INTERNATIONAL DESIGN CONFERENCE - DESIGN 2006
Dubrovnik - Croatia, May 15 - 18, 2006.

DESIGN 2006

SERVICE ENGINEERING: A NOVEL ENGINEERING DISCIPLINE FOR HIGH ADDED VALUE CREATION

Y. Shimomura, T. Sakao, E. Sundin and M. Lindahl

Keywords: value, service, service engineering, design methodology, computer aided design

1. Introduction

Service is attracting more and more attention as manufacturing industries are shifting from a "product seller" toward a "service provider." To design and develop services effectively and efficiently, a methodology of service design and a support by the computer system based on the methodology are needed. However, very few researchers have so far dealt with services from the viewpoint of engineering design, while services have often been traditionally a topic in the field of marketing and management (e.g. [Shostack 1981]).

The authors are carrying out fundamental research in "service engineering," which deals with services in an engineering manner [Tomiyama 2001]. A methodology of service modeling has been already proposed [Shimomura 2002]. Based on these, this paper aims at proposing a "service CAD," a computerized tool to support service design, which provides designers with a design environment equipped with knowledge about existing services.

The rest of the paper consists of the following sections. Section 2 discusses limitations of current product centered manufacturing. Section 3 proposes the concepts of a service CAD. Section 4 reviews the service modeling method which has been already proposed. Section 5 describes the specifications of a prototype service CAD, and its implementation with an example service. Section 6 gives discussions and Section 7 concludes the paper.

2. Limitations of Current Product Centered Manufacturing

A number of methodologies and tools for high functional products design have been developed and a lot of high functional products are available on the market as production industries have got more high technology conscious. However, very few of the products so far achieved success in the market accepted by consumers. Producers still stick to their own views on customer's value in developing their products although they incorporated advanced technologies successfully. In other words, they deal with activity of incorporating advanced technologies as the royal road to designing physical structure for a requested functionality. This is among the reasons why present high functional products in many cases do not accelerate sustainable market growth.

Producers should be more careful of how to carry out business activities which realizes sustainable market growth. To do so, the concept of value is among the crucial pieces, because value is the interface between consumers and producers. This concept in the high added value creation context is better explained using a metaphor in Figure 1. The point to be argued is too much focus on physical products without touching services provided together. In Figure 1, the plate is a design object, while two bars fit through the holes of the plate represent axes of product and service, which are regarded as important factors constituting a design object. The figure shows the same effect as when one pull out a drawer unevenly. The holes have a small space to play around the bars. If only a product or service is designed, the total value is not increased considerably (Operation 1 and 2, respectively). By designing a set of product and service, the total value gets higher effectively. It is true that many methods dealing with service in management area exist (e.g. Weber 2004). However, very few have tackled so far establishing a method and tool that pursue providing a high value for customers by designing services in parallel with designing products. It is also important to pay attention to the general trend that services instead of physical products get more wanted as economies mature (e.g. Tomiyama 2001). However, it is obvious that value is a concept varying from one consumer to another. Thus, consumer modeling utilized for the following design activities is automatically crucial. This will be a missing piece, since most of current mass production deals with a flat recognition on various consumers.

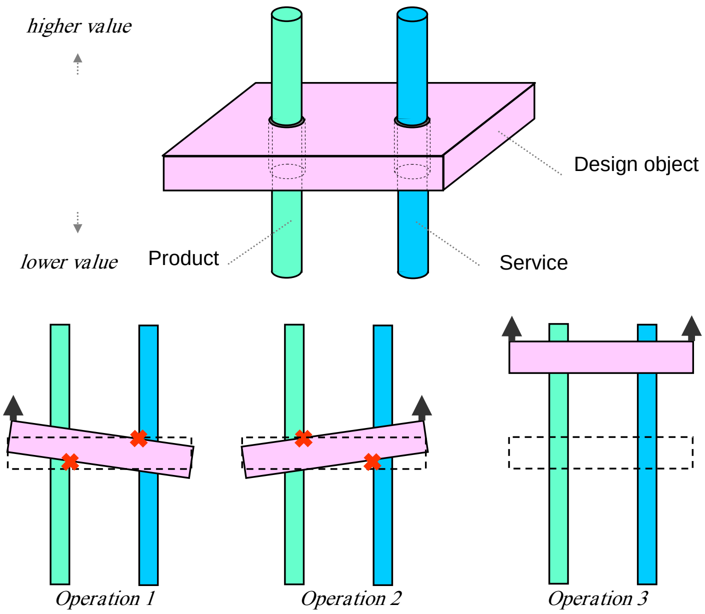

Figure 1. Design operations with constraint between product and service

3. Service Design and a Service CAD

In service engineering, service is defined as an action that a provider performs, through which a receiver changes into another state the receiver desires (Figure 2). An analysis of existing designs of services revealed that most of the service designs can be classified into the following three patterns [Shimomura 2002]:

1. (Re-)design of a new service by enhancing components of and improving existing services,
2. application of existing service to a different field, and
3. creative new design.

For the first two classes of service design, the success or failure, the quality, and the efficiency of service design depend to a great extent on the knowledge about service design and existing service cases. However, systematized knowledge about service design hardly exists, while in contrast in mechanical design existing design knowledge can be stored in a reusable form.

Regarding the first pattern of service design, we could observe at least the following three operation patterns [Shimomura 2002]:

(1-1) Substitution of components with something else,
(1-2) removing a part of service, and
(1-3) combination of different existing services.

Pattern (1-1) is an operation to substitute a component of an existing service with another one. Patterns (1-1) and (1-2) are operations to build a new service by changing and modifying the whole or a partial structure of the target service, while Pattern (1-3) comes up with a new combination of services.

Based on the above-mentioned analysis, we have proposed the concepts of service CAD (Computer Aided Design) [Shimomura 2002] to support engineers in designing services. The service CAD serves as an environment to develop a service by providing the knowledge about existing service cases and various operation rules stored in its database. Figure 3 shows the conceptual scheme of the service CAD, which consists of the followings.

1. Service case base: A database of existing service cases.
2. Design rule base: A database of operation rules for service design.
3. Reasoning engines: Reasoning engines reason about various properties of service such as similarity. A pluggable mechanism is employed so that a necessary reasoning engine is selected based on the designer's request.
4. Service evaluator: An evaluation module to evaluate a service design solution.
5. Service design process organizer: A module to support service design processes based on a specific design methodology by means of calling other components etc.

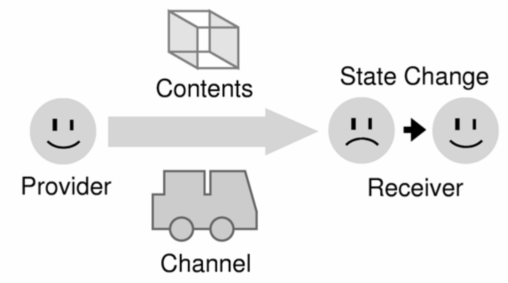

Figure 2. Definition of service

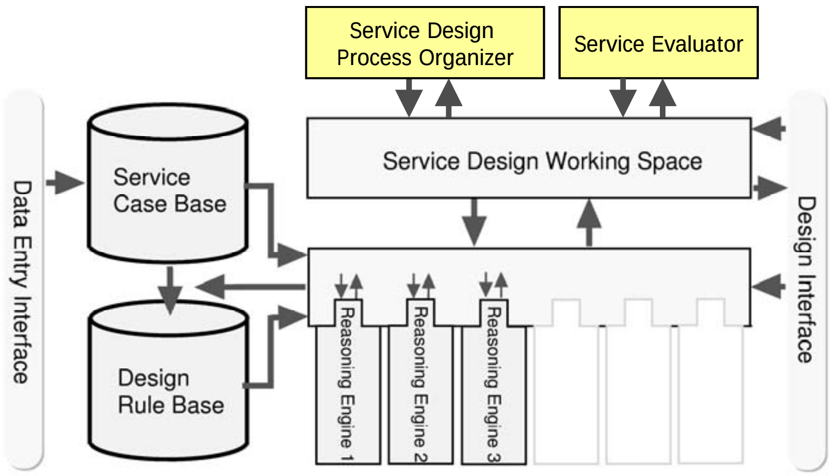

Figure 3. The conceptual scheme of service CAD

4. The Employed Service Model

In order to develop the service CAD, it is necessary to model a service in a form that can be represented in computer software. A service consists of the following components based on the definition of the service in Section 3 (Figure 2) [Shimomura 2002].

1. Provider: provides the service.
2. Receiver: receives the service.
3. Contents: cause the receiver's state change directly when provided by the provider through the service channel.
4. Channel: what contributes to the receiver's state change indirectly by transmitting, supplying or amplifying the service contents.

In addition through CoP. These parameters are introduced to represent subjective nature of service; even for an identical service, there are recognition differences between service receivers.

In many cases, supplying services requires an environment with a complex structure, in which a provider, a receiver, and several intermediate agents interact each other. A service can involve a chain of subservices. A flow model is a sub model of service and represents those multiple complex structure as a chain of agents who provide/receive a service (Figure 4) [Shimomura 2003]. The flow model includes descriptions of a service in terms of a target range which are called scope models. The value and the cost of a service are expressed as the total changes of RSPs contained in the scope model (Figure 4).

A view model represents a concrete method to realize the change of an RSP of a receiver (Figure 4). A view model describes functions of the channels and the contents which realizes a change of an RSP. This functional information includes a function name (FN), a function parameter (FP) that is the main parameter of the function, and a function influence (FI) that is the main influence to FP. Figure 5 depicts one of the view models of a cafe service as a scope model between a coffee shop owner and its customers. This view model expresses one of the RSP, "workspace available," of the customers who use the cafes as a space for their own business.

In Figure 5, FPs ("Extent of a Workspace," "Extent of a Plane," "Comfort") connected to the RSP directly are CoPs, and FPs "Width of a Space", "Depth of a Space," "Height of a Space," "Shape of a Desk," "Hardness of Desk Top," "Angle of Desk Top," "Height of a Chair," "Shape of a Chair," "Softness of a Chair," and "Angle of a Chair") connected to the RSP indirectly through other FPs are ChPs.

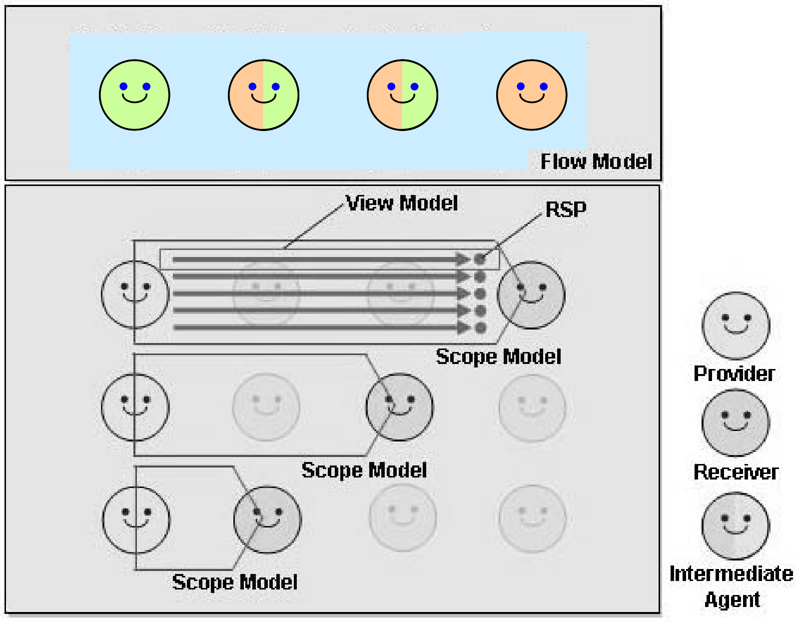

Figure 4. Relationships among flow model, scope model, view model and RSP

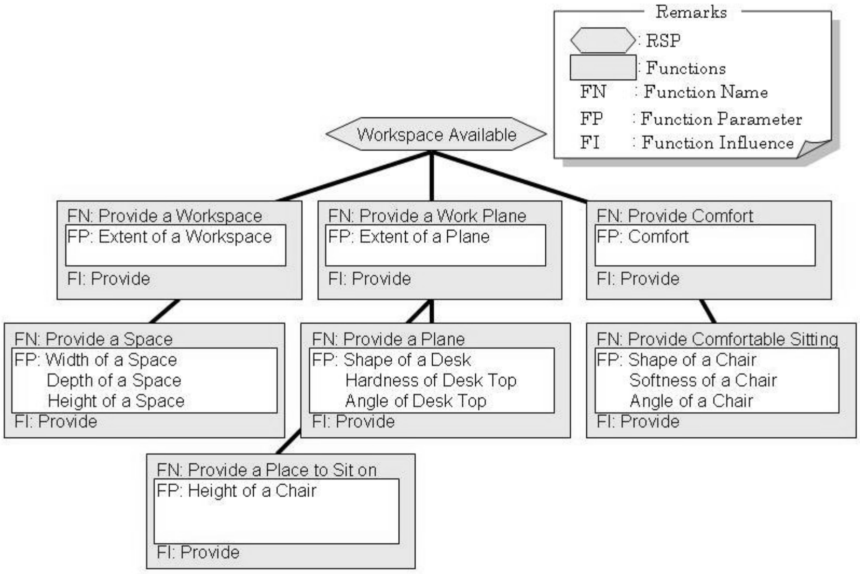

Figure 5. An example of a view model

5. The Prototype of a Service CAD

The prototype system of a service CAD, which is called "Service Explorer," has been developed. It is necessary that Service Explorer has the following five functions;

1. To allow a user to input and edit a service model,
2. to display component elements that designers focus on,
3. to register service cases in a service database,
4. to search in the service database, and
5. to reuse a service model data stored in the service database.

Required specifications and implemented specifications in details are as follows.

1. To allow a user to input and edit a service model
2. This is the most fundamental function of Service Explorer. In order to acquire the knowledge of service cases for service design efficiently, an easy graphical interface to describe a service model is needed. A service model is described as a graph structure consisting of nodes and arcs.
3. To display component elements that designers focus on
4. The system has to be able to (or not to) display component elements selectively depending on designers' demands so that they can understand the structure of the service efficiently. To do so, Service Explorer provides a function to display the function topology and the parameter structure.
5. To register service models in the service database
6. It is desirable that the service database can store service cases independently of the specific OS/application. For this purpose, Service Explorer employs XML as the database description language.
7. To search in the service database
8. Service Explorer is equipped with a search function to look up the service database depending on designers' requests. On the current Service Explorer, designers can search for service models with keywords contained in the composition elements (RSP, FN, FP, FI, and Entity).
9. To reuse a service model data stored in the service database
10. This is the function to reuse composition elements or structure of a service model stored in the service database, when designers inputs and edits a service model.

5.1 The Implementation of the Prototype

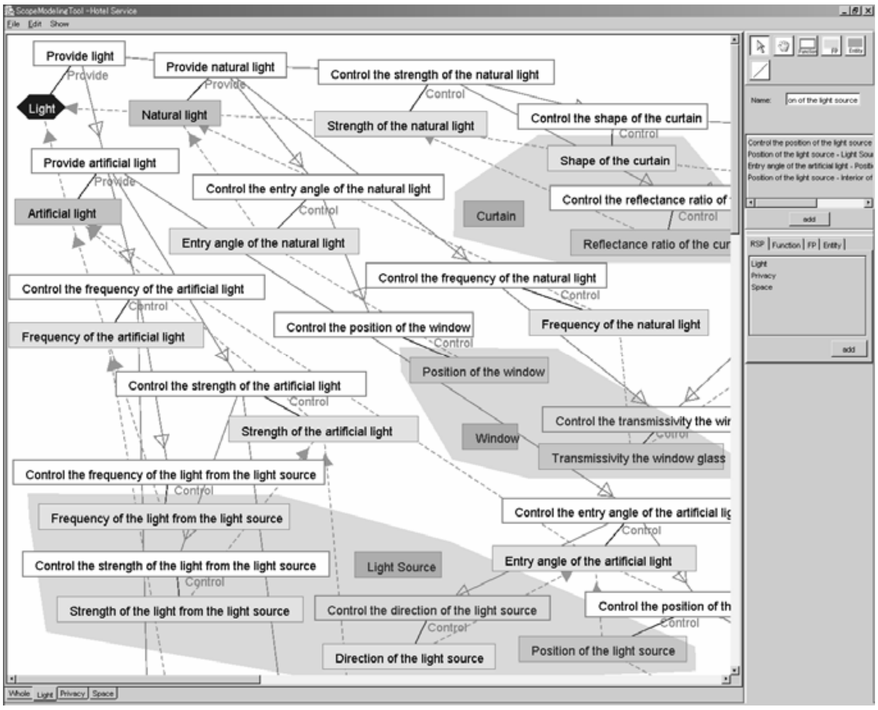

Figure 6. A screenshot of a flow model for a hotel

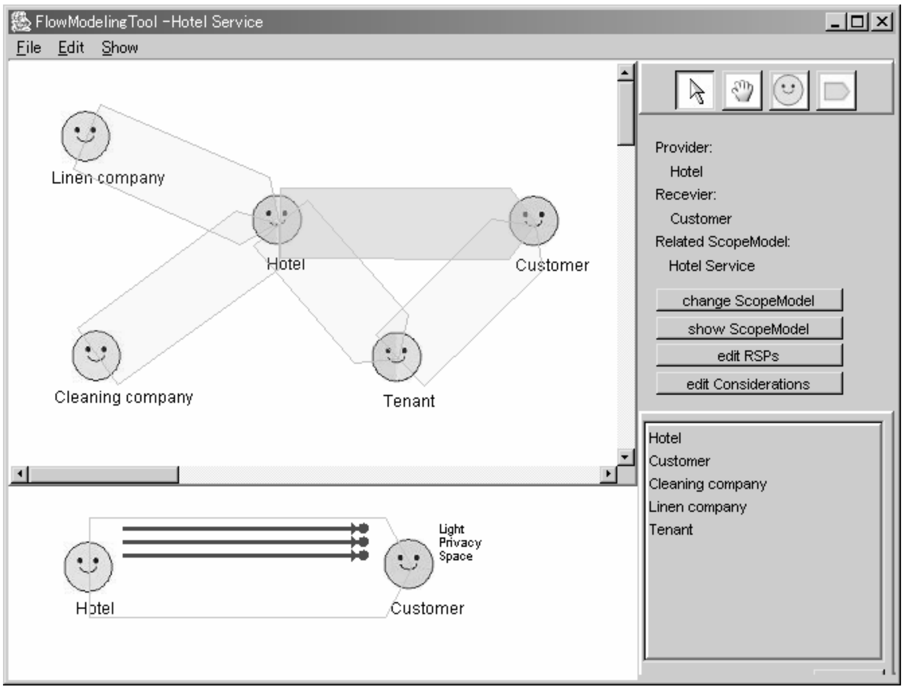

Figure 7. A screenshot of a view model for “light”

Based on the above-mentioned functional specifications, Service Explorer was developed in Java SDK 1.4.1 and XML version 1.0. The MVC model [Krasner 1988], which has been used widely in general GUI applications, was adopted as basic architecture of Service Explorer. By applying the MVC model, high flexibility and reusability of Service Explorer, and robustness of the service model data are achieved.

Below how Service Explorer works is explained using screenshots of a design case of hotel service. Figure 6 shows the user interface to input and edit a flow model, which expresses a service as a chain of agents, depicting a design case of hotel service. Concepts such as “Hotel,” “Customer,” “Linen company,” “Clean company,” and “Tenant” are arranged as agents which participate in the service. Multiple scope models which represent provision/receipt relationships among them are defined in this flow model. Furthermore, the interface shown on Figure 6 allows designers to set up RSPs contained in the scope model.

Figure 7 shows the user interface to input and edit a view model for each RSP in the scope model defined on the flow model. Figure 7 depicts a view model about “Light” RSP in the scope model between a hotel and a customer. The RSP is closely related to the customer’s demand on brightness. In the view model, “Light” is described as a root element and “Provide Light” is described as a root function to change the RSP. Then, the function is detailed into “Provide artificial light” and “Provide natural light”. By detailing gradually the functional structure which realizes the change of the RSP, the RSP is finally linked to entities such as "Light source," "Curtain," and “Window." A view model in the prototype is represented by the graph structure consisting of the following elements.

1. RSP nodes
   An RSP is represented by a hexagon node. The color of the node expresses whether it is value or cost. Since “Light” RSP in Figure 7 is a value for the customer, the node is painted blue.
2. FN nodes
   An FN (e.g. “Provide natural light”) is represented by a square and white node.
3. FP nodes
   An FP which influences the RSP directly (i.e. CoP, e.g. “Natural light”) is represented by an orange square node, while an FP which influences an RSP indirectly through CoPs (i.e. through ChP such as “Strength of the natural light”) is represented by a yellow square node.
4. Arcs between FN nodes
   A relationship between functions is represented by an arc between FN nodes. For example, the arc between “Provide light” node and “Provide artificial light” node means that one is developed into the other in Figure 7.
5. Arcs between FN nodes and FP nodes
   An FI is represented by an arc between an FN node and an FP node. For example, the arc between “Provide natural light” node and “Natural light” node means “Provide” FI in Figure 7.
6. Arcs between FP nodes
   A relationship between FPs is represented by an arc between FP nodes. This relationship is generally defined as a result of the relationship between functions or the effect of the embodiment structure. The designer can set a causal relation on the relationship expressed by a directed arc. For example, the arc between “Strength of the artificial light” node and “Strength of the light from the light source” node means that one affects the other in Figure 7.
7. Entity topology
   An entity which has FPs as attributes is represented by a gray polygon. For example, a curtain is represented by the gray polygon which contains “Curtain” node, “Shape of the curtain” node, and “Reflectance ratio of the curtain” node in Figure 7.

Figure 8 shows the user interface to see the whole realization structure for the RSPs contained in a scope model. This gives a scope model obtained by integrating each view model for “Light,” “Privacy,” and “Space” RSP of the customer. Here, FN nodes and FI arcs are hidden from the screen intentionally in order to focus on the relations among parameters. The scope model shows that “Curtain” is indirectly linked to both “Light” and “Privacy” RSPs. That is, it affects the value of both “Light” and “Privacy.”

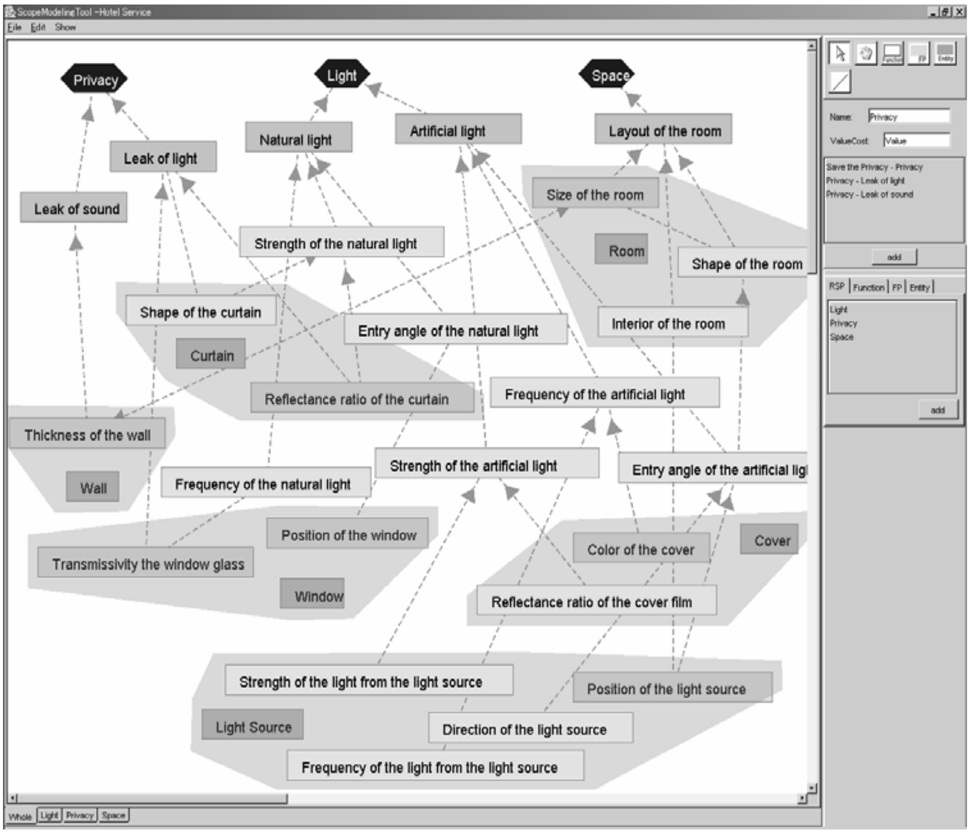

Figure 8. A screenshot of a scope model for a hotel

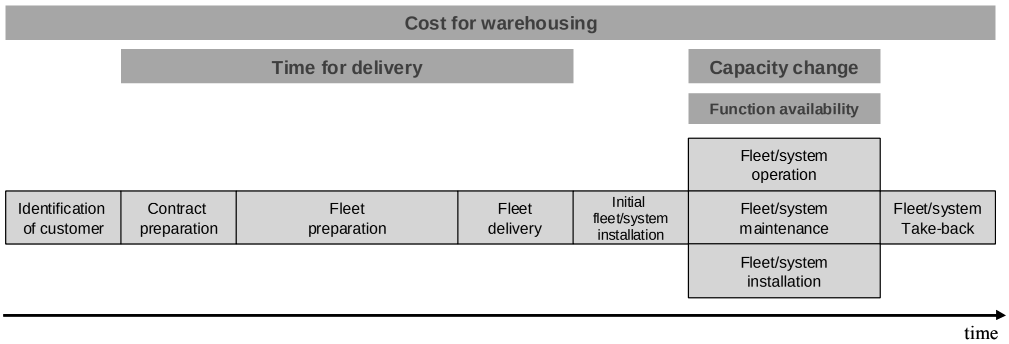

Figure 9. A time line model describing how the RSPs (cost for warehousing, time for delivery, capacity change and function availability) influence the warehouse service [Lindahl 2005]

Another example from the warehouse providing business shows how a manufacturer provide the service of warehousing to its customer instead of only the physical product (forklift truck). Figure 9 shows the different RSP's for the warehousing service. In order to understand how they are related, the RSP's are shown in a typical time line for the provided service.

When analysing the model with Service Explorer it was clear that the most important RSP was "function availability". This means that it is most crucial for the service providing company to focus on having a high function availability in the product-service offers. This can successfully be achieved by a proper combination of products and services as shown in Figure 1. The modelling of the warehouse service is descibed in more detail in [Lindahl 2005].

6. Discussion

6.1 Verifications of the Prototype

This section verifies the proposed methodology for service modeling and the capability to describe services of Service Explorer we have developed. It revealed that designers are able to express complex and multiple structures of services on Service Explorer as it was demonstrated by the flow model of hotel service in Figure 6. This information helps designers with an analysis about trade-offs among several agents. The examples in Figures 7 and 8 have proved that the view model on Service Explorer is capable of describing abstract needs of service receivers with functional structures including concrete entities to fulfill those needs effectively. In addition, the scope model is able to express multiple relations between receivers' needs and entities. For instance, "Curtain" in Figure 8 affects multiple RSPs, "Privacy" and "Light." This type of structure can be commonly seen in services.

However, the scope model in Figure 8 has no RSP related to a receiver's cost such as "Accommodations fee" and "Cost for access." Service Explorer should be extended to describe cost by showing several view models for cost RSPs in the near future. Moreover, the example revealed that it is not an easy task to set up RSPs. For this reason, we might have to develop a methodology to easily set up a set of RSPs and to support the process with Service Explorer.

Service Explorer is different from existing conceptual design support tools for mechanical products (for example, [Umeda 1996]) in that it can describe the functional structure to realize the receivers' needs. The abstract needs of service receivers and the relations between those needs and associated functions have not been represented in traditional support tools for mechanical product design. In contrast, Service Explorer does not have the capability to describe the detailed functional structures and the detailed embodiment structures of services, which are possible with some tools for product design. Extending Service Explorer to deal with connection with concrete product structures can also be a future research issue.

6.2 Relations with Other Modules

This section discusses the relationships among the current Service Explorer and other components explained in Section 3.

Firstly, it is possible for Service Explorer to support designers in designing services more efficiently by utilizing a design process support module ("Service Design Process Organizer" in Figure 3) based on the design methodology proposed in [Sakao 2004].

Secondly, an evaluation module based on the service evaluation method discussed in [Sakao 2003] can be added to Service Explorer to evaluate the design solutions in a service design process, and to present the information for improving the service to the designer.

Thirdly, with regard to the reasoning engines, it might be interesting to apply various reasoning mechanisms provided by a general-purpose reasoning platform called UAS (Universal Abduction Studio) [Takeda 2003]. UAS, which focuses on abduction as a reasoning form to use the multiple and different domain knowledge in a cross-sectional manner, is a reasoning environment to support a highly creative design by various types of abductive reasoning methods. Studies have been made on the reasoning mechanisms which focus on the similarity based on the definition of the ontology or the topological structure of the knowledge so far. Service Explorer can allow designers to design more creative services, by having these abducting reasoning engines and applying them to the service case knowledge in the service database.

7. Conclusions

This paper proposed a service CAD, which is a tool for supporting design and development of services, and demonstrated its effectiveness from an engineering viewpoint. To do so, a prototype of service CAD named Service Explorer was implemented, and an example service was illustrated for demonstration.

Our future work includes applying more service cases in industries so that Service Explorer can be further verified whether it has enough capability and usability as a service design tool. Additionally, we can also include the followings in our future research topics.

- To develop a method to support setting up RSPs.
- To verify the effectiveness of Service Explorer as a creative design support environment by means of implementing other modules.

Acknowledgement

This research was partially supported by a Research Fellowship Programme by the Alexander von Humboldt Foundation in Germany.

References

Krasner, G. E. and Pope, S. T., "A cookbook for using the model-view-controller user interface paradigm in Smalltalk-80," Journal of Object-Oriented Programing, 1(3):26-49, August/September 1988.

Lindahl M., Sundin E., Sakao T. and Shimomura Y. "An Application of a Service Design Tool at a Global Warehouse Provider", Proceedings of International Conference on Engineering Design (ICED-05), Melbourne, August 15-18, 2005, CD-ROM.

Sakao, T. and Shimomura, Y., "A Method and a Computerized Tool for Service Design," in Proceedings of the International Design Conference – DESIGN 2004, Dubrovnik, 2004, pp. 497-502.

Sakao, T., Watanabe K. and Shimomura Y., "A Method to Support Environmentally Conscious Service Design Using Quality Function Deployment (QFD)," in Proceedings of the Third International Symposium of Environmentally Conscious Design and Inverse Manufacturing (Eco Design 2003), Tokyo, IEEE Computer Society, 2003, pp.567-574.

Shimomura, Y., Tomiyama, T., "Service Modeling for Service Engineering," in Proceedings of The 5th Intl. Conf. on Design of Information Infrastructure Systems for Manufacturing 2002, 2002, pp. 309-316.

Shimomura, Y., Watanabe, K., Arai, T., Sakao, T. and Tomiyama, T., "A Proposal for Service Modeling," in Proceeding of the Third International Symposium on Environmentally Conscious Design and Inverse Manufacturing (Eco Design 2003), Tokyo, IEEE Computer Society, 2003, pp.75-80.

Shostack, G.L., "How to Design a Service," in Donnelly, J.H. and W.R. George (eds.), American Marketing Association, 1981, pp.221-229.

Takeda, H., Sakai, H., Nomaguchi, Y., Yoshioka, M., Shimomura, Y. and Tomiyama, T., "Universal Abduction Studio -Proposal of A Design Support Environment For Creative Thinking In Design-," in Proceedings of the 14th International Conference on Engineering Design -ICED03-, Stockholm, Sweden, 2003, CD-ROM.

Tomiyama, T., "Service Engineering to Intensify Service Contents in Product Life Cycles," in Proceedings of the Second International Symposium on Environmentally Conscious Design and Inverse Manufacturing (EcoDesign 2001), IEEE Computer Society, 2001, pp. 613-618.

Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y. and Tomiyama, T., "Supporting Conceptual Design based on the Function-Behavior-State Modeler," Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 10, No. 4, 1996, pp. 275-288.

Weber, C., Steinbach, M., Botta, C., Deubel, T., "Modeling of product-service systems (PSS) based on the PDD approach," in Proceedings of the International Design Conference – DESIGN 2004, Dubrovnik, 2004, pp. 547-554.

Yoshiki Shimomura, Ph.D., Professor
Department of System Design / Tokyo Metropolitan University
Minamiosawa 1-1, Hachioji-shi, Tokyo 192-0397, JAPAN
Tel.: +81-(0)426-77-2729
Fax.: +81-(0)426-77-2729
Email: yoshiki-shimomura@center.tmu.ac.jp
URL: http://www.comp.metro-u.ac.jp/smmlab