect relationships among the amount of coffee, coarseness of coffee grounds, the amount of water, and the properties of coffee taste. When users want to brew coffee using a coffee maker, they need to apply certain amount of knowledge not only concerning how to operate the machine, but also how to produce satisfactory coffee by manipulating some factors of coffee quality. In the previously-mentioned user studies by Choi and Sato [3], the authors investigated the use of a coffee maker, focusing on users' utilization and development of domain knowledge. The results also showed the importance of users' domain knowledge in interaction; users who had general coffee-making knowledge (domain knowledge) more effectively accessed their experiences with other coffee makers in order to use the given new machine, and more creatively found alternative methods when unexpected situations occurred. Therefore, in order to help users increase their ability to produce satisfactory coffee in irregular or changing situations, designers need to consider how to help users expand their knowledge of coffee-making through interaction.

4.2 Design for promoting sustainable behaviors

The Learning-Based Approach can also provide a new perspective on sustainable design; products and systems that are designed to support users' learning of energy-saving mechanisms (domain knowledge) will be helpful for promoting sustainable behaviors. Blevis [1] postulates that sustainability should be a core topic of interactive system design. In particular, the author calls upon designers to consider how to motivate users' sustainable behaviors through technologies and designs. One of the common strategies adopted for encouraging environmentally friendly behaviors in everyday lives is to design products and systems that can allow users to be aware of the effects of their behaviors on the environment (e.g. [7]). For example, in order to encourage people to save energy, these types of products and systems explicitly show the impacts of their energy use behaviors, in the form of the resulting costs or the history of use. Fuel consumption indicators in car information systems can be another example. Hybrid cars that use two or more fuel resources usually display the information about the type of fuel that is being used and the current states of energy use, especially focused on the gas consumption efficiency. These approaches assume that by making people's invisible behavioral patterns visible, systems can help people consciously increase their sustainable behaviors. Although awareness-increasing designs may contribute to promoting people's sustainable use in an indirect way, simply viewing the impacts of behaviors is not sufficient for generating the ability to determine appropriate choices for the environment.

In order to help people actively manage their sustainable behaviors through interaction, there must be more effective ways to develop systems to help users obtain knowledge about utilizing systems in a sustainable way (i.e. problem-solving knowledge in the domain of sustainability). In this case, domain knowledge can include the understanding of the factors that may affect the environment, the mechanisms by which the factors shape positive or negative influences on the environment through system use, and the ways in which users can reduce their environmentally negative behaviors by properly manipulating the factors through interaction. Take, for example, a system for indicating the current states of using various energy sources in a building. If the system provides users with the knowledge about the relationships among different factors – such as water, electricity, gas, air pollution, etc. – and informs the mechanisms for changing the current states, users will be able to develop their own ability to determine better ways of using energy. The Learning-Based Approach emphasizes users' acquisition of autonomous abilities for increasing their sustainable behaviors through system use. This goal can be effectively achieved by allowing user not only to *perceive* their environmental impacts, but also to have opportunities to *learn* about the mechanisms that shape and control the impacts.

5. Conclusions

This research proposes the Learning-Based Approach to product and system design. This approach emphasizes that products and systems should be designed to support users' development of domain knowledge through interaction. In other words, it is important for designers to consider how to help users increase their domain problem-solving capability through system use. The present paper first reviews existing theories and research about users' learning processes in system use. Then, basic mechanisms of the Learning-Based Approach are proposed, describing how domain knowledge can be incorporated into design processes. The mechanisms include: 1) Designers' understanding of domain knowledge structure and users' tasks, 2) Mapping between domain knowledge structure and users' tasks, 3) Communication of domain knowledge via UI designs, and 4) users' constant development of their domain knowledge. Finally, some case studies are conducted to explain the proposed approach using real design examples. By supporting users' learning of domain problem-solving mechanisms, products and systems can help them constantly increase their ability to produce quality results. Also, with enough domain knowledge, users will be able to creatively utilize systems in order to meet their variable needs. As a result, users will have more convivial experiences and higher-levels of satisfaction with overall system use.

The future development of this research will include the development of a more detailed design methodology for embodying the Learning-Based Approach in design practice. Then, the methodology will be demonstrated in the processes of developing a design prototype. Finally, observational user studies involving the prototype will be conducted in order to evaluate the effectiveness of the overall methodology.

6. References

[1] Blevis, E. (2007) Sustainable interaction design: invention & disposal, renewal & reuse. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, pp 503-512.

[2] Carroll, J. M. (1990) *The Nurnberg Funnell: Designing Minimalist Instruction for Practical Computer Skill,* MIT Press, Boston, MA.

[3] Choi, J. and Sato, K. (2008) Interaction as Learning Process: Incorporating Domain knowledge into System Use, In *Proceedings of NordiCHI 200*8, ACM Press.

[4] Choi, K., Choi, J. and Sato, K. (2008) Socio-Cultural Factors for New Product Acceptance in Home Environment Design, *Journal of the Human-Environmental System,* vol. 11, no. 1, pp 65-71.

[5] Engeström, Y. (1999) Activity Theory and Individual and Social Transformation, In Y. Engeström, R. Miettinen, & P. Punamäki (Eds.), *Perspectives on Activity Theory,* Cambridge University Press, Cambridge, UK.

[6] Illich, I. (1973) *Tools for Conviviality,* Harper & Row Publishers, New York.

[7] Jacobs, M., Löfgren, U. and Mazé, R. (2005) Free Energy: Alternative Designs for Awareness and Choice, In *Proceedings of Pride and Pre-Design Conference*.

[8] Johnson, P. (1992) *Human-Computer Interaction: Psychology, Task Analysis and Software Engineering,* McGraw Hill, London, UK.

[9] Kaptelinin, V. (2001) Activity Theory: Implications for Human-Computer Interaction, In B. A. Nardi, (Ed.), *Context and Consciousness: Activity Theory and Human-Computer Interaction*, 3rd Ed., MIT Press, Boston, MA.

[10] Kuutti, K. (2001) Activity Theory as a Potential Framework for Human-Computer Interaction, In B. A. Nardi, (Ed.), *Context and Consciousness: Activity Theory and Human-Computer Interaction*, 3rd Ed., MIT Press, Boston, MA.

[11] Lektorsky, V. A. (1980) *Subject, Object, Cognition,* Progress Publishers, Moscow.

[12] Nielsen, J. (1994) *Usability Engineering,* Morgan Kaufmann Publishers, San Francisco, CA.

[13] Norman, D A (2007) *The Design of Future Things*, Basic Books, New York.

"This work was supported by the Korea Research Foundation Grant funded by the Korean Government" (KRF-2008-411-J05001)