The proceeding of World Congress on Mass Customization and Personalization, Hong Kong, October 1-2, 2001

User Process Based Product Architecture

S. Teeravarunyou, K. Sato

Institute of Design
Illinois Institute of Technology, Chicago, USA

Abstract

Mass customization is a phenomenon in design, business and manufacturing across many industries. It tailors the complexity of products to serve individual user needs. Nevertheless, this complexity is not only with regard to structuring the product, it also must consider the complexity of user behavior in the context of use. User knowledge and User Process based Product Architecture is a way that brings the right product to the right users. A proposed framework and method is one mechanism to obtain knowledge regarding the customization needs of users, which in turn generates further knowledge for designing the product.

Keywords: User-Centered Design, User Process Architecture, Mass Customization

1 INTRODUCTION

The traditional design process begins with a study of user applying marketing research through the production process. Then a product development team comprised of Industrial designers and Engineers receives the user data from the research to construct the product. It was not until recently that many technology firms began to incorporate user data from ethnographic and anthropologic research into a social and cultural context. Even though the user centered design concept is incorporated into the development of the product, the method for understanding users is not well defined and the mechanisms for carrying through with the user data in the development process is not well established. Therefore, this research uses the concept of knowledge lifecycle (see figure 1) composed of different types of knowledge which emerge in the product development lifecycle. These two types of knowledge interplay between users and designers in the context of use and design.

This paper tries to identify two problems that exist in the links between this knowledge. First, the problem for the designer is to design a product that users find predictable and intuitive even though each user perceives a product from a different perspective. Mass customization shows the difficulty of managing the complexity of products according to user needs. For an example, the designer's model of the product does not accommodate the user's expectation of use. "User Process Based Product Architecture" is a concept introduced to develop a platform to accommodate supporting complex product variation and customization.

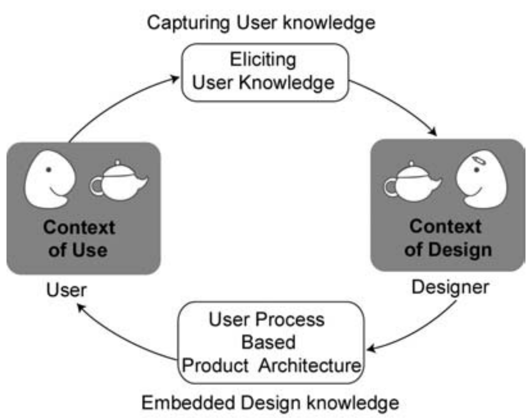


Figure 1: Knowledge Lifecycle

Second, user knowledge is not only what users know about products but it also explains the user behavior of using the product. It is incorporated into the process of learning and problem solving experiences with the product in the context of use. Users try to adapt and further modify the intended way of use, functions and attributes in order to adapt to given situations outside the design consideration [10]. Nevertheless, user knowledge is less accommodating in the design phase. The capturing and integration of user knowledge into products engages users indirectly to design and it creates the possibility of incremental changes.

The goal of this research is to develop a new framework - design process for mass customization, and a method- User Process Based Product Architecture - UPPA that allow product planners to serve the concept of diverse user needs. The framework in this study presents a way to build architecture from the research and design in three areas – identifying user knowledge, integrating user knowledge into product solutions, and mapping user's conceptual models to the design plan.

2 EXISTING ARCHITECTUAL FRAMEWORK AND METHODS

In order to propose a new framework, the existing architectural frameworks and methods were examined to find important variables in architectural framework.

2.1 Architectural framework

A framework of product architecture enables the planner to position issues and manage the complexity of the architecture through abstraction.

Framework	**Zachman framework Enterprise architectural framework** [12]	**4+1 View model in Unified Modeling Language (UML)** [2]	**Product Family Architecture PFA** [5]
Field	Business and Information	Software Engineering	Product Design
Goal	To manage the enterprise system	To partition the architecture of software into technical and business architecture	To create a product family and variation
Level of architecture	*Role in Enterprise Design Process:* 1.Owners-Business Objective scope (contextual model) 2. Designers Enterprise model (conceptual model) 3. Builder-Manufacturing System model (logical), Technology constrained model (physical) and detail representation (out off context) *Product Abstraction:* 1.Data (Product) 2.Function (Process) 3.Network	*Architectural Views:* 1. Conceptual View Major design elements and their relationships 2. Module View Functional decomposition, interfaces and layers 3. Code View Organization of source code, libraries, and binaries 4. Execution view The dynamic structure of the system	*Modularity Views:* 1. Functional view 2. Behavioral view 3. Structural view *Issues for developments:* 1. Modularity 2. Commonality 3. Integrateability

Table 1: Architectural framework

Table 1 shows examples of architectural frameworks for different types of systems represented in different perspectives and dimensions of abstractions. Zachman's framework shows the enterprise structure in multiple levels. Product Family Architecture-PFA and 4+1 view model focus on the product level. Three of them have the common element in terms of a functional view. Zachman described functions as processes separated from the data or system entity. In the 4+1 view model and PFA, functions can be considered as system function in products.

Most of architectural frameworks integrate the modularity as a significant factor since it is the foundation for the product variant and flexibility. Notice that the user view is not well represented in these frameworks. The new framework places an emphasis on the user view from research through the product development process.

2.2 Architectural modularity in Product, Software and Information design

Existing concepts of architecture in the different areas of application provides a useful guideline to develop other types of architecture such as user process architecture since they share the same principle in term of architecture scheme, modeling techniques, entity characteristics and module relationship.

In Table 2, the architectural schemes of products have the same fundamentals of mapping functions to the components. Only the classification methods are different. The main goal of classification in many methods is to cluster components based on functions. Function structure by Otto [6] is a concept to identify products based on their grouping functions without knowing their components beforehand while the Sequence diagram has already established the components based on the component discoveries in Robustness Analysis.

	Product Architecture [6]	**Software Architecture** [2]	**Information Architecture** [1]
Architecture scheme	Mapping functions to physical components	Mapping functions to classes	Mapping functions to information entities
Classification method	Function Structure	Use Case Model, Robustness Analysis, Sequence Diagram	CRUD matrix between functions and entities Create, Read, Update, Delete
Modeling Technique	Structural model	Object oriented model	Information model
Entity **- Object** **- Function** **- Attributes**	 Physical Object Product function Characteristic of Products	 Class System function Class properties	 Content module Functions Attributes
Module Relationship **- Interface** **- Interaction**	 Physical interface Flows across interface - Material, Information, Energy	 Input and output of functions and attributes Flows across interface - data	 Input and output of functions and attributes Flows across interface - information

Table 2: Architecture concepts in different product natures

3 DESIGN PROCESS FRAMEWORK FOR MASS CUSTOMIZATION

A new design process framework was proposed by using the principle of architectural framework. It contains four elements – user knowledge, user process, product and function (see figure 2). From the analysis of Zachman's framework, the functions should be separated from the product and the process since it has the potential to be an element linking between them according to the analysis in table 2. In the same way, the process should also independent from the product since it creates flexibility in process. The framework regarding user-centered design has three parts – User Knowledge Identification, User Knowledge Integration and, User Process Mapping. These elements are present in user research, knowledge transfer and product development respectively.

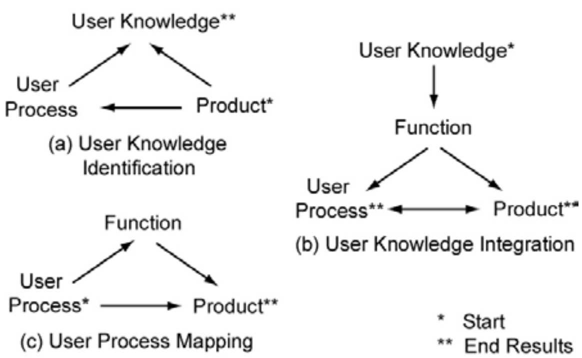


Figure 2 : Design Process Framework

The design process framework presents three stages to enable the goal of bringing the right product to the right users.

Stage 1: Identifying user knowledge from the existing products to the user process. This includes the user needs identification (figure 2a).

Stage 2: Integrating user knowledge as patterns to the existing product through the functions (figure 2b).

Stage 3: Mapping the designer's model to the user's conceptual use from the user process to the product configuration (figure 2c).

4 USER KNOWLEDGE IDENTIFICATION

The diagram in figure 2a presents how to find the users knowledge by using product and user process as a means to trigger the user knowledge at the end. Previous work of the Object Mediated User Knowledge Method [10] – OMUKE is a method being proposed as a way to capture user knowledge that helps design teams develop clear insight into user needs and use context. This method is based on participatory design and applied user observation.

Triggering Mechanism

The mechanism behind this method follows the framework in figure 2a – (see the arrow go directly from the product to the user process). The representation of objects is used as a triggering mechanism for users to externalize their knowledge from objects and process that would otherwise remain inaccessible.

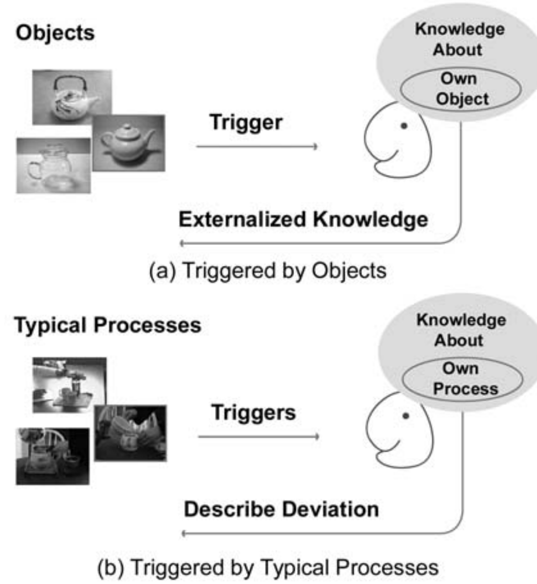


Figure 3: Triggering mechanisms

Viewing objects helps users to make unbiased reports of their experiences without influence from existing norm use. In this method, the photographic image of artifacts is a conceptual representation of a real artifact. It leaves the users enough space to discuss what the characteristic is and how it is used. The second triggering mechanism is a typical process – well understood among users. Users were asked to describe what parts of the process presented in the video clips were alike or different from their processes. The process helps users describe the actual experience through a given example.

Inquiry Decomposition

User experience and knowledge can be analyzed by looking into the structure of the user process and the object used. The users behavior can be represented as a hierarchical structure produced by Hierarchical Task Analysis – HTA [9].

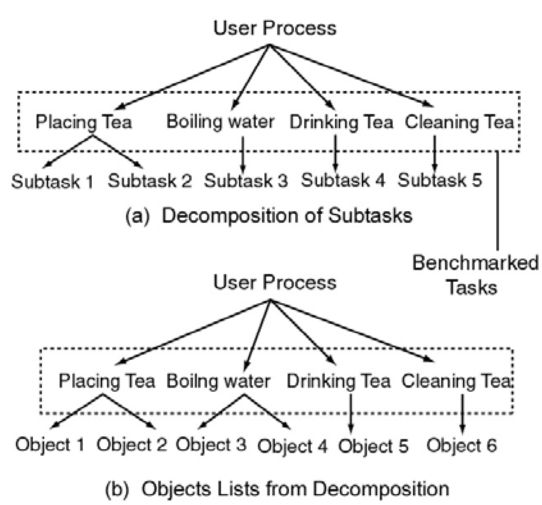


Figure 4: User Process Decomposition

Instead of describing the user process in details, we identify benchmarked tasks based on user observations. The benchmark tasks were decomposed into small subtasks (Figure 4a). In the same way, the objects used in each benchmark task were listed (Figure 4b). These benchmarked tasks will be used as a structure for queries both objects and processes.

The method has been implemented in the form of software to support both on-site and internet-based remote modes of user studies. The software has a database management system for storing images and video clips of typical processes. The procedure of the method is simple by assigning users to select objects related to what they use from the object library in figure 5a. The video clips related to those objects open automatically upon the object selection as show in figure 5b. Users need to describe their tasks step by step and describe the deviation of their process from the typical process.

The example in figure 5c presents a result of user knowledge findings from the tea case example. It explains the user's modification of existing products. A subject quickly makes fast boiling water by putting the warm water into the microwave instead of the cold water. This user knowledge can be integrated into the future product configuration.

5 USER KNOWLEDGE INTEGRATION

User knowledge cannot transfer directly to products. It needs to be interpreted as functions or user requirements as shown in figure 2b. The pattern of user knowledge represents a solution for the problem happening with the users and it can be transferred to other users in similar situations.

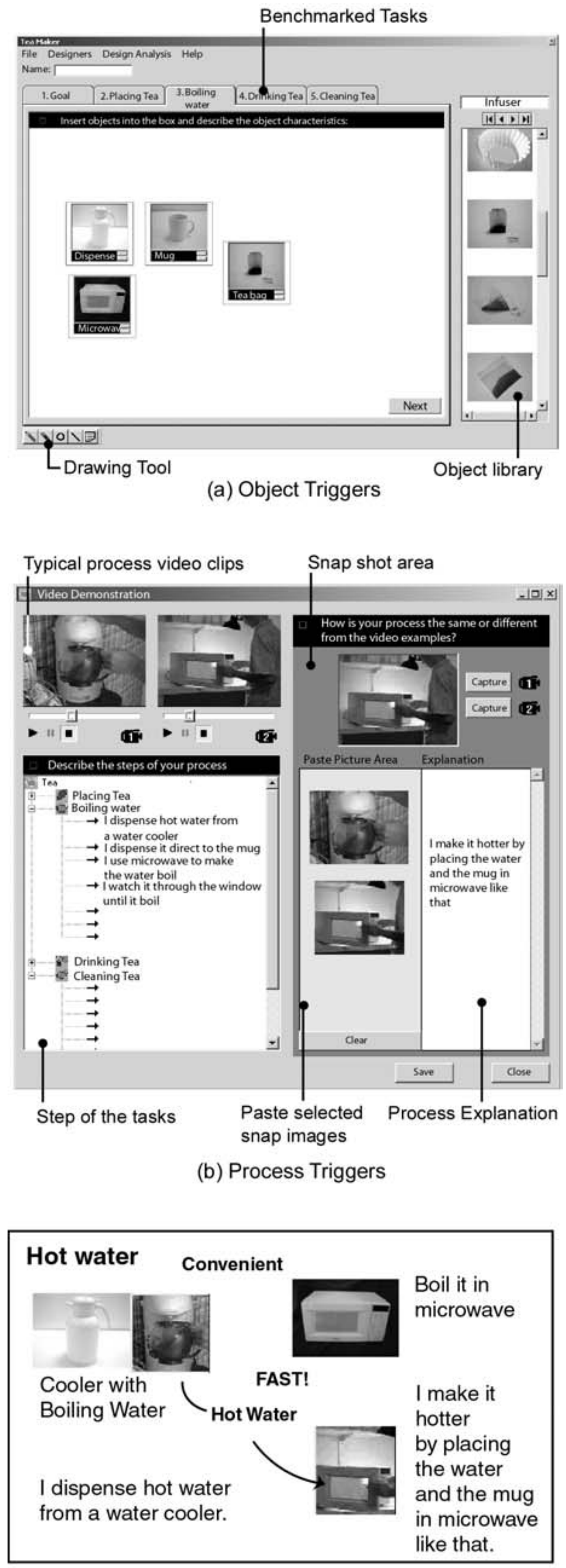
5.1 Patterns of User Knowledge

Patterns of User Knowledge are design documents. They are a combination of a solution from users and designers. The patterns represent user knowledge, user needs, context of use, and possible solutions.

The example in table 3 is a pattern named "Fast boiling water". From the result of OMUKE method, user knowledge was developed before the user needs were

Pattern Name	Fast Boiling Water	
User Knowledge	Microwave oven, warm water and mug for the boiling process make fast boiling water. <object><process of use><rationale>	
User Needs	Time of boiling water takes around 10 minutes. Many users are impatient to wait during this process.	
Context of Use	The user makes tea at school where there is a dispenser for hot water and a microwave oven. User always wants a fast boiling process during this stage.	
Possible solution	*Function: Make fast boiling water*	
	Product The hot tank can keep hot water in the tea maker for the next use.	*User Process* Users store water in the tank which can be kept for a week.

Table 3: Pattern of User Knowledge Format



(a) Object Triggers

(b) Process Triggers

(c) An example of what user made

Figure 5: OMUKE software

identified. The user knowledge form can be described as the object, process of use and the rational behind the use. The context of use includes the user experience and the environment of products. The product configuration is a possible solution that will be integrated into a new product while the user process represents the scenario of use that expects to have.

To accommodate the existing product and user process, the pattern of knowledge should be considered as a module or partial integration. Some user knowledge has the potential to create an entirely new product.

6 USER PROCESS BASED PRODUCT ARCHITECTURE - UPPA

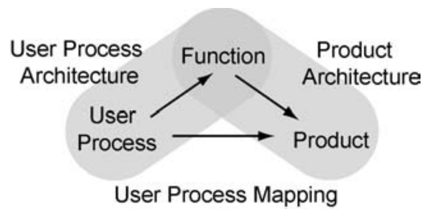


Figure 6: Relation between the user process and product architecture

The user process and product are dependent. And the user process cannot be described without the product. The User Process Architecture can be defined using the definition of the Product Architecture - the mapping of functions [10] to the physical components. The User Process Architecture is a mapping from the user's task to the process module. Figure 6 presents the relationship between three elements in the framework and the method to achieve the User Process Based Product Architecture - UPPA. User Process Architecture creates a flexibility of the process on the user side while the Product Architecture creates a flexibility of products on the product side. Both of them link with the functions. The mapping between two architectures is the User Process Mapping.

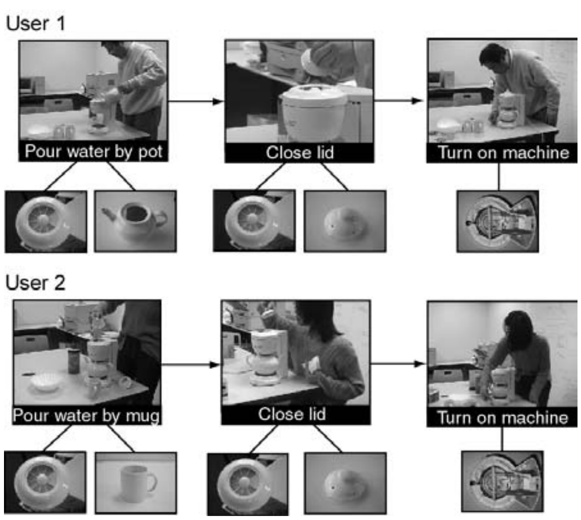


Figure 7: User Process Decomposition

6.1 Experimental mapping between the User Process and Product Architecture

The first experiment is the mapping of the product components according to the user process decomposition. An observation was conducted with two subjects using the given tea maker.

The user process was decomposed into subtasks and the product components were matched to each subtask as shown in figure 5. From the study, both users perceived functions of using the products in different ways. This can be seen clearly from the user's task. In the boiling water task, the first user carried water by using a pot while the second users had selected the mug. User's task and product functions presents a significant relationship since the product functions derived different subtasks.

6.2 User Process Architecture

The user process module collects many subtasks and it requires input and output to continue the next steps. The input of each module is considered as the operations and the object use while the output is the transformation of the process to achieve the outcome of what user expects or goal states.

User Function

User Functions represent what users interact when using the machine and they are at a higher level from the tasks that form many subtasks as can be seen in figure 8. These functions are used as links between the user process and the product components. User functions are equivalent to the system function-SF. When the user function transfers to the system function, additional product functions will be added to fulfill the product architecture.

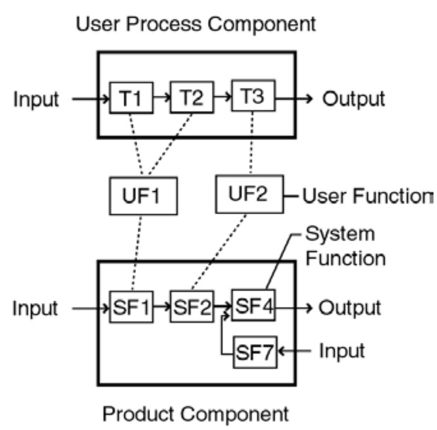


Figure 8: User function and Product components

User Process Component	User Function	Product Component
Carry water module	Operate water	Pot (Transfer water-SF)
Stove operation	Set boiler	Switch (Run current-SF)

Table 4: An example of linking by functions

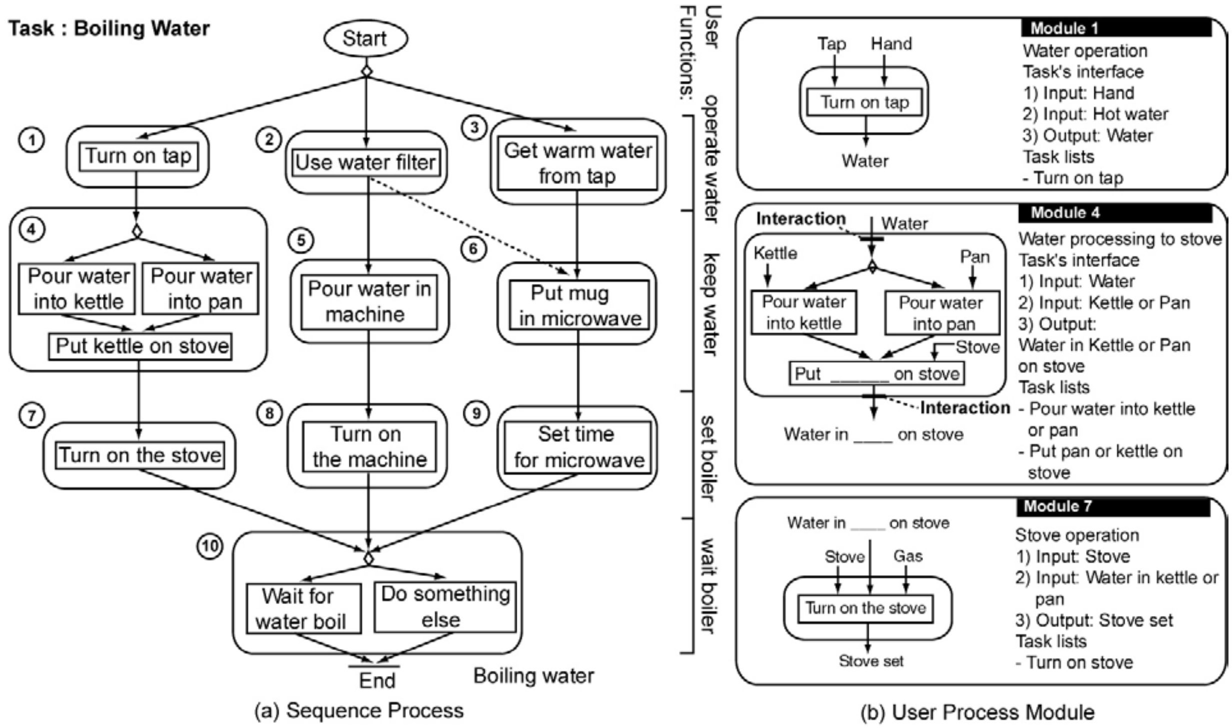


Figure 9: Module Classification

To illustrate a clearer picture, the diagram in figure 8 presents the task linear flow - T1, T2, T3, the user function - UF1, UF2, and system functions-SF1, SF2, SF4, SF7 in a new product component. UF1 is a user function forming the task T1 and T2 that links to the system function - SF1.

Table 4 is an example of user functions that links to the product components. When using the product for the first time, user perceives the product components and knows how them functions. Then users interpret the system function to the user function and perform the tasks to achieve the process of carrying water.

Module Classification

Examples of tasks from the task decomposition such as preparing tea leaves, boiling water, drinking tea drink and cleaning tea were collected. Figure 9a presents the three most common process alternative techniques of the

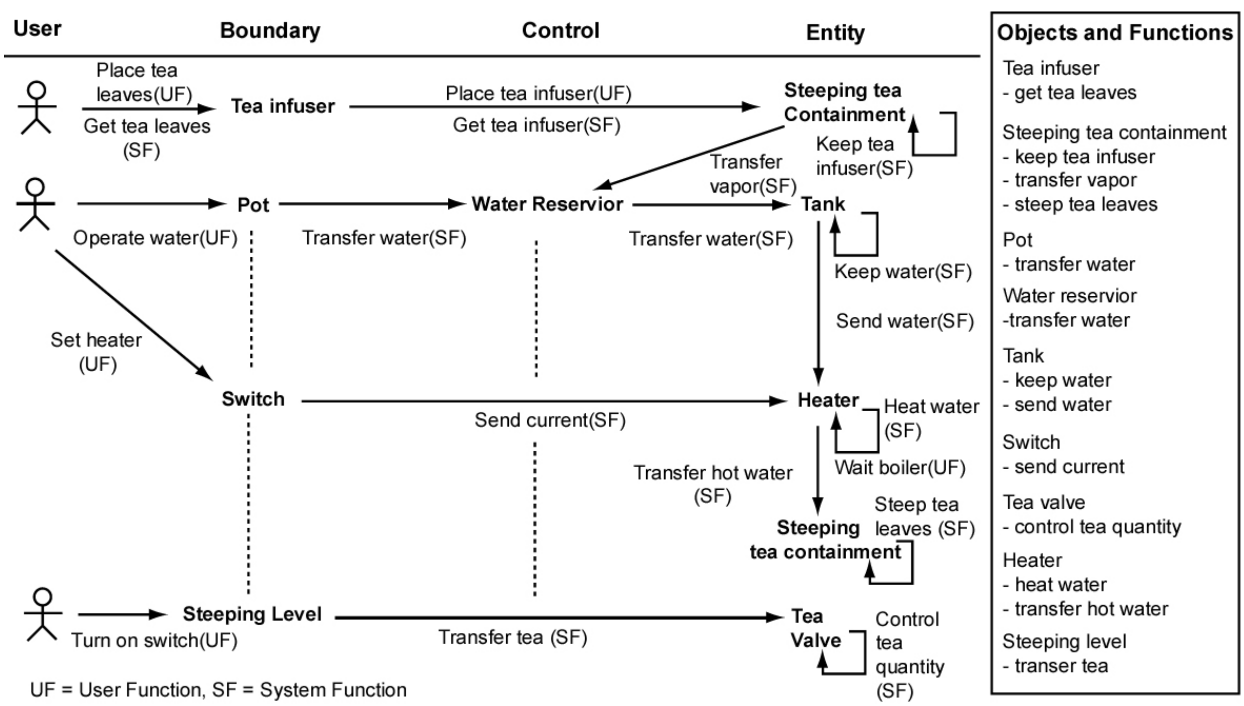


Figure 10: Function flows and Object discoveries

sequence processes such as boiling water with a microwave oven or a gas stove found in the user study. The vertical dimension refers to the same user functions that were used to divide the process module. The diagram represents 10 modules. There can be other alternative processes by moving one process to the other process such as the process indicated by a broken arrow between module two and six. In the boiling tea task, module 10 shared the same common process with other modules.

Module Interaction

Module interaction uses the principle from the theory of the technical system [3]. Each process module has an input, output and effects. The process interaction is the flow across the interface such as material, energy and information as shown in module 4 of figure 9b. Each module has input and output. For example, the output of the module 4 is the water in kettle or pan relating to the input of materials such as water, a pan and kettle.

6.3 Product Architecture

After establishing the user process architecture, the product architecture was built based on the system functions.

Object discoveries and Function Flow

The method in this research adapts the Robustness Analysis [7] used in software architecture. It is used to check that the system behavior is reasonable and finds objects to fulfill the system. The diagram in figure 10 presents the interaction between users and objects in the system. The user operation is a starting point of analysis as input. In this research, there are four channels that users can interact with the system. To find the objects, four important parts of the Robustness Analysis are defined: User, Boundary, Control and Entity. In terms of product design, the boundary is an interface object between users and the system. Control is a regulator of entity and a link between Boundary and Entity. The entity is an object execution of the tasks.

For example, if the user's goal is to boil the water, the tank is the object to execute the water process of keeping water. The pot used for carrying water is the interface object between users and system while the water reservoir is a link between the pot and tank to facilitate the water and vapor transfers. Water reservoir is discovered after this analysis.

After discovery new objects in the product system, Sequence Diagram [4] can be used to identify the functions in details that fit to the product components.

Behavior and Physical Structure

Based on the Function-Behavior-Structure model (FBS), the product architecture cannot directly map functions to the physical components without passing through the behavior structure. Other product types such as software can ignore the physical constraints – i.e., physical interface among modules. The behavior structure is considered as the connectors or interface between each module as shown in figure 11. The link indicates the behavior of components. Some of the components have a loosely coupling interface such as a pot. The physical structure in figure 12 is the final stage of the product architecture. The goal is to group the behavior structure together according to the design and manufacturing constraints. The grouping of the behavior structure

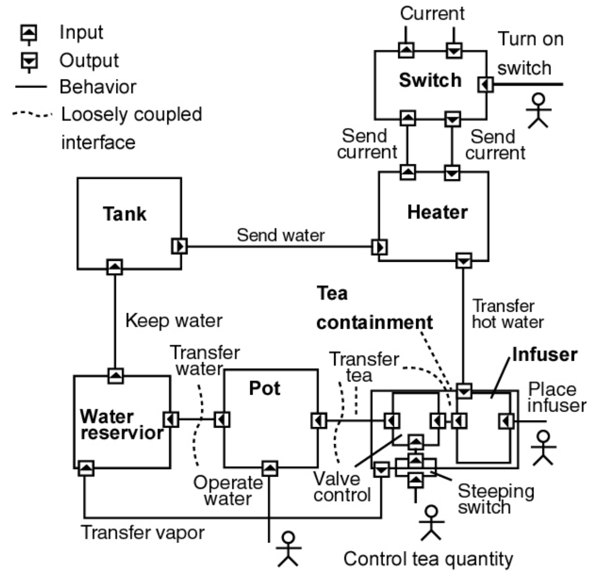


Figure 11: Behavior Structure

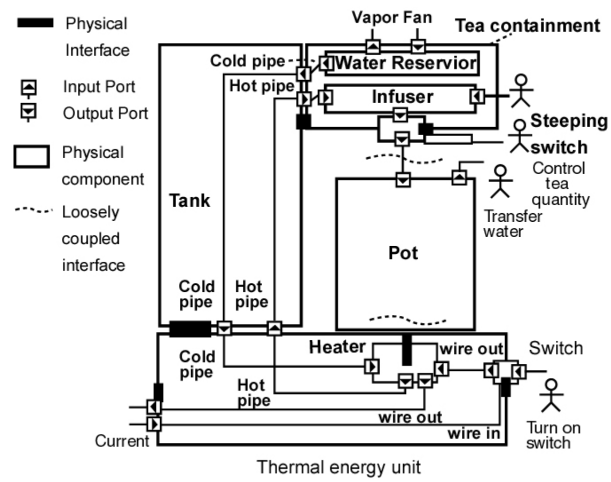


Figure12: Physical Structure - Tea maker side view

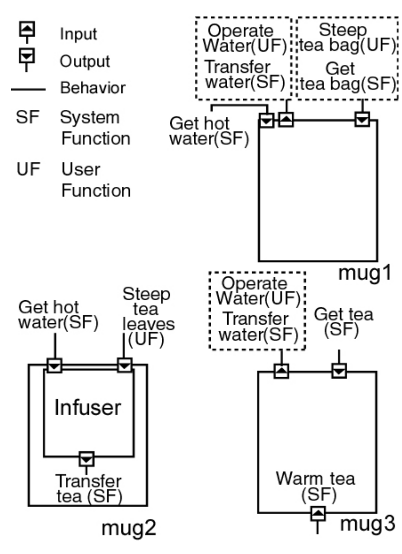


Figure 13: Alternative components based on behavior allocation - Mugs

creates a variety of physical modules. After building modules, the input and output port of each component should be defined. The physical interfaces connect all components and links all the behavior of the system beginning with the user operation or the system input. The allocation of physical components creates a variation of forms and physical structure.

The variation in behavior allocation creates different intents of product use. The example in figure 13 explains different uses of the mug based on the behavior allocation. If the user requires warming the tea drink, they need to select mug3 rather than mug1 or mug2. The module mug3 requires other components to link the behavior. In this case, the machine must have a heat pad or some mechanism to warm the tea drinks. The different behavior allocations represent not only the product structure but it reflects to the user behavior as well.

6.4 User Process Mapping

The user process modules shown in figure 14 were derived from the user tasks based on user knowledge and needs. The variation in combination with the user process modules describes the objects used in the process. A user process is composed of modules such as Np1, Np3, Np6. By swapping the process modules, alternative processes can be generated. This presents how the product architecture mapping integrates into the user process architecture.

Figure 14 shows a picture of the mapping between User Process Architecture and Product Architecture. Many user tasks are formed by the user functions which are equivalent to the system functions. System functions

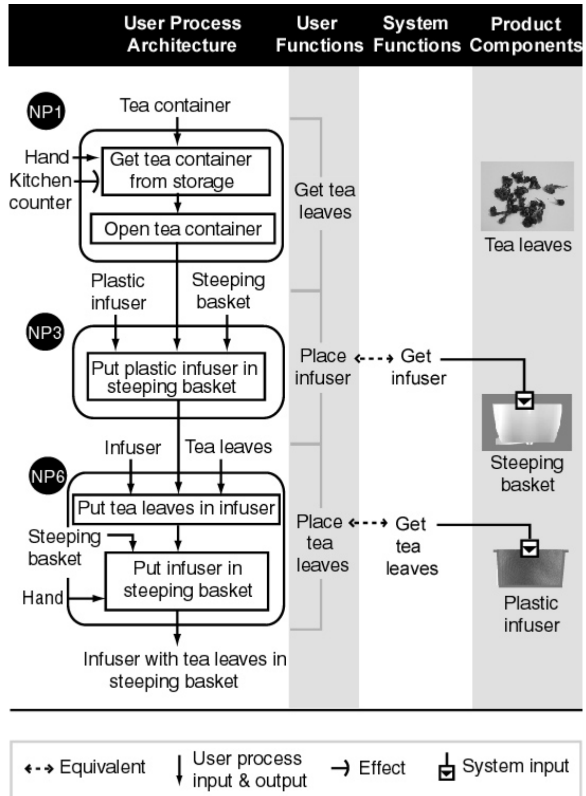
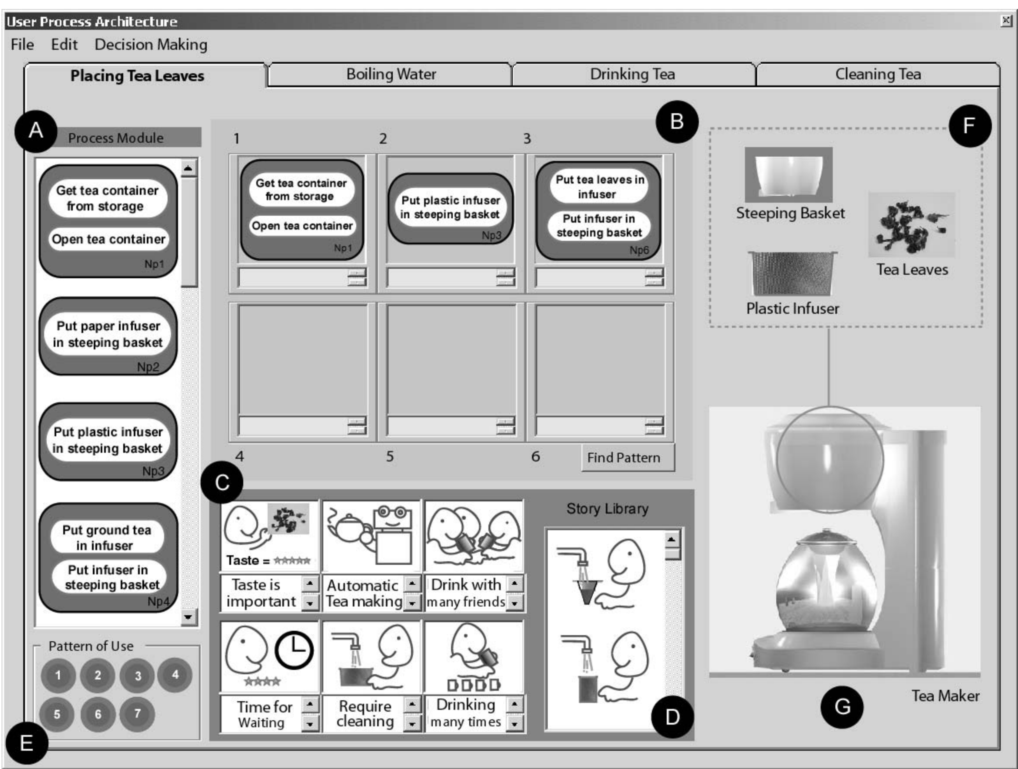


Figure 14: User Process Mapping in Placing Tea leaves



A) Process Module
A collection of the process modules in each task

B) Customized Process Module
Supporting customization and creating a new module

C) Process Attributes
Scenario built describes the process attributes and customizing user experience

D) Process Attributes Archive
A collection of iconic process attributes

E) Pattern of User Process Architecture
Process of use arranged from the design expert

F) Product Components
Components relate to the selected user process

G) Product Customization
Visualizing product from the selected user process architecture

Figure 15: Application Tool for UPPA

link different kinds of product components as one to one and one to many relationship but the user process module will limit only one product component by specifying the input and output of the process module. For example, ”the Get infuser” is a system function that links to different kinds of products components such as paper and plastic infuser. This makes a difficulty for structuring the end product architecture. Nevertheless, the process architecture limits the specific type of product input in the process module. As a result, only one product will be selected. This mechanism is used to avoid the conflict during the process of assemblying product architecture. The product architecture will have a completed structure after finishing all user process structure in all tasks, for example, the task begins from the preparing tea to cleaning tea.

The last element of product module that is not mention before is the process attributes since it requires an assembly of each process module to perceive it more clearly.

Process Attributes

Process attributes describe the quality and characteristics of process such as performance. These attributes are derived from the subjective and objective criterias. The subjective criterias were evaluated from the person’s judgement such as the taste of tea drink.

The objective criterias refer to the performance of the processes such as numbers of tea leaves use. Both of them can be used to describe the experience of use.

Attributes of the user process can be identified from a combination of process modules. For example, Np1, Np3 and Np6 link the object components such as tea leaves, plastic infuser and steeping basket. These objects represent the process attributes such as automatic tea making, reuse tea leaves and others

7 DISCUSSION AND CONCLUSION

To illustrate how the User Process Based Product Architecture - UPPA was applied to the practice, the concept can be implemented as a system for developers or end-users as shown in figure 15. The tool has three important parts. First, the tool can help in constructing the User Process Architecture based on the given process module. It also allows users or developers to customize and create their own process modules. Second, the system embeds the design and user knowledge to deliver a quality designed product. This knowledge has been collected as patterns of the User Process Architecture. To facilitate the communication and understanding of the user process, process attributes have been described as scenarios. Third, it has a mechanism to link with the Product Architecture. The product components will be presented after the User Process Architecture matching with the existing architectural patterns.

For the product developers, the process of the overall concepts are divided by user needs in two parts – Known and Unknown needs as shown in figure 14.

Known Needs

If developers find users having the same User Process Architecture as in the existing one, the process of use and the product components related to the user process will be presented. If developers cannot find related User Process Architecture, they or users are then able to create their own.

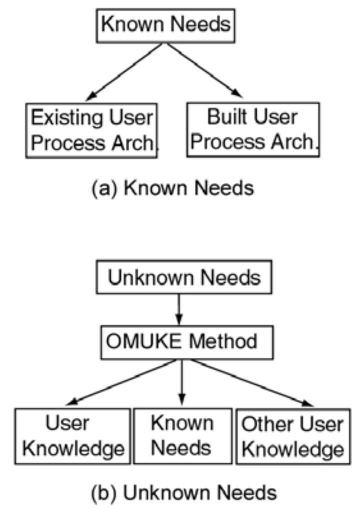


Figure 16: Process for Developer

Unknown needs

If developers do not know what a users need is, the OMUKE method will be used to elicit the user knowledge and user needs from the users. There are three strategic paths. First, if the new user knowledge was found, product developers might make a decision or not to integrate into the existing product system. Second, if the user needs were found, the next step would be to go back to the known needs criteria. Third, if users do not give any new user knowledge and identify their needs, other user knowledge will be accommodated to this point.

Comparison between User Based Product Architecture and other methods

In conclusion, the comparisons of architectural methods are described as follows:

1) Product Architecture by Otto [6]

Otto used the principle of Product Architecture by transferring user data into the function structure then mapping the functions to the physical structure.

User Requirements → Function Structure → Physical Structure

2) Unified Modeling Language in Software Architecture - UML [4]

Use Case Model is a primary concept to identify business modeling of requirements for users, actors and the system. Then using the Robustness Analysis as a preliminary design to link the analysis with Use Case and the design in the Sequence Diagram.

Use Case Model → Robustness Analysis → Sequence Diagram

3) User Process Based Product Architecture - UPPA

User Process Architecture embedded both existing and new user requirements. It links the Product architecture with user functions, and then identifies a new object based on the applied robustness analysis. In the end, the product architecture will be built based on the Function-Behavior-Physical Structure.

User Process Architecture → User Function → Applied Robustness Analysis & Function Flows → Product Architecture with Function-Behavior-Physical Structure.

Those three methods have the same principle of structuring the product architecture from the functions to product components. UPPA separates the User Process Architecture out from the Product Architecture since it creates an interchangeability of the User Process and Product architecture. UPPA does not directly build the Product Architecture. It is an indirect concept of building the product architecture according to the User Process Architecture.

From the experiment with the application tool, the strengths of the system provided from UPPA are:

- A vision of user experience is perceived clearly through the new product. It facilitates the perception of future products and its functions.
- It supports the building of User Process Architecture logically.
- Provides a new way of use as was discovered during the structuring user process module.
- User knowledge from many users is well promoted with UPPA since it supports the integration of user knowledge.

Weakness have been found in UPPA:

- It does not have direct control of the product architecture and it requires a transitive thinking between the user and system functions.
- It consumes time in structuring the user process architecture since it is based on logical thinking of selection and rational of process module combination.

To resolve problem the above, a mechanism for the links between the user process and the Product Architecture should be developed. This will reduce the transitive thinking between both architectures. UPPA is not only a mechanism to bring the right products to the right users but it combines with other elements such as user behavior, user knowledge and user needs. The methods of conducting those elements are important.

UPPA provides a new perspective of the product and product development process. It can be applied to other application domains such as service system. It would also be interesting to consider how UPPA could be applied to other process architectures in business and manufacturing views for mass customization. For example, the customization process of enterprise process architecture. This could fulfill the limitations of current Mass Customization capabilities.

8 ACKNOWLEDGEMENT

We would like to thank Christopher Barlow and Sharon Poggenpoh, Chujit Jeamsinkul, Pollawat Petavanich and Lynne Ferguson for their valuable comments on this paper.

9 REFERENCE

[1] Cook, A.M., 1996, Building Enterprise Information Architecture: Reengineering Information Systems, Prentice Hall, Upper Saddle River, 101-114.

[2] Hofmeister, C., Nord, R., Soni, Dilip, 2000, Applied Software Architecture, Adison Wesley Longman Inc, Massachusetts, 9-15.

[3] Hubka, V., Eder, W. E., 1988, Theory of Technical Systems Springer-Verlag, Berlin, Heidelberg, 36-57.

[4] Jacobson, I., Christerson, M., and Overgarrd, O., 1992, Object Oriented Software Engineering: A Use Case Driven Approach: Addison Wesley.

[5] Jiao, J., Tseng, M., 1998, Design for Mass Customization by Developing Product Family Architecture, GMRG thesis, 45.

[6] Otto K.N., Wood L.K., 2001, Product Design: Techniques in Reverse Engineering and New Product Development, Prentice Hall, Upper Saddle River, 175-176.

[7] Rosenberg, D., Scott, K., 1999, Use Case Driven Object Modeling with UML: A Practical Approach, Addison-Wesley Pub Co., 61-80.

[8] Soni, D., Nord L. R.,C. Hofmeister., 1995. “Software Architecture in Industrial Applications”, In Proceedings of the 17th International Conference on Software Engineering(ICSE’ 95), Los Alamitos, Calif.:IEEE Computer Society Press, 196-207.

[9] Shepard, A., 2000, Hierarchical Task Analysis Taylor & Francis, 7-25.

[10] Teeravarunyou, S., Sato, K., 2001, “Object Mediated User Knowledge Elicitation”, Asian Design Conference, Korea.

[11] Ulrich, T.K., Eppinger D. S., 1995, Product Design and Development, McGraw-Hill Inc: 131-132.

[12] Zachman, J.A., 1987, “A Framework for Information System Architecture”, IBM Systems Journal, 26/3.

Authors’ contact information
Prof. Keiichi Sato – sato@id.iit.edu
Mr. Sakol Teeravarunyou - sakol@id.iit.edu