INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED11
15 - 18 AUGUST 2011, TECHNICAL UNIVERSITY OF DENMARK

A METHOD FOR SELECTING BASE FUNCTIONS FOR FUNCTION BLENDING IN ODER TO DESIGN FUNCTIONS

Syo Sakaguchi, Akira Tsumaya, Eiko Yamamoto and Toshiharu Taura
Kobe University, Japan

ABSTRACT

This study aims to develop a method for supporting the designs of new functions by extending the conventional design processes in conceptual design. By focusing on concept blending that can create new concepts, we had previously developed a method of function blending in the design process. However, the selection of the functions (base functions) to be blended still remained an unsolved problem. In this paper, we propose a method for selecting base functions to design new functions. Design is often considered to be a problem-solving process. Our method for selecting the base functions has been developed by analysing the nature of the problem-solving process. In particular, we have addressed the antonymic relations between the verbs in the base functions, which play important roles.

Keywords: Conceptual design, function blending, problem solving, antonymy relation, base function

1 INTRODUCTION

In product design, it is necessary to design new and creative products continuously. In this study, we recognize that the essence of designing products with high novelty and creativity lies in answers to the question “What should we create?” We believe that “What” is the specification of the functions. Accordingly, in conceptual design, the focus should not only be on deriving novel mechanisms but also on creating novel functions.

In engineering design, the creativity of products significantly depends on the conceptual design. Conceptual design comprises two stages [1]:

1. Construction of a function structure by decomposition of the functions that satisfies the given specifications.
2. Devising of a mechanism for realizing the function on the basis of the constructed function structure.

Many researchers have proposed methods for supporting conceptual designs by constructing function structures that focus on devising the mechanism for realizing the function. For example, Gero has proposed the Function-Behavior-Structure model and a method for supporting designs using the model [2]. Stone and Wood have proposed functional basis that systematizes the terminologies used to describe the functions for product similarity computation [3]. Chakrabarti has proposed the SAPPhIRE model and system for solving design problems by analogy [4]. The function structures constructed by these methods can be regarded as tools to devise a mechanism for realizing the given specifications efficiently. On the other hand, “What should we decompose into the function structure?” can be another viewpoint. In other words, the construction of the function structure including the process of identifying the specifications (function) itself is expected to be an important issue. Thus, by focusing on “What should we create?”, we can extend the processes of constructing the function structure.

In cognitive science, concept blending has been considered to be an effective method for creating new concepts [5]. We had previously investigated “function blending in function dividing process” as a method for supporting design functions by integrating the notion of blending processes into function dividing processes (the details are explained later) [6,7]. We refer to the process to obtain lower-level functions in function structures from upper-level functions as a “function dividing process.” However, selecting base functions for function blending in order to construct a creative function structure remains an unsolved issue. For function blending in function dividing processes, we have systematized blending operations by focusing on the linguistic similarities between the words that had expressed functions in our previous study [6]. Moreover, in creative concept blending, the distance between the base concepts, which are the concepts to be blended, has been found to be a factor for generating concepts with high creativity [8]. Therefore, we also consider the similarity between base functions for the function blending operations. Based on the argument that designing is a problem-solving process, this paper reveals undiscovered relations between the base functions to be blended that can be used to devise new and creative function structures. Furthermore, a method for supporting the design of new functions by function blending including the process of selecting the base functions is demonstrated.

2 FUNCTION DIVIDING PROCESS MODEL

In this section, we explain the function dividing process model [7]. This model is used as a framework for the function blending model in the function dividing process, which is explained in the next section.

2.1 Function Dividing Process

First, we clarify the definition of function in this study—function is defined as "an entity's behavior that plays a special role." Function is described by using a subject (*S*), a verb (*V*), and object (*O*). The function dividing process is defined as a process for obtaining the representations for lower-level functions from upper-level functions. For a function *f* = (*S*, *V*, *O*), this process is defined as follows:

$$\begin{aligned} & m: f_{upper} \rightarrow F_{lower} \\ & f_{upper} = (S, V, O), F_{lower} = \{ f_i \mid f_i = (S_i, V_i, O_i) \} \\ & F_{upper} = \{ f_{upper} \} \end{aligned} \tag{1}$$

Here S_i, V_i, and O_i are the subject, verb, and object of a lower-level function, respectively. The relations for the subjects and verbs of the upper- and lower-level functions are as follows:

$$S = \{ S_i \mid S_i \, is\ one\ of\ \ components\ \ that\ realize\ \mathrm{S} \} \tag{2}$$

$$V = \{ V_i \mid V_i is\ one\ of\ \ movements\ \ that\ realize\ \mathrm{V} \} \tag{3}$$

For example, components such as "engine" and "tire" correspond to S_i when *S* is "car." Furthermore, movements such as "run" and "stop" correspond to V_i when *V* is "move." In other words, the subjects (components) in the lower-level functions realize the subjects in an upper-level function.

There are two types of processes for obtaining lower-level functions from an upper-level function—*decomposition-based dividing* that causes decomposing of the structure of the product and *causal-connection-based dividing* that depends on the causal relations of the physical phenomena involved among the components of the lower-level function.

2.2 Decomposition-based Dividing Process

In the decomposition-based dividing process, the subset of a subject $\{S_i\}$ of lower-level functions is the component of the subject *S* of upper-level functions; the subset of verbs $\{V_i\}$ of lower-level functions is the movement that realizes the verb *V* of upper-level functions. Further, the object of each lower-level function, O_i, is equal to that of each upper-level function *O*. This process can be formulated as follows:

$$\begin{aligned} & m_1 : f_{upper} \rightarrow F_{lower} \\ & F_{lower} = \{ f_i \mid f_i = (S_i, V_i, O_i), O_i = O \} \end{aligned} \tag{4}$$

Figure 1 shows an example of this dividing process. The lower-level functions "fan blows dust" and "vacuum bag holds dust" are obtained from the upper-level function "vacuum cleaner sucks up dust" by the decomposition-based dividing process. Here, "fan" and "vacuum bag" are components that realize "vacuum cleaner," while "blow" and "hold" are movements that realize "suck up." The object of both lower-level functions is "dust," which is identical to the object of the upper-level function.

2.3 Causal-connection-based Dividing Process

Two functions are said to be causally connected when the object of one function is equal to the subject of the adjacent function at the same level. In the causal-connection-based dividing process, the lower-level functions are connected by a chain of causal relations; the object of the terminal lower-level function O_n is equal to that of the upper-level function O. Moreover, the verb of the terminal lower-level function, V_n, is equal to that of the upper-level function, V. This process is formulated as follows:

$$\begin{aligned} & m_2 : f_{upper} \rightarrow F_{lower} \\ & F_{lower} = \{ f_i | f_i = (S_i, V_i, O_i), n = |F_{lower}|, \\ & S_1 \in S\ (1 \le i \le n-1), S_{i+1} = O_i \in S (1 \le i \le n-1), V_n = V, O_n = O \} \end{aligned} \tag{5}$$

For example, in Figure 1, the lower-level functions “motor rotates impeller” and “impeller blows dust” are obtained from the upper-level function “fan blows dust” by the causal-connection-based dividing process. In this case, “motor” and “impeller” are components that realize “fan,” and the object “impeller” of “motor rotates impeller” is identical to the subject of “impeller blows fun” because “motor rotates impeller” and “impeller blows dust” are connected causally. Moreover, the terminal function “impeller blows dust” and the upper-level function “fan blows dust” have the same verb and object.

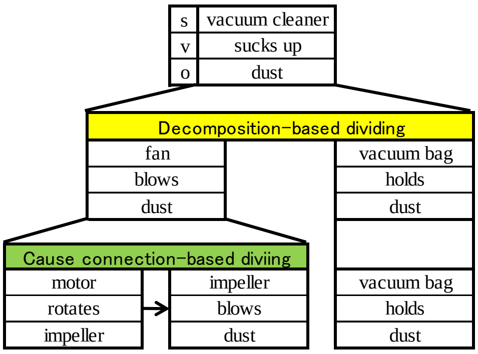


Figure 1. Example of function dividing

3 FUNCTION BLENDING IN FUNCTION DIVIDING PROCESS

We have developed the method of “function blending in function dividing process” to derive new function structures by adding function blending processes to function dividing processes [6]. We will now briefly introduce the function blending processes employed in this paper.

3.1 Function Blending in Function Dividing Process

“Function blending in function dividing process” is defined as processes that obtain a new set of lower-level functions from a set of the upper-level functions by the function blending operation; it is represented as follows:

$$m : F_{upper} \rightarrow F_{lower} \tag{6}$$

Here F_{upper} denotes the set of upper-level functions while F_{lower} denotes the set of lower-level functions. Function blending operations are classified into two types—“integration operation” that integrates two functions and “conversion operation” that replaces a function with another one. On the basis of these operations, the function blending in the function dividing process is also classified into “function blending in function dividing process by integration” and “function blending in function dividing process by conversion.”

3.2 Word Match

In order to conduct integration and conversion operations, it is necessary to determine the similarity of two functions. Therefore, we introduce “abstraction-concretion operation” and “word match.” The “abstraction-concretion operation” consists of an abstraction operation and a concretion operation. The abstraction operation is an operation to obtain the abstract word from the given word by extracting its common and/or essential nature. Concretion operation is the operation to obtain a more concrete word from the given word by clarifying the content of the given word. A set of words, $P(w)$, obtained by the abstraction-concretion operation on a word w is defined as follows:

$$P(w)=\{w_i | w_i \, \textit{is a word obtained by abstracting or concreting } \, w\} \tag{7}$$

Here, suppose w_1 is included in $P(w_2)$, that is, $w_1 \in P(w_2)$. In this case, the relation between w_1 and w_2 is defined as the "superordinate-subordinate relation." This relation is well known as a hierarchal semantic relation. Therefore, we can find such relations in existing dictionaries and thesauri such as WordNet [9].

We also define a situation in which two words are similar through word abstraction-concretion operations such as "word match." A set $I(w)$ of words that matches a word w is defined as follows:

$$I(w)=\{w_i | w_i = w \vee w_i \in P(w) \vee 0 < |P(w) \cap P(w_i)| \} \tag{8}$$

In addition, a set $M(w_1, w_2)$ of words when two words match each other is defined as follows:

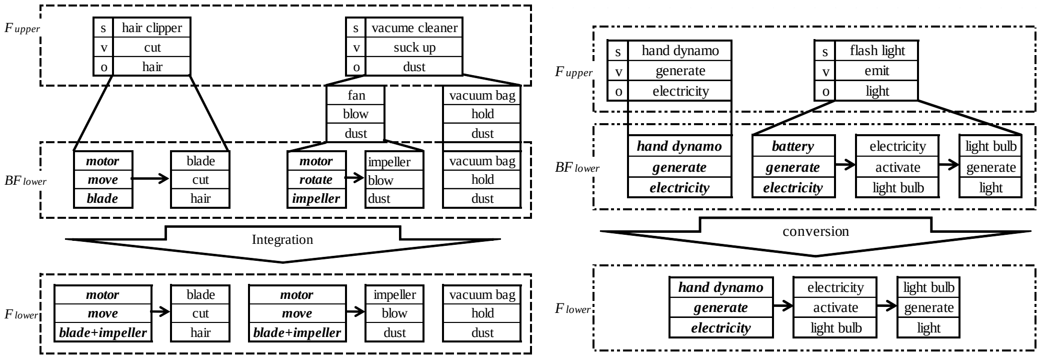
$$M(w_1,w_2)=\{w_i | w_i = w_1 \in P(w_2) \vee w_i = w_2 \in P(w_1) \vee w_i \in P(w_1) \cap P(w_2) \vee w_i = w_1 = w_2\} \tag{9}$$

3.3 Function Blending in Function Dividing Process by Integration

When the verbs and objects of two functions correspond to word match, the operation of integrating two functions by joining their subjects is defined as "function blending in function dividing process by subject integration." Similarly, when along with the verbs or objects, the subjects of two functions also correspond to word match, the operation of integrating two functions by joining their objects or verbs is defined as "function blending in function dividing process by verb or object integration." The operation of joining two words is described as the "+" operator. Using "+" operator, these operations can be formulated as follows:

$$\begin{aligned}
&m_3 : F_{upper} \rightarrow F_{lower} \\
&f_1 \in BF_{lower},\, f_1 = (S_1, V_1, O_1),\, f_2 \in BF_{lower},\, f_2 = (S_2, V_2, O_2) \\
&when V \in M(V_1, V_2), O \in M(O_1, O_2) \\
&f_{new} = (S_1 + S_2, V, O), F_{lower} = \{ f_i \mid f_i \in (BF_{lower} - \{f_1\} - \{f_2\}) \cup \{f_{new}\}\}
\end{aligned} \tag{10}$$

An example of function blending in function dividing process by integration is shown in Figure 2(a); the function blending of electric hair clipper and vacuum cleaner is shown here. In this case of "motor moves blade" and "motor rotates impeller," the subjects and verbs correspond to word match. Therefore, their objects "blade" and "impeller" are integrated with the "+" operator.



(a) by object integration (b) by subject conversion

Figure 2. Example of function blending in function dividing process

3.4 Function Blending in Function Dividing by Conversion

When the verbs and objects of two functions correspond to word match, the operation of replacing a function with another is defined as the "subject conversion operation." The operation of function blending in function dividing process by subject conversions is formulated as follows:

$$\begin{aligned}&m_4 : F_{upper} \to F_{lower}\\&f_1 \in BF_{lower},\ f_1 = (S_1,V_1,O_1),\ f_2 = (S_2,V_2,O_2)\\&when V_1 \in I(V_2), O_1 \in I(O_2)\\&f_{new} = (S_2,V_1,O_1), F_{lower} = \{ f_i \mid f_i \in (BF_{lower} - \{ f_1 \}) \cup \{ f_{new} \} \}\end{aligned} \qquad (11)$$

An example of function blending in function dividing by conversion is “flashlight with hand generator,” as shown in Figure 2(b). In this case, in the case of a function “hand dynamo generates electricity” of the product “hand generator” and a function “battery generates electricity” of the product “flashlight,” their verbs correspond to word match while their objects also correspond to word match. Therefore, the subject “battery” can be replaced with “hand dynamo.”

4 PROBLEM SOLVING

We will now discuss the notion of “problem” and attempt to classify problem-solving processes.

4.1 Problem-Solving Processes

Problem-solving processes have been discussed in various fields. In the field of engineering design, design has also been discussed in the framework of problem solving where specifications are the problem and design solutions are the solutions to the problem.

In terms of the difference between the present state and goal, problem-solving processes can be classified into three types. The first type is the “spontaneous type” that involves a situation in which the present state is clearly different from the goal. The second type is the “problem searching type” that contains a situation wherein the difference between the present state and goal is recognized by an analysis of the object of problem solving. The third type “problem setting type” involves situations wherein the goal is set by the problem solver. In engineering design, product specifications are usually given as the goal. Accordingly, traditional “How should we realize the given function?” type engineering design can be understood as the “spontaneous type” or “problem searching type.” On the contrary, in this paper, we expand such designs to the creation (designing) of the specification itself (function). In the expansion, we also deal with the “problem setting type.”

On the other hand, problem-solving processes can also be classified into two styles by focusing on their aims. One style aims at “reducing or resolving complaints,” while the other aims at “increasing or generating satisfactions.” We regard the reasons for complaints as “functions that can be weak points” and the reasons for satisfaction as “functions that can be strong points.” In such a case, we define the aim of problem solving in function blending as “resolving weak points or enhancing strong points.”

4.2 Classification of Function Blending from Viewpoint of Problem Solving

The methods for the abovementioned function blending can be classified by the type of function blending operations from the viewpoint of the problem-solving styles, as shown in Table 1. “Weak-point compensation” achieves the resolution of weak points by integrating a function in the original function, while “weak-point elimination” achieves the resolution of weak points by the replacement of a lower-level function with another lower-level function. “Strong-point extension” achieves a more advantageous function structure by the integration of functions, while “strong-point differentiation” achieves a more advantageous function structure by the conversion of functions. Hereafter, we will refer to the two functions to be blended as “base functions.”

Table 1. Classification of function blending from viewpoint of problem-solving styles

	Integration	Conversion
Weak-point Resolving	Weak-point Compensation	Weak-point Elimination
Strong-point Enhancing	Strong-point Extension	Strong-point Differentiation

5 SEARCHING BASE FUNCTIONS

In this section, we derive a method to search the base functions to be blended from the viewpoints of the type of function blending operations and problem-solving style.

5.1 Base Functions for Problem Solving

From the viewpoint of problem solving, the base functions can be classified into one that comprises the problem and the other the solution (hereafter, the problem function and solution function, respectively). Accordingly, we call the product that manifests the problem and solution functions as the problem and solution products, respectively. In our subsequent discussions, *PF* represents the set of functions contained in the function structure of the problem product. The problem function *pf* = (*pS*, *pV*, *pO*), which is the component function of *PF*, comprises the following set of functions—the upper-level functions of the problem function pf_{upper} = (pS_{upper}, pV_{upper}, pO_{upper}) and a set of lower-level functions of the problem function pF_{lower}. Similarly, *SF* is a set of functions contained in the function structure of the solution product; *sf* = (*sS*, *sV*, *sO*), the solution function; sf_{upper}, the upper-level function of the solution function; and sF_{lower}, a set of the lower-level functions of the solution function.

5.2 Weak-point Compensation

When a problem function involves the weak point, the weak point can be compensated by integrating a solution function into the problem function. For example, the function of “cloth washer” can be achieved by compensating the problem function of “wash cloth” (the cloth is wet when it is only washed) by integrating the solution function of “dry cloth” into the problem function. Similarly, the function of “car” can be achieved by compensating the problem function of “move wheel” (the wheel cannot be stopped when it is only moved) by integrating the solution function of “brake wheel” into the problem function. We collected pairs of problem and solution functions and analyzed their relations from the linguistic viewpoint. We found that the relation can be explained by using the linguistic antonymy relation between the verbs contained in the problem and solution functions under the condition that the verbs have the same object.

Antonymy relations are included in various existing dictionaries and thesauri such as WordNet [9]. However, we cannot determine the antonymy relations between “dry” and “wash” and between “brake” and “move” directly in these dictionaries or thesauri. The antonymy relation should be extended for usage in the situation of the blending function. Here, when the words w_1 and w_2 are present in the antonymy relation, the relation is represented as follows:

$$w_1 = w_2' \qquad (12)$$

We then formulate the extended antonymy relation by using the definition of word matches as follows. When w_1 and w_2 are in the extended antonymy relation, they can be said to be in the situation of “antonymic word match.”

$$w_1 = I(I'(w_2)) \qquad (13)$$

Let us assume that w_1 is “wash” and w_2 is “dry.” Since “wash” and “wet” correspond to word match and “dry” and “wet” are in the antonymy relation, “wash” and “dry” are determined to be in the situation of antonymic word match. Similarly, “move” and “brake” are determined to be in the situation of antonymic word match since “stay” and “brake” corresponds to word match and “move” and “stay” have the antonymy relation.

Further, we denote the number of candidates of places where two function structures F_1 and F_2 are integrated as follows:

$$Int(F_1, F_2) \qquad (14)$$

Based on the above considerations, we can determine whether the problem and solution products with F_1 and F_2 can be integrated. From the abovementioned discussions, the following condition is derived: the verbs of the solution and problem functions are in the situation of antonymic word match, while their objects correspond to word match; in addition, the problem and solution products can be integrated. This condition can be formulated as follows:

$$sV \in I(I'(pV)) \wedge sO \in I(pO) \wedge Int(SF, PF) \geq 1 \qquad (15)$$

Function blending by weak-point compensation is shown in Figure 3. When *PF* and *SF* are the function structures of the problem and solution products, respectively, (upper left and right figures, respectively, in Figure 3), the function structure constructed by function blending is shown in the lower figure in Figure 3. The function structure includes both problem and solution functions. The weak point of the problem function is compensated by the solution function. For example, “electric hair clipper with vacuum cleaner” shown in Figure 2 can be explained by using this framework. In this case, the problem product is “electric hair clipper,” the problem function is “blade cuts hair,” the solution product is “vacuum cleaner,” and the solution function is “vacuum bag holds dust.” The verb “cut” of the problem function and the verb “brake” correspond to word match, and the verb “hold” of the solution function the verb “keep” also corresponds to word match. In addition, the antonym relation exists between “brake” and “keep.” When these relations exist, “cut” and “hold” are in the situation of antonymy word match. The places where the functions are integrated are the “motor moves blade” of electric hair clipper and “motor rotates impeller” of vacuum cleaner.

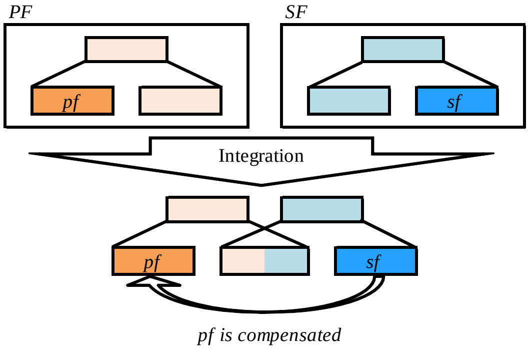


Figure 3. Function blending by weak-point compensation

5.3 Weak-point Elimination

When problem functions involve weak points, they can be eliminated by replacing the lower-level functions that lead to weak points with others that do not contain weak points. However, a normal conversion operation replaces the problem function with a function whose verb and object correspond to the word match of the problem function. Therefore, the problem product operated by the normal conversion operation must include a function that is similar to the problem function. Based on these considerations, a conversion operation to the upper-level function of the problem function is conducted in this study. The conditions for weak-point elimination are determined as follows. A solution function can be replaced with the upper-level function of the problem function, and the lower-level functions of the solution function do not include the problem function. These conditions are formulated as follows.

$$sV \in I(pV_{upper}) \wedge sO \in I(pO_{upper}) \wedge pf \notin sF_{lower} \tag{16}$$

The weak-point elimination is shown in Figure 4. The image of the function structure that is constructed by function blending is shown in the lower figure in Figure 4. The constructed function structure does not include the problem function, and thus, the weak point has been eliminated. For example, let us consider “bicycle with timing belt” as the product whose problem is expected to be solved. In this case, the problem product is “bicycle” and the problem function is “chain cuts object.” This problem function is found to be similar to the function structure of chainsaws. The upper-level function of the problem function is “power transmission mechanism rotates object.” On the other hand, the solution product “power transmission mechanism with timing belt” involves the function structure of the solution function “power transmission mechanism with timing belt rotates object.” Therefore, the weak point of bicycle is resolved by replacing the problem function with the solution function.

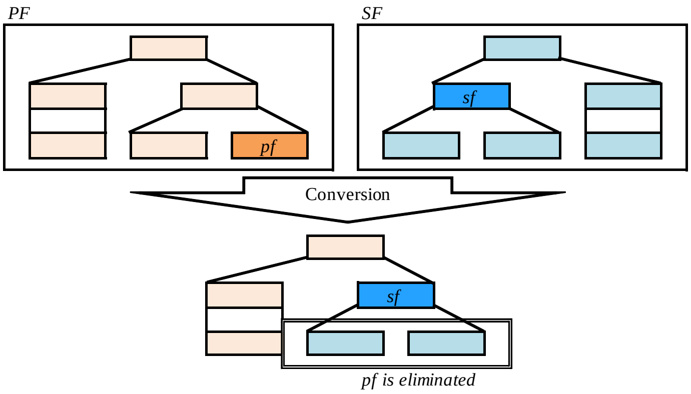


Figure 4. Function blending by weak-point elimination

5.4 Strong-point Extension

When the problem function involves a strong point, the strong point can be extended by integrating a function that is similar to the problem function and is included in the problem function with a different function structure. The conditions for strong-point extensions are determined as follows. The solution function can be integrated with the problem function, and the problem and solution functions belong to different function structures. The conditions are formulated as follows.

$$Int(sf, pf) = 1 \wedge sf_{upper} \neq pf_{upper} \qquad (17)$$

Function blending by strong-point extension is shown in Figure 5. A new function structure is constructed by blending a problem function and a solution function. For example, the product "convection microwave" can be explained as this process. In this case, the problem product is "microwave oven" and the problem function is "heating mechanism heats food," while the solution product is "convection oven" and the solution function is "baking mechanism bakes food."

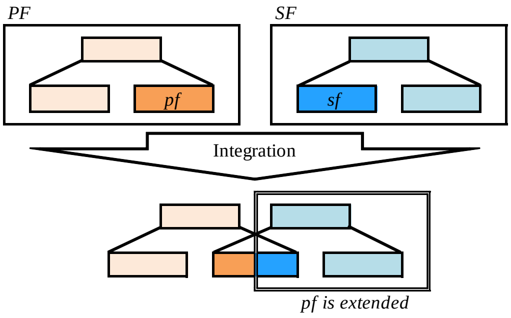


Figure 5. Function blending by strong-point extension

5.5 Strong-point Differentiation

When a problem function involves a strong point, the strong point can be differentiated by converting the subject of a problem function into the subject of a solution function that contains different lower-level functions from the problem function. The conditions for strong-point differentiation are determined as follows. The subject of the solution function can be converted into the subject of the problem function, and the solution product has different lower-level functions from the problem function. These conditions are formulated as follows:

$$sV \in I(pV) \wedge sO \in I(pO) \wedge sf_{lower} \neq pf_{lower} \qquad (18)$$

The blending function by strong-point differentiation is shown in Figure 6. The new function structure is obtained by replacing the problem function and its lower-level functions with the solution function and its lower-level functions, respectively, so that the strong-point is differentiated. For example, the product "flash light with hand generator" can be explained as a result of this process. Here, the problem product is "flash light" and the problem function is "battery generates electricity," while the solution product is "hand generator" and the solution function is "hand dynamo generates electricity."

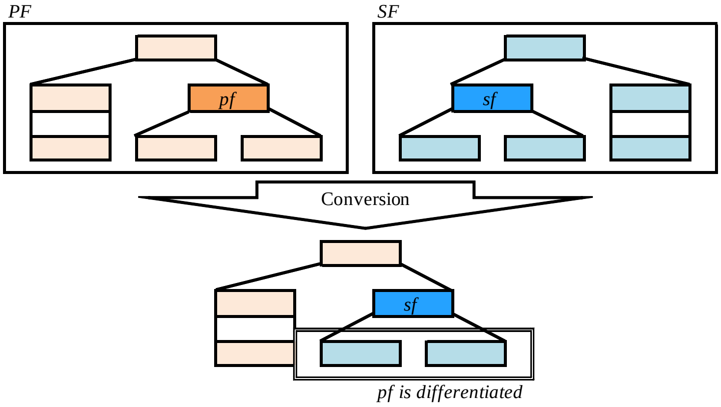


Figure 6. Function blending by strong-point differentiation

6 DEVELOPMENT OF SYSTEM FOR SELECTION OF BASE FUNCTIONS AND EXPERIMENTAL VERIFICATION

A system for the selection of base functions is developed. In this section, the outline of this system and an experiment to verify its validity are described.

6.1 System Outline

The selection of the base function method implemented on the developed system can be conducted by following a four-step procedure as follows:

1. Problem product selection:
   A problem product is selected by the designer from the product database.
2. Problem function selection:
   A function structure of the problem product selected in Step 1 is provided by the system, and a problem function is selected by the designer from the function structure.
3. Solution products searching:
   Pairs of solution products and solution functions for the selected problem product and the selected problem function are searched in the product database using the conditions discussed in section 5. The extracted pairs are shown by the system.
4. Solution product selection:
   From the solution products shown by the system, the designer selects a solution product for the function blending.

Figure 7 is a process flow of the system. The numbers in this figure correspond to the number of steps. This system searches the solution products of all the four problem-solving methods shown in Table 1.

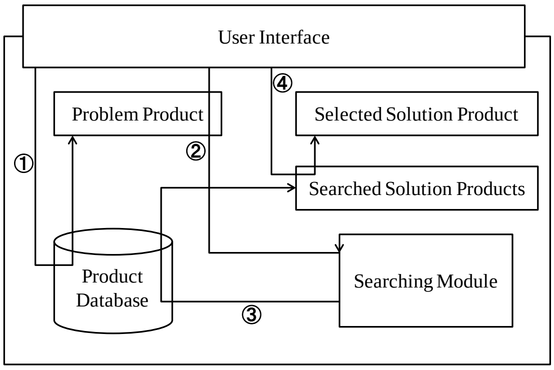


Figure 7. Process flow of system for selecting base functions

6.2 Verification

In order to conduct an experiment to verify the validity of the proposed method, we prepared a product database that included the function structures of 37 products, as shown in Table 2. Table 3 shows case examples of the existing products that can be explained as a process of function blending. Among them, we selected “electric hair clipper” as the problem product and “blade cut hair” as the problem function: the function structure of the product is shown in Figure 2(a). Table 4 shows the extracted pairs of the solution product and solution function by searching; those are 2 pairs for weak-point compensation, 2 pairs for weak-point elimination, 19 pairs for strong-point extension, and 2 pairs for strong-point differentiation.

The contents in Table 4 are verified as follows. For weak-point compensation, “vacuum cleaner” is searched as a solution product. In this case, the solution product changes the structure of the electric hair clipper by integrating the solution and problem product. For weak-point compensation, the solution function “filter holds dust” and problem function have the following relation. “Hold” and “cut” have the antonymic word match situation explained in 5.2, while “dust” and “hair” have the word match situation. In addition, for integrating the solution and problem product, “motor rotates impeller” and “motor moves blade” have the following relation. “Motor” and “motor” as well as “rotate” and “move” have the word match situation. The blended structure constructs an existing product “hair clipper with vacuum cleaner.” This confirmation of the feasibility of the existing product indicates that this system has the possibility of the selection of base functions for creative function blending.

For weak-point elimination and strong-point differentiation, “chainsaw” is searched as the solution product. Here, the solution function “chainsaw cuts object” can change the structure of electric hair clipper by conversion to problem function. The solution function “chainsaw cuts object” and the problem function have the following relation. “Cut” and “cut” as well as “object” and “hair” have the word match situation. Therefore, the solution and problem functions can be converted. By this conversion, the electric hair clipper subsumes a new function “chainsaw cuts object” that represents a different mechanism from “blade cut hair.” This function blending differentiates the problem function by eliminating weak points (for instance, electric hair clipper is stopped because the hair gets caught in the blade).

For strong-point extension, “hair iron” is searched as the solution product. In this case, the solution product can change the structure of the electric hair clipper by integrating the solution and problem function. The solution function “rod heats hair” and problem function have following relationship. “Heat” and “cut” and “hair” and “hair” are both in the situation of word match. Therefore, the solution and problem functions can be integrated. From this integration, the blended structure has a function “blade + rod change hair” (“change” is an abstract verb for “cut” and “heat.”). This function structure becomes a new structure that can set and cut hair. However, for example, the searched solution product “gasoline engine” cannot be considered as a product that extends strong points by function blending. For strong-point enhancing, the solver must set a goal for problem solving. Accordingly, for searching the solution function for strong-point extensions, the proposed method must also have a process of setting a goal for problem solving.

Table 2. Product database for selecting base functions

No.	Name	No.	Name	No.	Name
1	oven	2	steam cooker	3	microwave oven
4	flash light	5	gasoline engine	6	hand dynamo
7	hair iron	8	electric hot plate	9	heat gun
10	electric fan	11	steam humidifier	12	electric water boiler
13	clothes washer	14	cloth dryer	15	bicycle
16	car	17	chainsaw	18	blender
19	electric motor	20	dish washer	21	dish dryer
22	printer	23	image scanner	24	television
25	video	26	digital camera	27	projector
28	cell phone	29	air purifier	30	vacuum cleaner
31	dish drainer	32	hair dryer	33	razor
34	electric hair clipper	35	lighter	36	musical box
37	power transmission mechanism (with toothed belt)				

Table 3. Case examples of function blending

Method	Blended Product	Problem Product	Solution Product
Weak-point compenSation	automatic washing machine	cloth washer	spin dryer
	dish washer with drying elements	dish washer	dish dryer
	electric hair clipper with vacuum cleaner	electric hair clipper	vacuum cleaner
Weak-Point eliminaTion	bicycle with toothed belt	bicycle	power transmission mechanism with toothed belt
	curling dryer	hair iron	hair dryer
	motorcycle	bicycle	gasoline-driven car
Strong-point extension	TV/VCR combination	television	video player
	electric fan with air purifier dish dryer	electric fan dish drainer	air purifier hair dryer
	digital camera with printer digital camera with projector	digital camera digital camera	printer projector
	vacuum cleaner with air purifier convection microwave	vacuum cleaner microwave oven	air purifier convection oven
Strong-point differentiation	electric car	gasoline-driven car	electric motor
	steam oven	convection oven	steam cooker
	portable video player	television	video player
	flash light with generator	flash light	hand generator

Table 4. Pairs of solution product and solution function for problem function "blade cuts hair" of electric hair clipper.

Problem-solving method	Solution product	Solution function
Weak-point compensation	vacuum cleaner	filter holds dust
	blender	container contains food
Weak-point elimination	chainsaw	chainsaw chain cuts object
	chainsaw	chainsaw cuts object
Strong-point extension	gasoline engine	gasoline engine turns object
	gasoline engine	crankshaft turns object
	hair iron	hair iron sets hair
	hair iron	rod heats hair
	hair iron	plate heats hair
	heat gun	heat gun heats object
	heat gun	hot air heats object
	electric fan	motor turns fan
	clothes washer	clothes washer cleans cloth
	cloth dryer	cloth dryer dries cloth
	cloth dryer	drying mechanism dries cloth
	cloth dryer	drying tub dries cloth
	chainsaw	chainsaw cuts object
	chainsaw	chainsaw chain cuts object
	dish washer	dish washer cleans dish
	dish dryer	dish dryer dries dish
	dish dryer	drying mechanism dries dish
	dish dryer	hot air dries dish
	vacuum cleaner	motor turns fan
Strong-point differentiation	chainsaw	chainsaw cuts object
	chainsaw	chainsaw chain cuts object

6.3 Discussion

The proposed method has shown feasibility for base function selections for creative function blending from the viewpoint of problem-solving processes. Function blending operations are not just “unite” or “replace” functions. The experimental results indicate that the proposed method can create “new” functions that are different from either of the base functions. On the other hand, the argument that the paradigm of an analytical view that is in accordance with the viewpoint of a problem-solving process is not applicable to post-industrial society [10] is being widely accepted. Therefore, it will be necessary to construct an extended method for selecting base functions for function blending, which can be applicable to post-industrial society paradigms. We believe that designs in post-industrial societies can be clarified by comparing problem-solving processes and also believe that the results of the proposed method lead to the development of an extended function blending method.

7 CONCLUSION

We have developed a method for selecting base functions to be blended for designing functions. By analyzing the nature of the problem-solving process, we have classified the function blending process into four types on the basis of the type of function blending operations. In particular, we have pointed out that the antonymic relation plays an important role in the relation between the functions to be blended. Subsequently, we have formulated four types of base function selections. Finally, we have constructed a system for selecting the base functions. In the verification experiments of the system, we were able to show that some existing products could be explained as a process of blending the functions that are searched using the system. This result suggests that our proposed method can support the design of a new creative function.

REFERENCES

[1] Pahl G. and Beitz W. *Engineering Design-Systematic Approach, 2nd ed.*, 1996 (Springer-Verlag, Berlin).

[2] Gero J. S. Design Prototypes, A Knowledge Representation Schema for Design, *AI Magazine*, 1990, 11(4), pp.26-36.

[3] Stone R. B. and Wood K. L. Development of a Functional Basis for Design, *Journal of Mechanical Design*, 2000, 122(4), pp.359-370.

[4] Chakrabarti A., Sarkar P., Leelavathamma B. and Nataraju B. S. A Functional Representation for Aiding Biomimetic and Artificial Inspiration of New Ideas, *Artificial Intelligent Engineering Design, Analysis Manufacturing*, 2005, 19(2), pp.113-132.

[5] Nagai Y., Taura T. and Mukai T. Concept Blending and Dissimilarity: Factors for Creative Concept Generation Process, *Design Studies*, 2009, 30(6), pp.648-675.

[6] Park Y., Ohashi S., Yamamoto E. and Taura T. Design of Functions by Function Blending, In *Proc. International Conference on Engineering Design, ICED11*, Copenhagen, August 2011.

[7] Yamamoto, E., Taura, T., Ohashi, S. and Yamamoto, M. A Method for Function Dividing in Conceptual Design by Focusing on Linguistic Hierarchal Relations, *Journal of Computing and Information Science in Engineering*, 2010, 10(3), 031004.

[8] Taura T., Nagai Y. and Tanaka S. Design Space Blending - A Key for Creative Design, In *Proc. International Conference on Engineering Design, ICED’05,* Melbourne, August 2005, on CD-ROM.

[9] Fellbaum C. *WordNet: An Electronic Lexical Database,* 1998 (MIT Press, Cambridge, MA).

[10] Taura T. and Nagai Y. Discussion on Direction of Design Creativity Research (Part 1) - New Definition of Design and Creativity: Beyond the Problem-Solving Paradigm, In *Design Creativity 2010,* pp.3-8 (Springer-Verlag, London).

Contact: Toshiharu Taura
Kobe University
Organization of Advanced Science and Technology
1-1, Rokkodai, Nada 657-8501, Kobe, Japan
Phone/Fax: Int +81-78-803-6503
E-mail: taura@kobe-u.ac.jp

Toshiharu Taura is a vice-dean and professor in the Organization of Advanced Science and Technology (and professor in the Department of Mechanical Engineering) at Kobe University. He is currently working on several research themes that focus on the creative thought process of both engineering and industrial design, including the interdisciplinary aspects of design sciences.

Syo Sakaguchi is a graduate student in the Department of Mechanical Engineering at Kobe University. He is studying conceptual design by focusing on functional operations.

Akira Tsumaya is an associate professor in the Department of Mechanical Engineering at Kobe University. He is interested in knowledge and methodology for conceptual design stages, especially how to model design rationale and how to reuse design knowledge.

Eiko Yamamoto has conducted this study as an associate professor (lecturer) in Department of Mechanical Engineering at Kobe University. Her interests include concept design supporting system, impression analysis, information extraction, and knowledge acquisition based on natural language processing. She is currently working for Gifu Shotoku Gakuen University as an associate professor.