Design Method for Concurrent PSS Development

K. Kimita[1], F. Akasaka[1], S. Hosono[2], Y. Shimomura[1]
[1.] Department of System Design, Tokyo Metropolitan University, Asahigaoka 6-6, Hino-shi, Tokyo, Japan
kimita-kouji@sd.tmu.ac.jp, akasaka-fumiya@sd.tmu.ac.jp, yoshiki-shimomura@center.tmu.ac.jp
[2.] Service Platforms Research Laboratories, NEC Corporation, Shibaura 2-11-5, Minato-ku, Tokyo, Japan
s-hosono@bu.jp.nec.com

Abstract

In manufacturing, Product-Service Systems (PSSs) that offer products in combination with services have attracted considerable attention. However, due to the characteristics of PSSs, its development is more complex than that of a product. This paper proposes a design method that allows PSS designers to address conflicts in the development phase. To do so, first, we adopt the approach of axiomatic design to detect and avoid the conflicts. Second, for the representation of PSS structure, the modeling methods from Service Engineering are applied. The effectiveness of the method is demonstrated by applying it to a practical case-study.

Keywords

Requirement extraction, Design support, Resource management, Axiomatic design

1 INTRODUCTION

Environmental problems have grown in importance over the last couple of decades. Consequently, society should reduce the production and consumption volumes of artifacts to an adequate, manageable size without decreasing the current quality of life. For the purpose, it would be effective to pursue qualitative satisfaction rather than quantitative sufficiency, and thus, decouple economic growth from material and energy consumption [1]. For this purpose, manufacturing companies are starting to recognize that services and knowledge provided through a product are more important than the product itself [2]. As a result, "Product-Service Systems" (PSSs) [3-5], which create value by coupling a physical product and a service, have been attracting attention.

In order to achieve a successful PSS, the stakeholders are required to extend their responsibility in the life cycle [5]. This is because, from the viewpoint of environmental issues, providers need to establish proper organization for the management of product life cycle, such as reuse, remanufacturing, and recycling. They also need to educate receivers for efficient use and proper disposal of products. With respect to the value creation process, value is always determined by receivers [2]; providers, therefore, need to construct systems for the observation of receivers' needs and to establish networks to share relevant information. To compensate these extended responsibilities, new and varying types of stakeholders must be involved [5]. However, this situation may cause incompatibilities among their objectives and tasks and will inevitably induce conflicts. Namely, tasks aiming to achieve the objectives of a particular stakeholder may preclude the achieving the objectives of the others. The execution of a PSS containing excessive conflicts results in falling performance [6].

This paper, therefore, proposes a design method that allows PSS designers to address such conflicts in the development phase. The effectiveness of this method is demonstrated by its application to the example of an e-learning service.

2 OBJECTIVES OF THIS STUDY

2.1 Scope of this study

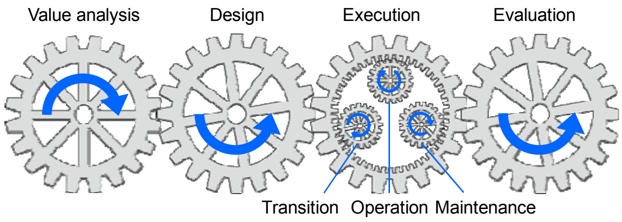

Figure 1: Life cycle of Product-Service Systems.

In this study, the life cycle of a PSS is defined as shown in Figure 1. The life cycle is comprised of four phases: value analysis, design, execution and evaluation. In the value analysis phase, the goals of a PSS, such as requirements and objectives of receivers, are first extracted, and the realization structures for the goals are then designed in the design phase. In the following phase, the designed PSS is executed; note that the execution phase is subdivided into three phases: transition, operation, and maintenance. In the transition phase, both providers and receivers make preparations for the operation phase; these preparations include not only manufacturing and installing physical products but also obtaining knowledge and skills required in the operation phase. During the operation phase, maintenance is carried out as necessary for tangible products, as well as intangible resources, such as employees. Finally, the designed and executed PSS is evaluated from the viewpoints of both providers and receivers.

2.2 Problems in the development of Product-Service Systems

According to the PSS life cycle, PSS development is carried out through the value analysis and design phases, and therefore, in these phases, designers need to consider the conflicts that may occur in the operation phase.

For the management of these conflicts, in product development, several approaches have been developed from various aspects such as requirement elicitation, process management, and so forth [7-11]. For example, in requirement engineering, various approaches have been proposed to tackle conflict management in requirement elicitation [7]. In the management field, on the other hand, critical success factors are defined for conducting collaborative product development involving several stakeholders [8]. However, few studies deal with these conflicts from the viewpoint of PSS design.

To address this issue, this paper proposes a design procedure in consideration of the life cycle of a PSS. In the procedure, a design method is introduced in order for designers to detect the conflicts and to avoid them as much as possible in the development phase.

3 APPROACH OF THIS STUDY

3.1 Overview

In this study, the conflicts are considered from the viewpoint of independence among the elements of a PSS structure. This is because if independence among the elements is guaranteed, an operation for a particular element has no influence on the others, and therefore can be executed without conflicts with them. In consideration of independence, in this study, we adopt the approach of axiomatic design [12-14]. Since it is difficult to represent the structure of a PSS systematically and concretely, there are few studies that adopt the approach of axiomatic design in the context of PSS design. In order to represent the elements of a PSS structure, therefore, the modeling methods from Service Engineering [15-18] are adopted and are arranged according to the design domains proposed in axiomatic design.

The remainder of this section introduces two disciplines: axiomatic design and Service Engineering.

3.2 Axiomatic design

Axiomatic design proposes fundamental design principles; it is a methodology about how to think and use fundamental principles during mapping process among the domains of the design world [12-14]. The principle defines the elements that have respective domains: customer needs (CNs), functional requirements (FRs), design parameters (DPs), and process variables (PVs) (see Figure 2).

In the design process, CNs in the customer domain are converted into FRs in the functional domain. FRs are a minimum set of independent requirements that completely characterize the functional needs of the design solution. FRs are embodied into DPs in the physical domain, and then, DPs determine PVs in the process domain to produce and/or control the DPs.

Suh et al. state that all designers go through the same process, although the objectives may be different among various designs [12]. Table 1 shows how all these different design tasks can be described from the viewpoint of the four design domains. For example, in organization, after requisite functions of the organization (FRs) are determined, the next task is to design business programs and organizations (DPs) to meet the functions, and then find human and other resources (PVs) to staff and operate the business.

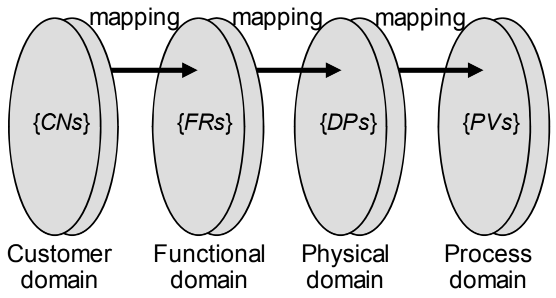

Figure 2: Four domains of the design world [12-14].

In axiomatic design, this mapping process is evaluated according to an axiom called "the Independence Axiom" [12-14], which is stated formally as:

> Maintain the independence of the functional requirements.

Under the independence axiom, the design should maintain the independence of FRs. Satisfying an FR with a DP that has effects on several FRs may cause a negative effect on the other FRs. Therefore, designers should associate FRs with DPs so that a DP has an effect on a single FR. In addition, the mapping process among the four domains can be expressed mathematically in terms of the characteristic vectors [12-14]. The set of FRs constitutes a vector {*FRs*} in the functional domain; the set of DPs in the physical domain constitutes a vector {*DPs*}. At each hierarchical level, the relationships between the {*FRs*} and the {*DPs*} can be represented with equation (1).

$$\{FRs\} = [\mathrm{DM}]\,\{DPs\} \qquad (1)$$

where [DM] is called "the design matrix".

For design of processes, in the same way, the design equation can be written as:

$$\{DPs\} = [\mathrm{DM}]\,\{PVs\} \qquad (2)$$

To satisfy the Independence Axiom, the matrix must be either diagonal or triangular [12-14]. When [DM] is diagonal, each of the FRs can be satisfied independently by one DP. Such a design is called an "uncoupled design" (see equation (3)). When the matrix is triangular, the independence of FRs can be guaranteed when the DPs are determined in a proper sequence. For example, in equation (4), if DP1 is firstly determined to satisfy FR1, DP1 can be considered as fixed value in satisfying the other FRs, and therefore D2 can be determined independently to satisfy FR2. Such a design is called a "decoupled design" (see equation (4)). All other designs violate the Independence Axiom and are called "coupled designs". Designers, therefore, need to develop design

Table 1: Features of the four domains of the design world for various designs [12].

Fields \ Domains	Customer domain	Functional domain	Physical domain	Process domain
Manufacturing	Attributes which customers desire	Functional requirements specified for the product	Physical variables which can satisfy the functional requirements	Process variables that can control design parameters
Materials	Desired performance	Required properties	Micro-structure	Process
Software	Attributes desired in the software	Output	Input variables and algorithms	Sub-routines
Organization	Customer satisfaction	Functions of the organization	Programs of offices	People and other resources that can support the programs
Systems	Attributes desired of overall system	Functional requirements of the system	Machines or components, sub-components	Resources (human, financial, materials, etc.)

solutions that have a diagonal or triangular design matrix.

$$\begin{Bmatrix} FR1 \\ FR2 \\ FR3 \end{Bmatrix} = \begin{bmatrix} X & 0 & 0 \\ 0 & X & 0 \\ 0 & 0 & X \end{bmatrix} \begin{Bmatrix} DP1 \\ DP2 \\ DP3 \end{Bmatrix} \quad (3)$$

$$\begin{Bmatrix} FR1 \\ FR2 \\ FR3 \end{Bmatrix} = \begin{bmatrix} X & 0 & 0 \\ X & X & 0 \\ X & X & X \end{bmatrix} \begin{Bmatrix} DP1 \\ DP2 \\ DP3 \end{Bmatrix} \quad (4)$$

3.3 Service Engineering

Service Engineering is a new engineering discipline with the objective of providing a fundamental understanding of services, as well as concrete engineering methodologies to design and evaluate services [15-18].

In Service Engineering, service is defined as an activity between a service provider and a service receiver to change the state of the receiver [15-17]. Note that the term "service" is used in a broad sense, and thus, the design target includes not only intangible human activities but also tangible products in a similar manner to that of PSSs. According to the definition, a receiver is satisfied when his/her state changes to a new desirable state. As the value of a service is determined by the receiver, service design should be based on the state change of the receiver. For design purposes, it is necessary to find a method to express the state changes of the receiver. The target receiver's state in a service design is represented as a set of parameters called receiver state parameters (RSPs) [15-17]. RSPs are changed by "service contents" and "service channels," as shown in Figure 3.

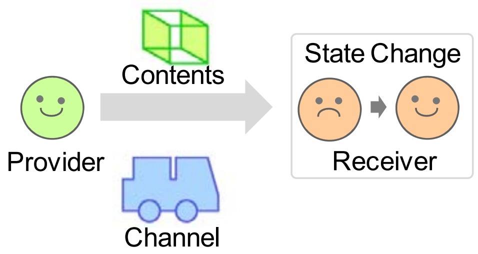

Figure 3: Definition of service [15-17].

In Service Engineering, it is assumed that service contents and service channels are comprised of various functions. In a realization structure for each RSP, function hierarchy is designed as shown in Figure 4 (a). In addition, these functions are performed by both service activities and product behaviors (Figure 4 (b)) that are actualized by attributes of entities (Figure 4 (c)). The entities represent not only physical products but also facilities, employees, and information systems. As a result, the framework representing the structure of a service can be illustrated as shown in Figure 4.

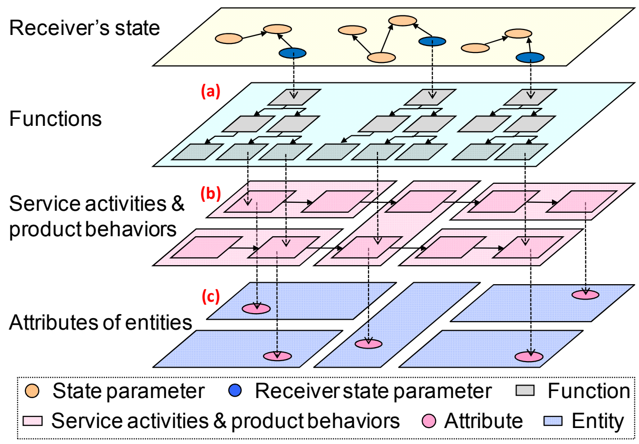

Figure 4: Framework for the representation of a service [18].

4 DESIGN METHOD FOR THE DEVELOPMENT OF PRODUCT-SERVICE SYSTEMS

4.1 Features of the four domains in PSSs

According to the models proposed in Service Engineering, in this study, features of the four domains in PSSs are defined as shown in Table 2.

In PSSs, the goal is to fulfill customer requirements. In the customer domain, therefore, customer requirements are defined as CNs and are represented as the receiver's state in the same manner as Service Engineering. The receiver's state includes not only his/her internal state but also his/her external state, such as the objectives of the organization to which the receiver belongs. With respect to the functional domain, functions realizing the state change are determined as FRs, and the service activities and product behaviors that perform the functions are then designed as DPs in the physical domain. Finally, in order to actualize the service activities and product behaviors, attributes of entities are determined as PVs in the process domain.

4.2 Design procedure

Overview

Figure 5 shows a PSS development process consisting of the value analysis and design phase in the life cycle. The process begins with the customer analysis, and the conceptual design is then carried out. In the conceptual design, designers first develop functional structure to fulfill the requirements extracted in the customer analysis. According to the functional structure, designers simultaneously determine the PSS features: service activities and product behaviors, and attributes of entities. Next, in the embodiment design, designers determine actual entities that embody the attributes determined in the conceptual design. Comparing these entities with the current one, designers specify the task in the transition phase, such as manufacturing products and preparing human resources. Finally, in the detailed design, designers develop plans to perform the tasks for the transition phase of the life cycle.

From the development process, especially, this study

Table 2: Features of the four domains in Product-Service Systems.

Fields \ Domains	Customer domain	Functional domain	Physical domain	Process domain
Product-Service Systems	Receiver's state	Functions for the receiver's state	Service activities and product behaviors	Attributes of entities

proposes a design procedure that includes the customer analysis and conceptual design (see the process colored gray in Figure 5). In this procedure, the conflicts that may occur in the operation phase are detected and are avoided as much as possible, and the inevitable conflicts are finally specified.

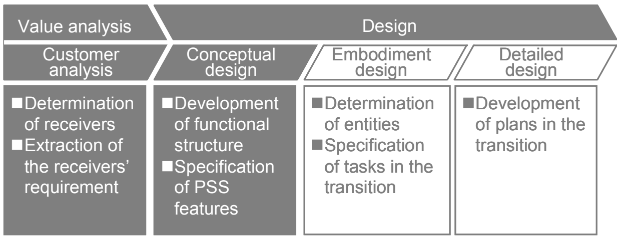

Figure 5: Development process for Product-Service Systems.

The remainder of this section introduces the design procedure in detail.

Step 1: Development of an initial flow model

In this design procedure, designers first define a chain of relevant stakeholders.

Many PSSs form complex structures consisting of many stakeholders. Between a receiver and a provider, there may be many intermediate stakeholders. As intermediate stakeholders also evaluate PSSs as service receivers, designers need to consider the requirements of the intermediate stakeholders, as well as those of an end receiver. In Service Engineering, a flow model is proposed to represent the chain of stakeholders (see Figure 6). In this step, an initial flow model is determined by defining the stakeholders and their relationships.

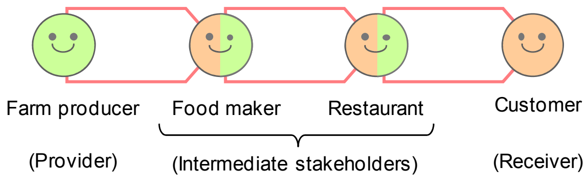

Figure 6: Flow model of a restaurant service [15-17].

Step 2: Extraction of RSPs of a receiver

In this step, in order to extract the RSPs of receivers, the designers describe the business process of the receiver. Next, from the business process, the stakeholders involved in the PSS are identified to extract their objectives, which are called practical goals [19]. Practical goals indicate objectives that should be achieved in individual tasks through the business process. For the identification of practical goals, a persona [19] is described for each stakeholder. The persona is a tool to give a simplified description of a customer and works as a compass in a design process [19]. In this procedure, the persona is described with a focus on the professional background, such as his/her daily tasks, and the designers then identify the practical goals of each stakeholder with reference to them. Based on this persona, a scenario is developed to clarify the context in which the PSS is operated. The scenario is described in the form of a state transition graph, as the purpose of receiving products and/or services in a PSS is to change the receiver's state into a more desirable one. The receiver's state is represented as a set of parameters called state parameters (SPs). From the SPs, RSPs, which correspond to target parameters in the PSS design, are extracted. Any SP can be defined as an RSP; however, for meaningful design to be realized, RSPs must be observable and related to the concrete requirements of a receiver.

Step 3: Decomposition in the four PSS domains

After the determination of RSPs, a PSS structure is designed through the decomposition of functions, service activities, and product behaviors, as well as attributes of entities.

In axiomatic design, a design method for large flexible systems is proposed [13]. Large flexible systems are defined as systems whose FRs are represented as time variant. In these systems, in particular, a subset of these FRs must be fulfilled at any given time, and the elements of this subset change as a function of time. With regard to PSSs, as receivers' requirements change depending on the phases of their scenario, PSSs can be regarded as large flexible systems. In this step, therefore, initial functions for the RSPs are determined in each phase of the scenario and are then decomposed until the design task is completed.

According to axiomatic design, the hierarchical structure of each domain is developed through zigzagging. Namely, the decomposition of these elements cannot be done by remaining in a single domain; however, it can be achieved through zigzagging between the domains [12-14] (see Figure 7). In this step, before decomposing the functions at a particular hierarchical level in the functional domain, the corresponding service activities and/or product behaviors must be designed as the elements of the same hierarchical level in the physical domain. In the same way, the designers determine attributes corresponding to the service activities and/or product behaviors through the zigzagging decomposition.

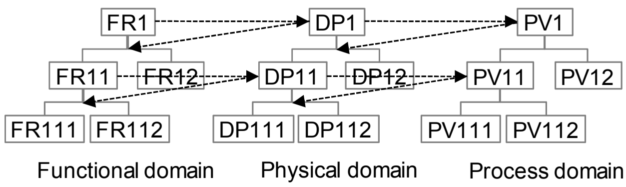

Figure 7: Zigzagging decomposition [12-14].

In this study, the conflicts existing in a PSS structure are considered from the viewpoint of independency among its elements, and therefore, at each hierarchical level, the design matrix must be diagonal or triangular to conform to the Independence Axiom. In particular, for functions at each hierarchical level, the corresponding service activities and/or product behaviors must be selected in order to guarantee the Independence Axiom; attributes must be associated with the corresponding activities and behaviors in the same manner.

Through this step, the designers can avoid conflicts by making attributes uncoupled. With regard to decoupled attributes, conflict can be avoided if the target values of the attributes are determined in a proper sequence. Finally, all of the other attributes are specified as one that contains the inevitable conflicts.

Step 4: Determination of components in a PSS

In the following process, these attributes are allocated to actual entities that may be manufactured or prepared in the transition phase. If decoupled/coupled attributes are allocated to separate entities, these entities depend on each other. Therefore, a stakeholder who takes responsibility for each entity needs to interact with the other stakeholders each time in changing the target value of the attributes. This situation complicates tasks in the transition phase and results in lower efficiency in terms of cost and time in the transition phase. Therefore, in this step, the designers determine PSS components as minimum units that are independent of each other.

PSS components consist of the modules proposed in axiomatic design. In axiomatic design, a module is defined in terms of the (FR/DP) or (DP/PV) relationship [13-14]. A module is defined as the row of the design matrix that fulfills an FR when it is provided with the input of its corresponding DP. In this study, on the other hand, a module is defined in terms of the (function/attribute) relationship. For the determination of PSS components, the designers first define modules from the (function/attribute) relationship. The relationship can be mathematically expressed as shown in equation 7.

{*Functions*} = [A] {*SAs, PBs*} (5)

{*SAs, PBs*} = [B] {*Attributes*} (6)

{*Functions*} = [C] {*Attributes*} (7)

Where {*Functions*}: a vector of functions

{*SAs, PBs*}: a vector of service activities and product behaviors

{*Attributes*}: a vector of attributes

[C] = [A]*[B]

As a result, in the design matrix [C], a module uncoupled from the others is defined as a single component; a minimum subset of decoupled modules is defined as a single component whose attributes need to be controlled sequentially; and a minimum subset of coupled modules is defined as a single component whose modules violate the Independence Axiom. For example, in Figure 8, module 1 is uncoupled from the other modules, and therefore, it can be defined as a single component solely. As module 3 requires the input of attribute 2, a subset of these decoupled attributes - that is, module 2 and 3 - is integrally defined as a single component. A subset of the coupled modules - that is, module 4 and 5 - is also defined as a single component.

	Attribute 1	Attribute 2	Attribute 3	Attribute 4	Attribute 5		
Function 1	X					Module 1	Component 1
Function 2		X				Module 2	Component 2
Function 3		X	X			Module 3	
Function 4				X	X	Module 4	Component 3
Function 5				X	X	Module 5	

Figure 8: Components in the design matrix.

A component of a PSS is defined as a minimum subset of modules that are independent from the other modules. The input of attributes in a particular component only influences the corresponding functions, and, therefore, has no influence on the other functions associated with the other components. Therefore, in the following process, the designers need to utilize components as the basis for the allocation of attributes to actual entities.

5 APPLICATION

In this chapter, the proposed method is applied to an e-learning service. The purpose of this application is to determine PSS components.

First, the initial flow model of the e-learning service was developed as shown in Figure 9. Next, the business process of the client company and the system vendor was described. The business process proceeded in five steps: planning the e-learning course, developing materials, preparing for the hosting of the e-learning course, holding the e-learning course, and evaluating the e-learning course. According to the process, the stakeholders involved in this PSS were identified (see Figure 9 below).

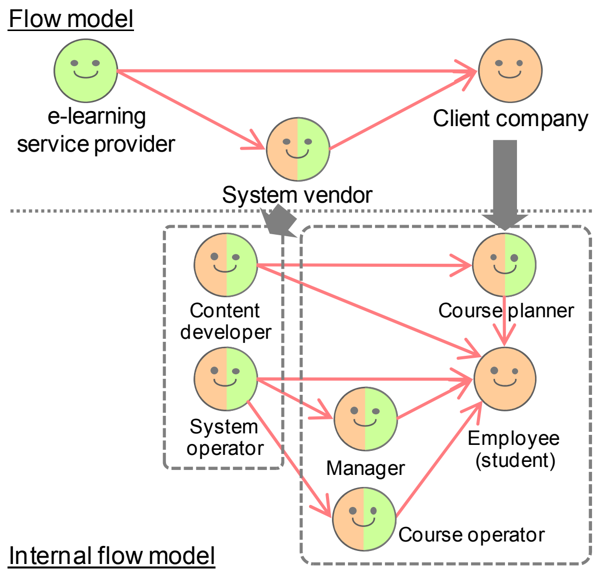

Figure 9: Flow model of the e-learning service.

In this example, the client company and the system vendor were comprised of six stakeholders: content developers, system operators, course planners, course operators, managers and employees who take the e-learning (students). For the extraction of their practical goals, a persona was subsequently described for each stakeholder. The course planners take responsibility for the planning of courses based on the program of human resource development. The contents developer provides the development environment of e-learning contents to the course planner and registers e-learning contents on the e-learning system. The system operator operates the e-learning system and manages relevant data, such as the list of students and their progress status. The manager plans the program of human resource development and manages it based on the reports of progress status provided from the e-learning system. The course operator takes responsibility for tasks relating to the conduct of courses, such as announcements, applications and student support, by using the e-learning system. Based on the personas, a scenario was described for each stakeholder, and RSPs were extracted as shown in Table 3.

Table 3: The list of Receiver State Parameters.

Phases	Receiver State Parameters
Planning the e-learning course	CN1: Feasibility of system operation CN2: Feasibility of course planning
Developing materials	CN3: Accuracy of content development CN4: Timely offerability of learning contents
Preparing for the hosting of course	CN5: Accuracy of processing CN6: Certainty of status follow-up
Holding the e-learning course	CN7: Responsiveness of student support
Evaluating the e-learning course	CN8: Accuracy of implementation report CN9: Exactness of evaluation

Table 4: Decomposition of functions, service activities/product behavior and attributes of entities.

Functions	Service activities and product behaviors	Attributes of entities
FR1: Organize a course	DP1: Course planning	PV1: Capability of planning
FR11: Draw up the proposal for a course	DP11: Education planning	PV11: Capability of education planning
FR12: Organize group training	DP12: Planning of group training	PV12: Capability of organizing group training
FR2: Design system operation	DP2: Design of system operation	PV2: Certainty of system operation
FR21: Ensure system security	DP21: System security	PV21: Certainty of Security
FR22: Support system	DP22: Support system	PV22: Certainty of Support
FR23: Determine the roles in operation	DP23: Operation design	PV23: Capability of operation design
FR3: Make up materials	DP3: Material preparation	PV3: Capability of material preparation
FR4: Develop e-learning contents	DP4: Development of e-learning contents	PV4: Quality of materials
FR41: Conduct logical design and physics design	DP41: Development of e-learning contents	PV41: Capability of contents development
FR42: Upload e-learning contents onto server	DP42: Management of e-learning contents	PV42: Capability of management
FR5: Announce a course	DP5: Course announcement	PV5: Quickness of communication
FR6: Follow up learning status	DP6: Management of learning status	PV6: Capability of information control
FR61: Manage learning status	DP61: Management of course participation	PV61: Real-time update
FR62: Report learning status	DP62：Status report	PV62: Capability of communication
FR7: Support students	DP7: Student support	PV7: Quickness of response
FR8: Conduct implementation report	DP8: Implementation report	PV8: Accuracy of implementation report
FR81: Report completion status	DP81: Management of completion status	PV81: Completeness of information
FR82: Collect the questionnaire	DP82: Questionnaire	PV82: Collection rate of questionnaire
FR9: Evaluate a course	DP9: Course evaluation	PV9: Accuracy of evaluation
FR91: Evaluate achievement of objectives	DP91: Evaluation of achievement objectives	PV91: Comprehension of objectives
FR92: Reflect user requests on a course	DP92: Reflection of user feedback	PV92: Amount of user feedback

		PV1		PV2			PV3: Capability of material preparation	PV4		PV5: Quickness of communication	PV6		PV7: Quickness of response	PV8		PV9			
		PV11: Capability of education planning	PV12: Capability of organizing group training	PV21: Certainty of security	PV22: Certainty of support	PV23: Capability of operation design		PV41: Capability of contents development	PV42: Capability of management		PV61: Real-time update	PV62: Capability of communication		PV81: Completeness of information	PV82: Collection rate of questionnaire	PV91: Comprehension of objectives	PV92: Amount of user feedback		
FR1	FR11: Draw up the proposal for a course	X																M11[*1]	C1[*2]
	FR12: Organize group training	X	X															M12	
FR2	FR21: Ensure system security			X														M21	C2
	FR22: Support system				X													M22	C3
	FR23: Determine the roles in operation					X												M23	C4
FR3: Make up materials							X											M3	C5
FR4	FR41: Conduct logical design and physics design						X	X	X									M41	
	FR42: Upload e-learning contents onto server							X	X									M42	
FR5: Announce a course										X								M5	C6
FR6	FR61: Manage learning status										X	X						M61	C7
	FR62: Report learning status										X	X						M62	
FR7: Support students													X					M7	C8
FR8	FR81: Report completion status													X				M81	C9
	FR82: Collect the questionnaire														X			M82	
FR9	FR91: Evaluate achievement of objectives													X		X		M91	
	FR92: Reflect user requests on a course														X		X	M92	

*1 M: module, *2 C: component

Figure 10: Full design matrix table of the e-learning service.

For each RSP, a realization structure was then developed. First, a function for each RSP was determined, and the zigzagging decomposition was carried out to design the corresponding service activities, product behaviors, and attributes. Table 4 shows the result of the decomposition of each domain. Simultaneously, in each hierarchical level, the design matrices were determined by developing the equation 5 and 6.

Finally, based on the design matrices, equation 7 was developed to obtain the (function/attribute) relationship, and components of the PSS were defined. Figure 10 shows the design matrix that describes the relationships between the functions and attributes. In the design matrix, uncoupled modules include M21-23, M5, and M7; decoupled modules include M11-12, M81-82, and M91-92; coupled modules include M3, M41-42, and M61-62. As a result, for example, the uncoupled modules, M21-23, M5, and M7, were determined to be a single component, respectively. With regard to decoupled modules, a subset of decoupled modules, for example the set of M11-13, was determined as a single component. In the same way, a subset of coupled modules, for example the set of M61-62, was determined as a single component (see Figure 10, right column).

6 DISCUSSION

6.1 Effectiveness of the proposed method

In this application, the proposed design procedure was applied to an e-learning service. According to the approach of axiomatic design, elements of each domain, which are functions, service activities and product behaviors, and attributes of entities, were decomposed in consideration of their independence.

As a result, the conflict could be avoided by making the following modules uncoupled: M21-23, M5, and M7. With respect to the decoupled modules, which are M11-12, M81-82, and M91-92, the conflict can be avoided if the target values of corresponding attributes can be determined in a proper sequence. In the case of M11-12, the target values of the corresponding attributes need to be determined in the following order: “Capability of education planning” and “Capability of organizing group training”. Finally, M3, M41-42, and M61-62 were identified as containing the inevitable conflicts, and therefore their attributes, for example “Capability of material preparation”, “Capability of contents development” and “Capability of management”, conflict with each other.

For a successful PSS, several stakeholders must be involved [5]. This situation may cause incompatibilities among their objectives and tasks and will inevitably induce conflicts. The design procedure proposed in this paper enables designers to detect such conflicts arising in the operation of a PSS and avoid them as much as possible in the design stage.

In addition, the nine components were determined to be minimum subsets of modules that are independent of each other. The input of attributes in a particular component only influences the corresponding functions and has no influence on the other functions associated with the other components. Therefore, if the interface among components is completely defined, tasks for each component in the transition phase can be assigned to independent stakeholders, respectively, and be executed concurrently. PSSs consist of various components, such as products, software, systems, and organization, that are assigned several stakeholders to prepare them. Since the components that are determined in the proposed method are independent of each other, stakeholders involved in the PSS are able to prepare the assigned components concurrently. Therefore, the proposed method is useful for the efficiency of tasks in the transition phase.

With regard to the representation of a PSS structure, according to the modeling method from Service Engineering functions, service activities and product behaviors, and attributes of entities are defined as the elements of the four domains of the design world. Whereas PSSs consist of various components, the modeling method can be considered as useful for describing the four domains of PSS design.

6.2 Possible improvements of the proposed method

In this application, the design procedure drew “the ideal PSS structure” that avoids conflicts as much as possible. However, in reality, in the transition phase to realizing such a structure, stakeholders need to prepare the requisite resources that include not only tangible resources, such as infrastructure, but also intangible resources, such as knowledge and skills. The preparation of these resources may be constrained by their current states; the constraints define the bounds on the acceptable design. Therefore, the extension of the proposed procedure is required to consider the constraints that arise in the transition from current structure to ideal one.

7 CONCLUSION

In this paper, a design method is proposed to address the conflicts in PSS development. In the method, the approach of axiomatic design is adopted to detect and avoid these conflicts. In addition, the modeling methods from Service Engineering are applied for the representation of PSS structure. The application reveals that the proposed procedure enables designers to detect and avoid conflict as much as possible.

Future studies will include the extension of the procedure to consider the constraints that arise in the transition phase.

8 REFERENCES

[1] Tomiyama, T., 1997, A Manufacturing Paradigm Toward the 21st Century, Integrated Computer Aided Engineering, 4: 159-178.

[2] Vargo, S.L. and Lusch, R.F., 2004, Evolving to a New Dominant Logic for Marketing, Journal of Marketing, 68/1: 1-17.

[3] Tukker, A. and Tischner, U., 2006, Product-services as a research field: past, present and future. Reflections from a decade of research, Journal of Cleaner Production, 14/17:1552-1556.

[4] Morelli, N., 2003, Product-Service Systems -a Perspective Shift for Designers-, A Case Study -The Design of a Telecentre-, Design Studies, 24:73-99.

[5] Mont, O.K., 2002, Clarifying the concept of product–service system, Journal of Cleaner Production, 10: 237–245.

[6] Duarte, M. and Davies, G., 2003, Testing the conflict–performance assumption in business-to-business relationships, Industrial Marketing Management, 32: 91– 99.

[7] Lamsweerde, A.V., Darimont, R. and Letier, E. 1998, Managing Conflicts in Goal-Driven Requirements Engineering, IEEE Transactions on Software Engineering.

[8] Lam, P.K. and Chin, K.S., 2005, Identifying and prioritizing critical success factors for conflict management in collaborative new product development, Industrial Marketing Management, 34: 761– 772.

[9] Kim, M., Park, S., Sugumaran, V. and Yang, H., 2007, Managing requirements conflicts in software

product lines: A goal and scenario based approach, Data & Knowledge Engineering 61: 417–432.

[10] Shaw, V., Shaw, C.T and Enke, M., 2003, Conflict between engineers and marketers: the experience of German engineers, Industrial Marketing Management, 32: 489– 499.

[11] Ouertani, M.Z., 2008, Supporting conflict management in collaborative design: An approach to assess engineering change impacts, Computers in Industry, 59: 882–893.

[12] Suh, N.P., 1990, The Principles of Design, Oxford University Press, New York.

[13] Suh, N.P., 1998, Axiomatic Design Theory for Systems, Research in Engineering Design, 10/4: 189-209.

[14] Suh, N.P. and Do, S.H., 2002, Axiomatic design of software systems, CIRP Annals - Manufacturing Technology, 49/1: 95-100.

[15] Arai, T. and Shimomura, Y., 2004, Proposal of Service CAD System: A Tool for Service Engineering, Annals of the CIRP, 53/1:397-400.

[16] Shimomura, Y. and Tomiyama, T., 2005, Service Modeling for Service Engineering, IFIP International Federation for Information Processing, 167: 31-38.

[17] Sakao, T. and Shimomura, Y., 2007, Service Engineering: A Novel Engineering Discipline for Producers to Increase Value Combining Service and Product-, Journal of Cleaner Production, 15/6: 590-604.

[18] Hara, T., Arai, T. and Shimomura, Y., 2009, A CAD system for service innovation: integrated representation of function, service activity, and product behaviour, Journal of Engineering Design, 20/4: 367-388.

[19] Cooper, A., 1999, The Inmates Are Running the Asylum, SAMS/Macmillan, Indianapolis, IA.