INTERNATIONAL DESIGN CONFERENCE - DESIGN 2012
Dubrovnik - Croatia, May 21 - 24, 2012.

# DESIGN METHOD FOR MODULAR PRODUCT-SERVICE SYSTEM ARCHITECTURE

K. Kimita and Y. Shimomura

*Keywords: service engineering, product-service systems, modular design, axiomatic design, design structure matrix*

## 1. Introduction

Environmental problems have grown in importance over the last couple of decades. Consequently, society needs to reduce the production of artefacts without impairing the current quality of life. To this end, it would be effective to pursue qualitative satisfaction over quantitative sufficiency and thus decouple economic growth from material consumption [Tomiyama 1997]. For this purpose, manufacturers are starting to recognize that services and knowledge provided by a product are more important than the product itself [Vargo 2004]. As a result, research that aims to create value by coupling a physical product and a service, such as Service Engineering (e.g. [Shimomura 2005]) and Product-Service Systems (e.g. [Mont 2002]), have been attracting attention.

A PSS is defined as a system of products, services, supporting networks and infrastructure that is designed to be competitive, satisfy customer needs and have a lower environmental impact than traditional business models [Mont 2002]. According to this definition, in order to achieve a successful PSS, the stakeholders involved in the PSS are obliged to extend their responsibility in the life cycle [Mont 2002]. This is because, from the viewpoint of environmental issues, providers need to establish proper organization for managing the product life cycle, such as reuse, remanufacturing, and recycling. They also need to educate receivers on efficient use and proper disposal of products. With respect to the value creation process, value is always determined by customers [Vargo 2004]; providers, therefore, need to construct systems for the observation of customers' needs and establish networks to share relevant information. To compensate these extended responsibilities, new and varying types of stakeholders must be involved [Mont 2002]. In addition, these stakeholders have to cope with dynamic changes, such as resources, market demands, changing customer requirements and continuous improvements arising from gained knowledge [Meier 2009]. The stakeholders, therefore, need to have the adaptability to compete within these changes. However, addressing the changes requires much more complex interactions among relevant stakeholders; these types of interactions involve tremendous coordination efforts.

To solve this problem, many researchers (e.g. [Meier 2009]) state that defining a PSS architecture is one of the most important decisions to solve coordination complexity. In particular, the modularization of PSS architecture is defined as one of the key principles. For example, the modular organization concept of PSSs is able to assign processes to resources of multiple stakeholders and provide flexibility in changing processes or resources [Meier 2009]. However, few studies have focused on the design of modular PSS architecture.

This study assumes that the modular PSS architecture enables relevant stakeholders to address the complex interactions among them and to cope with the dynamic changes. This paper, therefore, aims to provide a concrete methodology for the modularization of PSS architecture. In particular, we are proposing a design method for the determination of PSS modules. The effectiveness of the method is demonstrated by the application, where an e-learning service is used as an example.

## 2. Approach of this study

### 2.1 The module of product-service systems

In product design, the product architecture is defined as 'the scheme by which the function of a product is allocated to physical components' [Ulrich 1995]. According to this definition, the modularization of product architecture can be achieved while maintaining the functional independency of physical components and minimizing the coupling among the components.

In the same manner as the product architecture, we determine the PSS architecture by the allocation of PSS functions to entities. Note that the entities include not only physical products but also non-physical products, such as systems and organizations. Based on the concept of the PSS architecture, PSS modules can be defined as clusters of entities that maintain the functional independency and minimize the coupling among the clusters.

### 2.2 Approach for the determination of the modules of product-service systems

For the determination of PSS modules, the proposed method includes two phases: the decomposition and the integration of PSS entities. PSS entities are, first, decomposed into sub-entities that maintain their functional independency. For the decomposition, we adopt the Independence Axiom from Axiomatic Design [Suh 1990]. These sub-entities are subsequently reintegrated into clusters that minimize the coupling among them. For the integration, we adopt the clustering method in Design Structure Matrix (e.g. [Browning 2001]).

The remainder of this section introduces the relevant works, *Axiomatic Design* and *Design Structure Matrix*.

#### *2.2.1 Axiomatic design*

Axiomatic design proposes fundamental design principles. It is a methodology about how to use fundamental principles during the mapping process among the domains of the design world [Suh 1990]. The principle defines the elements that have respective domains: customer needs (CNs), functional requirements (FRs), design parameters (DPs), and process variables (PVs) (see Figure 1).

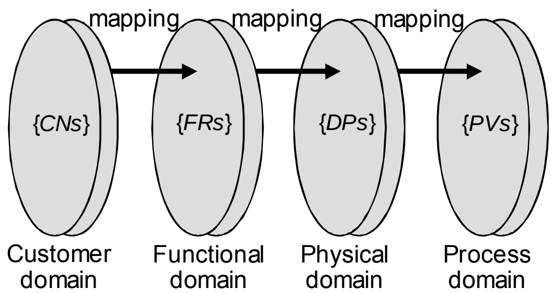


**Figure 1. Four domains of the design world [Suh 1990]**

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.

In axiomatic design, this mapping process is evaluated according to an axiom called the Independence Axiom [Suh 1990], 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 [Suh 1990]. The set of FRs constitutes a vector {FRs} in the functional domain; the set of DPs in the physical domain constitute a vector {DPs}. At each hierarchical level, the relationships between the {FRs} and the {DPs} can be represented with Equation 1.

$$\{FRs\} = [\mathrm{DM}]\ \{DPs\} \tag{1}$$

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

To satisfy the Independence Axiom, the matrix must be either diagonal or triangular [Suh 1990]. When DM is diagonal, each of the FRs can be satisfied independently by one DP. Such a design is called an 'uncoupled design' (see Figure 2(a)). When the matrix is triangular, the independence of FRs can be guaranteed when the DPs are determined in a proper sequence. For example, in Figure 2(b), if DP1 is first determined to satisfy FR1, DP1 can be considered as a fixed value in satisfying the other FRs. Therefore D2 can be determined independently to satisfy FR2. Such a design is called a 'decoupled design'. All other designs violate the Independence Axiom and are called 'coupled designs'. Designers, therefore, need to develop design solutions that have a diagonal or triangular design matrix.

$$\begin{Bmatrix} FR1 \\ FR2 \\ FR3 \end{Bmatrix} = \begin{bmatrix} X & 0 & 0 \\ 0 & X & 0 \\ 0 & 0 & X \end{bmatrix} \begin{Bmatrix} DP1 \\ DP2 \\ DP3 \end{Bmatrix} \qquad \begin{Bmatrix} FR1 \\ FR2 \\ FR3 \end{Bmatrix} = \begin{bmatrix} X & 0 & 0 \\ X & X & 0 \\ X & X & X \end{bmatrix} \begin{Bmatrix} DP1 \\ DP2 \\ DP3 \end{Bmatrix}$$

(a) Uncoupled design (b) Decoupled design

**Figure 2. The design matrix [Suh 1990]**

This adopts the design matrix to evaluate the functional dependency of PSS entities. According to the result of the design matrix, PSS entities are decomposed into sub-entities that maintain the functional independency.

*2.2.2 Design structure matrix (DSM)*

The design structure matrix (DSM) (e.g. [Browning 2001]) is becoming a popular representation and analysis tool for system modelling. A DSM displays the relationships between elements of a system. As shown in Figure 3, a DSM is a square matrix with identical row and column labels. An off-diagonal mark represents an element's dependence on another. Reading across a row reveals what other elements are provided by the element in that row. Scanning down a column reveals what other elements the element in that column depends on. For example, in Figure 3, element B provides input to elements A, C, D, F, H, and I, and it depends on outputs from elements C, D, F, and H.

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| Element A | A | | | | | | | | |
| Element B | ■ | B | ■ | ■ | | ■ | | ■ | ■ |
| Element C | ■ | ■ | C | | ■ | ■ | | ■ | ■ |
| Element D | ■ | ■ | | D | ■ | | ■ | ■ | ■ |
| Element E | ■ | | ■ | ■ | E | | ■ | ■ | ■ |
| Element F | | ■ | ■ | | | F | | | |
| Element G | | | | ■ | ■ | | G | | |
| Element H | | ■ | ■ | ■ | ■ | | | H | |
| Element I | ■ | | ■ | | ■ | | | | I |

**Figure 3. Example of design structure matrix [Browning 2001]**

DSMs are classified into two main categories: static DSMs and time-based DSMs. Static DSMs represent system elements, such as components of a product architecture or groups in an organization. Static DSMs are usually analyzed with clustering algorithms. In time-based DSMs, on the other hand, the order of the rows and columns indicates a flow through time. Therefore, time-based DSMs represent characteristics of a process sequence, such as feedforward and feedback. Time-based DSMs are typically analyzed using sequencing algorithms.

This study adopts the clustering algorithms in static DSMs. These algorithms are generally clustering along the diagonal marks by reordering the rows and columns of the DSM. Clustering requires several considerations. The foremost objective is to maximize interactions between elements within clusters while minimizing interactions between clusters.

## 3. Design method for modular product-service system architecture

### 3.1 Overview

According to the concept of the modular product architecture, a PSS module is defined as clusters of entities that maintain functional independency and minimize coupling among clusters. For the determination of the PSS modules, this paper proposes a design procedure that is comprised of two phases: the decomposition, and the integration of PSS entities. Thorough Steps 1 to 3 in the following section, the dependency among PSS functions and entities is analyzed from the viewpoint of the Independence Axiom, and then PSS entities are decomposed into sub-entities that maintain the functional independency. Subsequently, through Steps 4 to 6, the dependency among the sub-entities is analyzed by using the clustering method in the Design Structure Matrix. Then, sub-entities are reintegrated into clusters of sub-entities that minimize the coupling among the clusters for the determination of PSS modules.

### 3.2 Design procedure for the determination of modular product-service system architecture

#### *3.2.1 Step 1: Extraction of customer requirements*

From the viewpoint of offering products in combination with services, value is always determined by customers. Therefore, this procedure begins with the extraction of customer requirements in a PSS. In order to extract the customer requirements, the designers describe the business process of the customer. Next, from the business process, the stakeholders involved in the PSS are identified to extract their objectives, which are called practical goals [Cooper 1999]. Practical goals indicate objectives that should be achieved in individual tasks through the business process. For the identification of practical goals, a persona [Cooper 1999] is described for each stakeholder. The persona is a tool to give a simplified description of a customer and works as a compass in the design process. In this procedure, the persona is described with a focus on professional background, such as his/her daily tasks. The designers then identify the practical goals of each stakeholder. 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 customer's state is represented as a set of parameters called state parameters (SPs) [Shimomura 2005]. In this method, the target customer requirements are represented as a set of parameters called receiver state parameters (RSPs) [Shimomura 2005]. Any SP can be defined as an RSP; however, for a meaningful design to be realized, RSPs must be observable and related to the concrete requirements of a receiver.

#### *3.2.2 Step 2: Development of a realization structure for each RSP*

In this step, the designers determine a realization structure for each RSP. In this method, we adopt the view model [Shimomura 2005] as the modeling method for describing the realization structure.

A view model is described in terms of the functional relationships among RSPs, functions and entities. It is assumed that contents that realize customer requirements in a PSS are comprised of various functions [Shimomura 2005]. These functions are expressed by function names (FNs) as lexical expressions and function parameters (FPs) as target parameters of the functions. In addition, the realization structure is associated with entities and their attributes that actualize the functions in the view model. Entities in the view model represent not only physical products but also facilities, employees, and information systems.

As shown in Figure 4, the view model works as a bridge from an RSP to entities and thus allows designers to clarify the roles of the artefacts in consideration of the RSP.

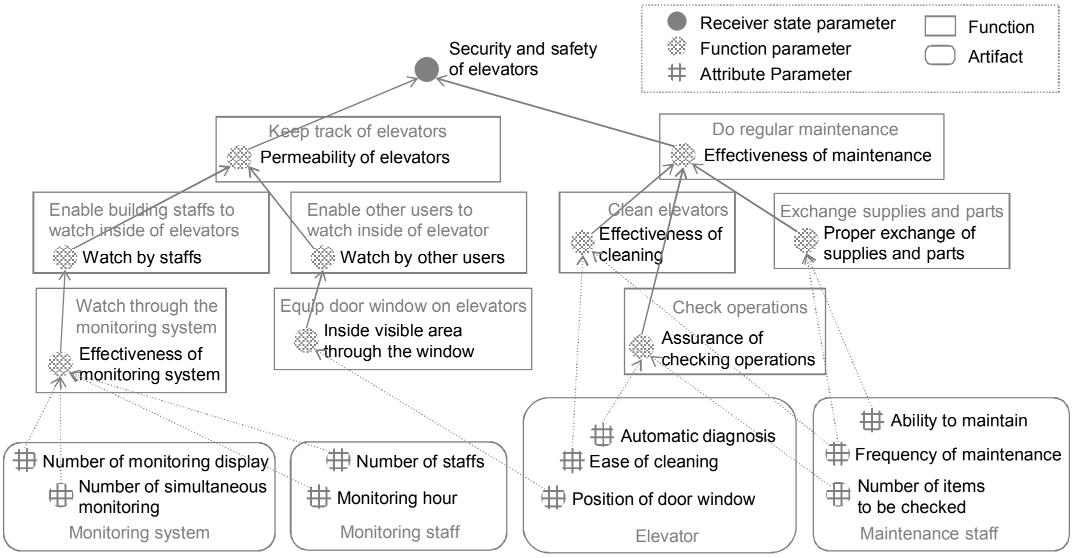


**Figure 4. Example of the view model (elevator maintenance service) [Shimomura 2005]**

### *3.2.3 Step 3: Decomposition of PSS entities*

In this step, designers first analyze the functional independency of attributes of entities in the view models developed in Step 2. Designers develop the design matrix that represents the relationships among the function parameters (FPs) and attribute parameters (APs) in the view models, and then make the design matrix diagonal or triangular. According to the Independence Axiom from Axiomatic Design, designers analyze the functional independency of the APs.

Based on the result of the design matrix, designers next specify sub-entities that maintain the functional independency. If decoupled/coupled APs are allocated to separate entities, these entities depend on each other. Therefore, the stakeholder who takes responsibility for each entity needs to interact with the other stakeholders while changing these APs. This situation complicates the task of developing and operating these entities, and results in lower efficiency in terms of cost and time. Therefore, in this step, the designers decompose the entities into sub-entities that maintain the functional independency.

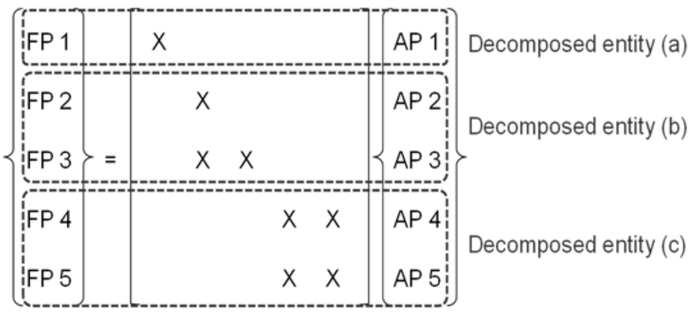


**Figure 5. Example of entity decomposition**

For APs uncoupled from the others, as shown in Figure 5(a), a sub-entity can be defined for each AP. For APs decoupled from the others, as shown in Figure 5(b), the designers need to define a single sub-entity containing a minimum subset of decoupled APs. If decoupled APs are allocated to several sub-entities, coordination among these sub-entities can be minimized when designers select a proper AP for a particular function improvement. For APs coupled to the others, as shown in Figure 5(c), designers need to allocate a minimum subset of coupled APs to a single sub-entity.

### *3.2.4 Step 4: Specification of the dependency among PSS entities*

In this method, dependencies among PSS entities are represented as functional interrelationships. Rodenacker states that functions of many technical systems can be represented as input/output relationships of ‘energy’, ‘material’ and ‘information’ [Rodenacker 1971]. Since PSSs are also considered one of the technical systems in this method, dependencies among PSS entities are specified from the viewpoint of the input/output relationships: ‘energy’, ‘material’ and ‘information’.
In this step, therefore, the designers develop a functional input-output model based on the view models described in Step 2.

### *3.2.5 Step 5: Integration of PSS entities*

Next, a DSM is developed according to dependencies among PSS entities, which are described in terms of functional input-output. The DSM is described using a binary variable that represents whether input/output relationships exist among sub-entities.
According to the dependencies in the matrix, a clustering of off-diagonal elements is conducted. In this method, the equation proposed by Fernandez [Fernandez 1998] is adopted as the evaluation formula of the clustering. This formula aims to minimize the coordination cost among teams in a product development project. In this study, the formula is applied to PSS design in order to find clusters of entities that minimize coordination cost.
The formula first calculates a coordination cost for each entity in the DSM, and then the sum of the coordination costs for each entity provides a total coordination cost. Equations 2-3 show the coordination cost for an entity *i*.

If both entity *i* and *j* are in any cluster *k*;

$$\text{Coordination Cost}(\text{Entity}_i) = \sum_{j=1}^{size} \big(\text{DSM}(i,j) + \text{DSM}(j,i)\big) * \sum_{k=1}^{Cl} \text{cl\_size}(k)^{pow\_cc} \quad (2)$$

If no *k* cluster contains both entity *i* and *j*;

$$\text{Coordination Cost}(\text{Entity}_i) = \sum_{j=1}^{size} \big(\text{DSM}(i,j) + \text{DSM}(j,i)\big) * size^{pow\_cc} \quad (3)$$

where;

*size* is the size of the DSM: the number of entities in the DSM
$\text{DSM}(i,j)$ is the value of the dependency between entity *i* and *j*. Note that when $i = j$, $\text{DSM}(i,j) = 0$, $\text{DSM}(j,i) = 0$
*Cl* is the maximum number of clusters (set to the number of entities in this analysis)
*cl_size* is the number of entities contained in cluster *k*
*pow_cc* is a parameter that controls the type of penalty assigned to the size of the cluster in the coordination cost (set to 2 in this analysis)

The total coordination is expressed as Equation 4. This objective function is the expression that the algorithm attempts to minimize.

$$\text{Total Coordination Cost} = \sum_{i=1}^{size} \text{Coordination Cost}(\text{Entity}_i) \quad (4)$$

The clustering algorithm that employs Equations 3-5 consists of two operations: partitioning and shuffling. Partitioning elements determines the size (*cl_size*) and number of a cluster; shuffling elements conducts the reordering entities in the DSM. In order to obtain a high-quality result with a limited calculation amount, we adopt in this analysis a genetic algorithm for the partitioning and a local search algorithm for the shuffling.

*3.2.6 Step 6: Determination of PSS modules*

In this method, PSS entities are decomposed into sub-entities that maintain functional independency in Step 3. In Step 5, the integration of these sub-entities is carried out to specify the clusters that minimize the dependencies among them. As a result, functions of these clusters can be precisely specified, and dependency among these clusters fully characterized. In this method, the cluster of sub-entities is therefore defined as a minimum subset of PSS modules. PSS modules consisting of these clusters indicate that the change in a particular cluster only influences the corresponding functions, and therefore has no influence on the functions associated with other clusters. Such conditions enable a relevant stakeholder to reduce coordination efforts resulting from changes. In the following process of PSS development, the designers enable the assigning of PSS modules to resources of multiple stakeholders and provide flexibility in changing resources.

## 4. Application

In this chapter, the proposed method is applied to an e-learning service where a client company corresponds with the customer. The purpose of this application is to determine the PSS modules.

First, the business process of the client company was described. The business process proceeded in five steps: planning the e-learning course, developing materials, preparing for the hosting of the course, holding the course, and evaluating the course. According to the process, the stakeholders involved in this PSS were identified. For the extraction of their practical goals, a persona was subsequently described for each stakeholder. Based on the personas, a scenario was described for each stakeholder. Next, 'Quality of course planning', 'Efficiency of education management', and 'Safety and stability of system' were extracted as RSPs that correspond to the requirements of the client company.

Next, functions that realize these RSPs were developed. Through the subsequent determination of PSS entities that actualize these functions, the relationships among FPs and APs were obtained. Figure 6 shows an example of the view model for 'Quality of course planning'. An FP, 'Compatibility with HR development', is associated with an AP named 'Capability for developing others' in an entity called 'Course planner'.

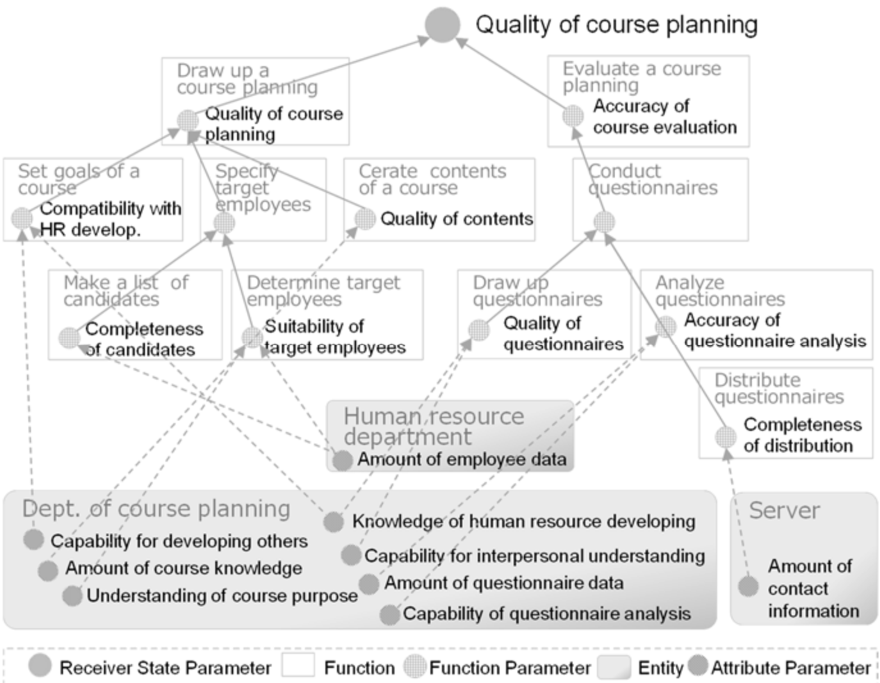


**Figure 6. A view model for 'Quality of course planning'**

To analyze the functional independency of the APs, a design matrix was developed based on the relationships among FPs and APs in the view models. Functional independency of the APs was analyzed from the view point of the Independence Axiom by making the design matrix triangular. Figure 7 shows the result of the design matrix.

| F1 Quality of contents | F2 Compatibility with HR develop. | F3 Completeness of candidates | F4 Suitability of target employees | F5 Quality of questionnaires | F13 Quality of announcement text | F14 Achievement rate of distribution | F10 Achievement rate of assignment of reminder | F6 Completeness of distribution | F7 Accuracy of questionnaire analysis | F8 Accuracy of progress management of study | F9 Accuracy of attribution of reminder | F11 Adequacy of operation planning | F21 Planning of maintenance | F12 Adequacy of schedule | F15 Achievement rate of support of request | F16 Achievement rate of e-contents | F17 Conversion rate of e-contents | F18 Achievement rate of server registry | F19 Efficiency of management of e-contents | F20 Stability of system | F22 Achievement rate of backup | Attribute parameters | | Decomposed entities |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X | | | | | | | | | | | | | | | | | | | | | | A1 | Amount of course knowledge | E1 Div. of course planning 1 |
| X | | | | | | | | | | | | | | | | | | | | | | A2 | Capability for developing others | |
| | X | | | | | | | | | | | | | | | | | | | | | A3 | Knowledge of human resource developing | E2 HR department (a) |
| | | X | X | | | | | | | | | | | | | | | | | | | A11 | Amount of employee data | E3 Div. of course planning 2 |
| | | | X | | | | | | | | | | | | | | | | | | | A4 | Understanding of course purpose | |
| | | | | X | X | | | | | | | | | X | | | | | | | | A5 | Retention rate of planning data of study | E4 Div. of course planning 3 (b) |
| | | | | | X | | | | | | | | | | | | | | | | | A6 | Comprehension of human interaction | |
| | | | | | | X | X | X | | | | | | | | | | | | | | A12 | Retention rate of contact information | E5 Mail server |
| | | | | | | | | | X | | | | | | | | | | | | | A7 | Collection rate of questionnaire | E6 Div. of course planning 4 (c) |
| | | | | | | | | | X | | | | | | | | | | | | | A8 | Capability of questionnaire analysis | |
| | | | | | | | | | | X | X | | | | | | | | | | | A13 | Frequency of update of the study progress | E7 Div. of course management |
| | | | | | | | | | | X | X | | | | | | | | | | | A14 | Collection rate of student data | |
| | | | | | | | | | | | X | | | | | | | | | | | A15 | Capacity to analyze a study progress | |
| | | | | | | | | | | | | X | X | X | | | | | | X | | A16 | Retention rate of a load data of system | E8 Div. of system operation 1 |
| | | | | | | | | | | | | | X | X | | | | | | | | A9 | Capability of pursue a system efficient | |
| | | | | | | | | | | | | | | X | X | | | | | | | A17 | Capability of pursue a support efficient | E9 Div. of course planning 5 |
| | | | | | | | | | | | | | | X | X | | | | | | | A10 | Frequency of updating of request data | |
| | | | | | | | | | | | | | | | | X | X | | | | | A20 | Retention rate of knowledge of e-contents | E10 Div. of contents development |
| | | | | | | | | | | | | | | | | | X | | | | | A21 | Retention rate of text knowledge | |
| | | | | | | | | | | | | | | | | | | X | | | | A18 | Retention rate of registry knowledge | E11 Div. of system operation 2 |
| | | | | | | | | | | | | | | | | | | | X | | | A22 | Retention rate of a data of e-contents | E12 Contents data base |
| | | | | | | | | | | | | | | | | | | | | X | | A19 | Capability of pursuit an effort | E13 Div. of system operation 3 |
| | | | | | | | | | | | | | | | | | | | | | X | A23 | Frequency of backup | E14 Back up server |

**Figure 7. Design matrix table for e-learning service**

Based on the analysis of functional independency of the APs, the entities were decomposed into sub-entities that maintain the functional independency. For example, as shown in Figure 7(a), for APs uncoupled from the others, such as 'A3: Knowledge of human resource developing', a sub-entity such as 'E2: HR department' was defined. On the other hand, as shown in Figure 7(b), for APs decoupled from the others, such as 'A5: Retention rate of planning data of study' and 'A6: Comprehension of human interaction', we allocated these APs to a single sub-entity: 'E4: Div. of course planning 3'. As shown in Figure 7(c), for APs coupled from the others, such as 'A7: Collection rate of questionnaire' and 'A8: Capability of questionnaire analysis', we allocated these APs to a single sub-entity: 'E6: Div. of course planning 4'.

For the integration of the sub-entities, the view models were first converted into functional input-output models to determine dependencies among the PSS entities. Figure 8 shows an example of the functional input-output model converted from the view model for 'Quality of course planning'.

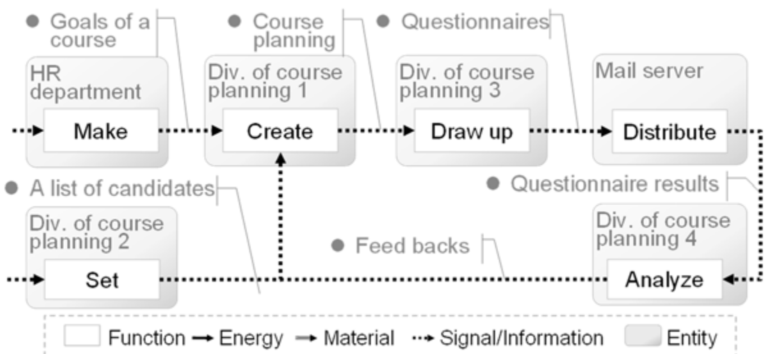


**Figure 8. Input-Output function representation for 'Quality of course planning'**

Next, for the specification of independency among the sub-entities, a DSM was developed according to functional interrelationships among the sub-entities.

According to the dependencies in the DSM, the clustering of off-diagonal elements was conducted. As a result, we specified four clusters of sub-entities as shown in Figures 9(a)-(d). Points in Figure 9 represent whether or not there are input-output relationships among the sub-entities.

| | | E1 | E2 | E3 | E9 | E13 | E14 | E7 | E8 | E12 | E10 | E11 | E6 | E5 | E4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Div. of course planning 1 | E1 | E1 | | | | | | | 1 | | 1 | | | | 1 |
| HR department | E2 | 1 | E2 | 1 | | | | | | | | | | | |
| Div. of course planning 2 | E3 | 1 | | E3 | | | | | | | | | | | |
| Div. of course planning 5 | E9 | | (a) | | E9 | | | 1 | 1 | | | | | | |
| Div. of system operation 3 | E13 | | | | | E13 | | | 1 | | | | | | |
| Back up server | E14 | | | | | | E14 | | | | | | | | |
| Div. of course management | E7 | | | | | | | E7 | | | | | | 1 | |
| Div. of system operation 1 | E8 | | | | 1 | | 1 | | E8 | | (c) | | | | |
| Contents data base | E12 | | | | | | (b) | | | E12 | | | | | |
| Div. of contents development | E10 | | | | | | | | | | E10 | 1 | | | |
| Div. of system operation 2 | E11 | | | | | | | | | 1 | | E11 | | (d) | |
| Div. of course planning 4 | E6 | 1 | | | | | | | | | | | E6 | | |
| Mail server | E5 | | | | | | | | | | | | 1 | E5 | |
| Div. of course planning 3 | E4 | | | | | | | | 1 | | | | | 1 | E4 |

**Figure 9. DSM clustering result of e-leaning service**

Based on the result of the integration, we finally determined the PSS modules. In this application, each cluster of sub-entities, as shown in Figures 9(a)-(d), was defined as a PSS module. Table 1 shows the sub-entities within each PSS module, and the functions and attributes of each sub-entity. The content development module takes responsibility for developing the e-learning contents based on the program of human resource development. The course management module operates courses based on the progress reports provided from the e-learning system. The contents database management module registers e-learning contents on the system. The course evaluation module takes responsibility for tasks relating to the evaluation of courses, such as conducting questionnaires.

**Table 1. Relationships among PSS modules, sub-entities, attribute parameters and function parameters**

| PSS modules | Sub-entities | | Attribute parameters | | Function parameters | |
|---|---|---|---|---|---|---|
| Content development | E1 | Div. of course planning 1 | A1 | Amount of course knowledge | F1 | Quality of contents |
| | | | A2 | Capability for developing others | | |
| | E2 | HR department | A3 | Knowledge of human resource developing | F2 | Compatibility with HR develop. |
| | E3 | Div. of course planning 2 | A11 | Amount of employee data | F3 | Completeness of candidates |
| | | | A4 | Understanding of course purpose | F4 | Suitability of target employees |
| Course management | E9 | Div. of course planning 5 | A17 | Capability of pursue a support efficient | F12 | Adequacy of schedule |
| | | | A10 | Frequency of updating of request data | F15 | Achievement rate of support of request |
| | E13 | Div. of system operation 3 | A19 | Capability of pursuit an effort | F20 | Stability of system * |
| | E14 | Back up server | A23 | Frequency of backup | F22 | Achievement rate of backup |
| | E7 | Div. of course management | A13 | Frequency of update of the study progress | F8 | Accuracy of progress management of study |
| | | | A14 | Collection rate of student data | F9 | Accuracy of attribution of reminder |
| | | | A15 | Capacity to analyze a study progress | | |
| | E8 | Div. of system operation 1 | A16 | Retention rate of a load data of system | F11 | Adequacy of operation planning |
| | | | A9 | Capability of pursue a system efficient | F21 | Planning of maintenance |
| | | | | | F12 | Adequacy of schedule * |
| | | | | | F20 | Stability of system * |
| Contents data base management | E12 | Contents data base | A22 | Retention rate of a data of e-contents | F19 | Efficiency of management of e-contents |
| | E10 | Div. of contents development | A20 | Retention rate of knowledge of e-contents | F16 | Achievement rate of e-contents |
| | | | A21 | Retention rate of text knowledge | F17 | Conversion rate of e-contents |
| | E11 | Div. of system operation 2 | A18 | Retention rate of registry knowledge | F18 | Achievement rate of server registry |
| Course evaluation | E6 | Div. of course planning 4 | A7 | Collection rate of questionnaire | F7 | Accuracy of questionnaire analysis |
| | | | A8 | Capability of questionnaire analysis | | |
| | E5 | Mail server | A12 | Retention rate of contact information | F14 | Achievement rate of distribution |
| | | | | | F10 | Achievement rate of assignment of reminder |
| | | | | | F6 | Completeness of distribution |
| | E4 | Div. of course planning 3 | A5 | Retention rate of planning data of study | F5 | Quality of questionnaires |
| | | | A6 | Comprehension of human interaction | F13 | Quality of announcement text |
| | | | | | F12 | Adequacy of schedule * |

* uncoupled functions

## 5. Discussion

In this application, the proposed method was applied to an e-learning service. As a result, the PSS entities were first decomposed into 14 sub-entities from the viewpoint of the Independence Axiom. Subsequently, these sub-entities were reintegrated into clusters. Based on the these clusters, we

determined the four PSS modules: ‘Content development’, ‘Course management’, ‘Contents database management’ and ‘Course evaluation’. Since each module maintains functional independency and minimizes the dependencies among them, changing a sub-entity can be independently conducted within each module, if the dependencies with the other modules are guaranteed. In the case of improving the function parameter ‘F1: Quality of contents’, for example, changing the sub-entity ‘E1: Div. of course planning 1’ can be conducted without considering the influences on other functions. In addition, coordination caused by changing this sub-entity can be carried out within the ‘Content development module’, if the dependencies with the other modules, such as the dependency between ‘E1: Div. of course planning 1’ and ‘E8: Div. of system operation 1’ is guaranteed.
In the following process of PSS development, the designers enabled the assigning of PSS modules to multiple stakeholders. Therefore, the determination of PSS modules are constrained by the current status of the stakeholders. In this application, for example, the PSS module ‘content development’ contains an attribute parameter ‘knowledge of human resourse developing’. A stakeholder who takes responsibility for this PSS module needs to have sufficient knowledge about human resource developing.

## 6. Conclusion

In order to address coordination complexities among PSS stakeholders, this paper proposes a concrete methodology for the determination of PSS modules. The application shows the four PSS modules that reduce coordination among the PSS entities in the case of improving the function parameters. Therefore, if these PSS modules are assigned to multiple stakeholders, it enables each stakeholder to provide flexibility in changing PSS functions to cope with dynamic changes, such as their resources, market demands, and changing customer requirements.
Future studies will include the consideration of constraints on the determination of the PSS modules.

**References**

*Browning, T.R., “Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions”, IEEE Transactions on Engineering Management, Vol.48, No.3, 2001, pp.292-306.*

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

*Fernandez, C.I.G., “Integration analysis of product architecture to support effective team co-location”, ME thesis, MIT, Cambridge, MA, 1998.*

*Meier, H., Völker, O., “Organizational Requirements by Offering Industrial Product-Service Systems”, Proceedings of the 42nd CIRP Conference on Manufacturing Systems, Grenoble, 2009, Electronic Resource.*

*Mont, O.K., “Clarifying the concept of product–service system”, Journal of Cleaner Production, Vol.10, 2002, pp. 237–245.*

*Shimomura, Y. and Tomiyama, T., “Service Modeling for Service Engineering”, IFIP International Federation for Information Processing, Vol.167, 2005, pp.31-38.*

*Suh, N.P., “The Principles of Design”, Oxford University Press, New York, 1990.*

*Tomiyama, T., “A Manufacturing Paradigm Toward the 21st Century”, Integrated Computer Aided Engineering, Vol. 4, 1997, pp. 159-178.*

*Ulrich, K., “The role of product architecture in the manufacturing firm”, Research Policy, Vol.24, 1995, pp.419-44.*

*Vargo, S.L. and Lusch, R.F., “Evolving to a New Dominant Logic for Marketing”, Journal of Marketing, Vol.68, No.1, 2004, pp.1-17.*

Dr. Koji Kimita
Tokyo Metropolitan University
Asahigaoka 6-6, Hino-shi
191-0065 Tokyo, Japan
Email: kimita-kouji@sd.tmu.ac.jp