INTERNATIONAL CONFERENCE ON ENGUNEERING DESIGN

ICED 03 STOCKHOLM, AUGUST 19-21, 2003

Similarity Conservation: A key to forming the function space

Toshiharu Taura and Shusuke Tobita

Abstract

Although the importance and necessity of the function decomposition process is accepted in both industry and academia, its methodology has not been thoroughly clarified. In particular, the space-forming process for the function decomposition has not been clarified. In this study, we pay attention to the similarity conservation between the space for the required function and the space for the decomposed function, and propose a hypothesis for forming a suitable space to find a design solution efficiently. A computer system is implemented and it is shown that the similarity conservation is a key in the space-forming process.

Keywords: Function space, Similarity of space, Design process, Function decomposition, Conceptual design

1. Introduction

In the early stage of the design process, the required functions are usually decomposed into some partial functions. Although this decomposition process is not always necessary to find design solutions, it is well known that it is useful in the design process. Not only its importance has been pointed out in an empirical study[1], but its rationale has been analyzed in a theoretical study[2].

Although the importance and necessity of the function decomposition process is accepted in both industry and academia, its methodology has not been thoroughly clarified. Considering that the function decomposition process is divided into the space-forming process and the design solution searching process in the formed space, the former process in particular has not been clarified. Suh[3] concluded that the required function should be decomposed independently and briefly, however the space-forming mechanism in the function decomposing process in not analyzed. In this study, the space-forming process in which the function is decomposed is analyzed.

2. Method

2.1 Basic idea

In our previous study, we proposed the following hypothesis[4].

Hypothesis 1 " In function decomposition, elements which are near each other in the space for the required functions are mapped onto elements which are near each other in the space for the decomposed functions in some cases."

This hypothesis means that between two machines, if the function structures described using the decomposed functions are near each other, they manifest similar functions as a whole, and the inverse is also valid. This hypothesis is explained using the example in Figure 1. First, let us define the following terms.

Def.1 A **Total function** is defined as a function which an object (machine) manifests as a whole.

Def.2 A **Partial function** is defined as a function which a component of an object (machine) manifests.

Def.3 A **Partial function structure** of an object (machine) is defined as a subset of the partial functions manifested by the object (machine).

Here, the total function can not be determined by the partial function. The total function becomes clear through calculating each component's attribute.

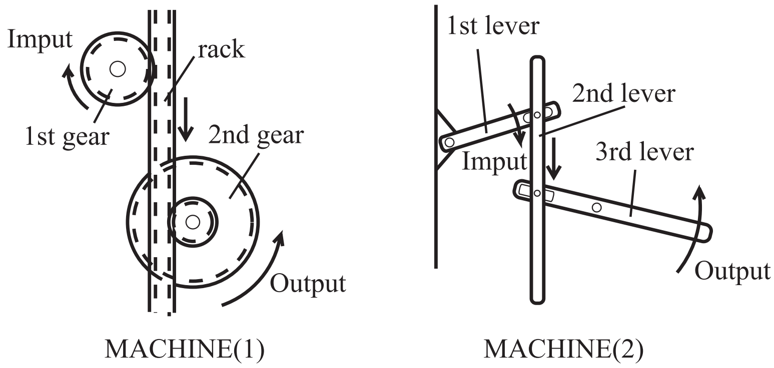

Figure 1. An example of function similarity between machines

Let us consider the relation between a total function and a partial function structure in the example in Figure 1. First, let us examine the total function of these machines. Both of these machines are found to manifest the same total function, such as "Input is Rotational Behavior", "Output is Rotational Behavior" and "The Speed of Behavior is Amplified". Next, let us examine the partial function structure of these two machines. MACHINE(1) is understood to be composed of some components, a 1st gear, a rack, and a 2nd gear. The 1st gear has functions such as "Input is Rotational Behavior", "Output is Straight-Line Behavior" and "The Speed of Behavior is Constant", the rack has functions such as "Input is Straight-Line Behavior", “Output is Straight-Line Behavior” and “The Speed of Behavior is Constant”, and the 2nd gear has functions such as “Input is Straight-Line Behavior”, “Output is Rotational Behavior” and “The Speed of Behavior is Amplified”. In the same way, the partial function structure of MACHINE(2) is as follows. The 1st lever has functions such as “Input is Rotational Behavior”, “Output is Straight-Line Behavior” and “The Speed of Behavior is Constant”, and 2nd lever has functions such as “Input is Straight-Line Behavior”, “Output is Straight-Line Behavior” and “The Speed of Behavior is Constant”, and the 3rd lever has functions such as “Input is Straight-Line Behavior”, “Output is Rotational Behavior” and “The Speed of Behavior is Amplified”. Here, one will notice that both partial function structures are the same, although they are composed of different mechanical components. As a result, between two machines which manifest the same total function, but are composed of different mechanical components, similarity in their partial function structure is obtained.

In our previous study, we simulated the function decomposition process and showed that design solutions can be found efficiently by adopting this hypothesis. However, this simulation is operated in the spaces defined in an ad hoc way. That is, the viewpoints for recognizing the partial function are pre-defined. Here, the question occurs “How is the function space for the partial function formed?”

In the design process, the function decomposition process proceeds from the total function to the partial function. Although the space for the total function can be formed on the basis of the fact that the design specification is described using the total function, the space for the partial function cannot be fixed, because we cannot find a rational criterion to evaluate the space.

One can assume that the space for the partial function is formed so that the design solution can be found efficiently. However, the following question occurs “How can the designer find the space for searching for the design solution efficiently, before they begin to search?”

In order to discuss this problem scientifically, we extend our previous discussion. That is, the similarity conservation between the space for total function and the space for partial function structures is a key to form the function space. Let consider this idea using the examples in Figure 1 again. If we pay attention to the partial functions from another viewpoint, for example, color, we cannot find any similarity between these two machines and it may be easily assumed that this viewpoint is not useful for finding a design solution. This discussion suggests that the similarity conservation between two spaces is strongly related to the space forming process for design solution searching efficiency. In other words, Hypothesis 1 is valid only when the design solution can be found efficiently. Therefore, we propose the following hypothesis.

Hypothesis 2 “ In the function decomposition process for finding a design solution efficiently, the space for the partial function is formed so that elements which are near each other in the space for the total functions are mapped onto elements which are near each other in the space for the partial functions.”

2.2 Mapping criterion

In this study, we define the criterion for evaluating the degree of similarity conservation between the space for the total functions and the space for the partial functions (Figure 2). By applying this criterion, we can find a more suitable space for the function decomposition process.

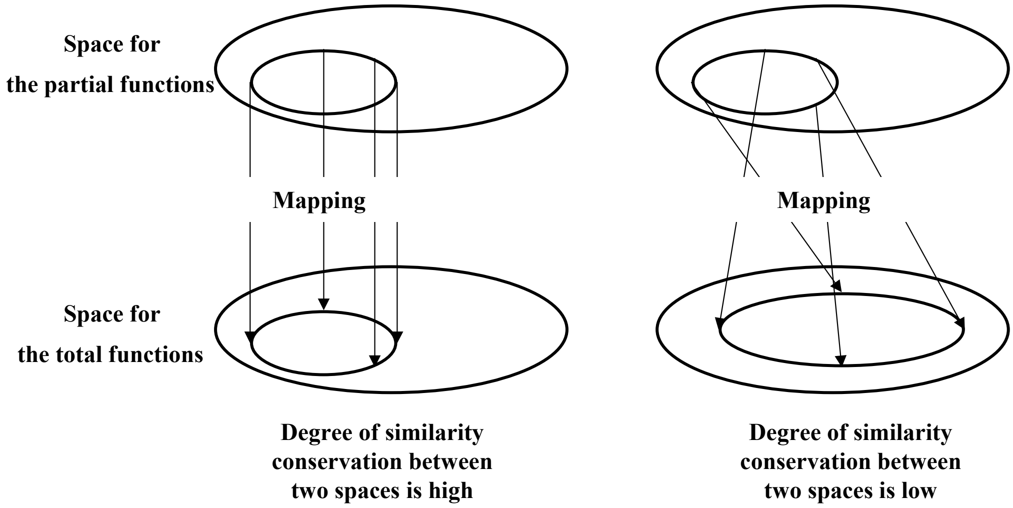

Figure 2. The difference of the degree of similarity conservation between two spaces

3. Ways of searching

A computer system is implemented and the above method is simulated to investigate how the degree of similarity conservation between Total Function Space (TFS) and Partial Function Space (PFS) is related to searching for a required machine efficiency. In this simulation, we use one ways of searching a required machine. In this study, it is called "Gradually Approaching Search".

3.1 The definition of the function space

In this study, the function spaces are defined as follows.

Def.4 **Total function space** (TFS) is defined as a space whose elements are objects (machines), and its classes are the objects' total functions.

Def.5 **Partial function space** (PFS) is defined as a space whose elements are objects (machines), and its classes are generated by classifying the objects from the viewpoints of their components' functions.

Def 6 The **difference between two objects** (machines), which expresses the distance between two elements in one space, is defined as the following. Consider two elements in the total function space, s1 and s2, which express the total function of the two objects (machine). They share A number of classes and they do not share B number of classes. The following formula can be obtained as the difference between the two objects (machines).

$$d = \frac{B}{A+B} \quad (1)$$

In PFS, the order of the classes (partial function) is also considered.

Def 7 The **degree of similarity conversion** between TFS and PFS is defined as follows. When distances between the **standard element** and all other elements in PFS are $dp_n (n = 1,2,3,\cdots,N)$, and those in TFS after mapping are $dt_n (n = 1,2,3,\cdots,N)$, the degree of similarity conservation between TFS and PFS (S) is calculated by using the following formula. The maximum of S is 1.0, and the minimum of S is 0.0.

$$S = 1.0 - \frac{\sum_{n=1}^{N} |dt_n - dp_n|}{N} \quad (2)$$

3.2 Gradually Approaching Search

In "Gradually Approaching Search", when the distance between a required element and a standard element is X in TFS, at first, the neighborhood of the standard element is selected in PFS. Then, the components that correspond to the partial function are searched, and the total function is determined. Next, the searched elements in PFS are mapped onto TFS. One machine closest to the required machine is selected. If it satisfies the required total function completely, the search is finished. If it does not, the selected one becomes a new standard element and the loop is repeated until the required machine is found(Figure 3). This algorithm is shown in Figure 4. This searching models the situation in which a designer finds a required function after changing the composition of the partial function gradually.

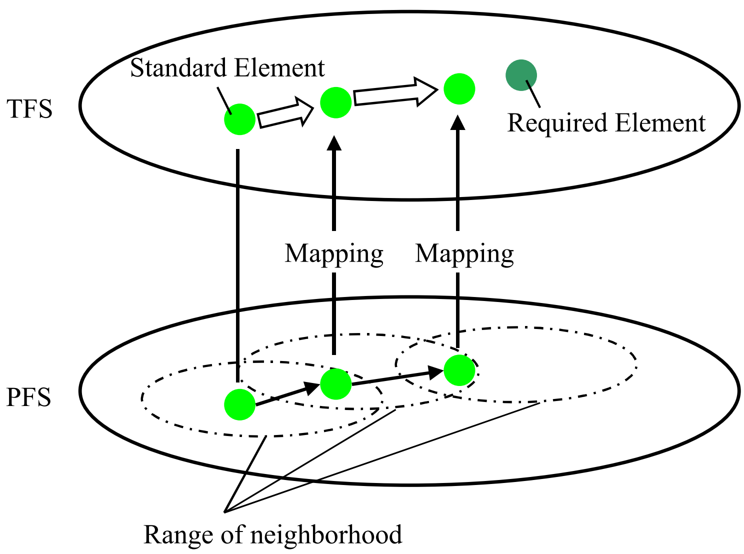

Figure 3. Mechanism of "Gradually Approaching Search"

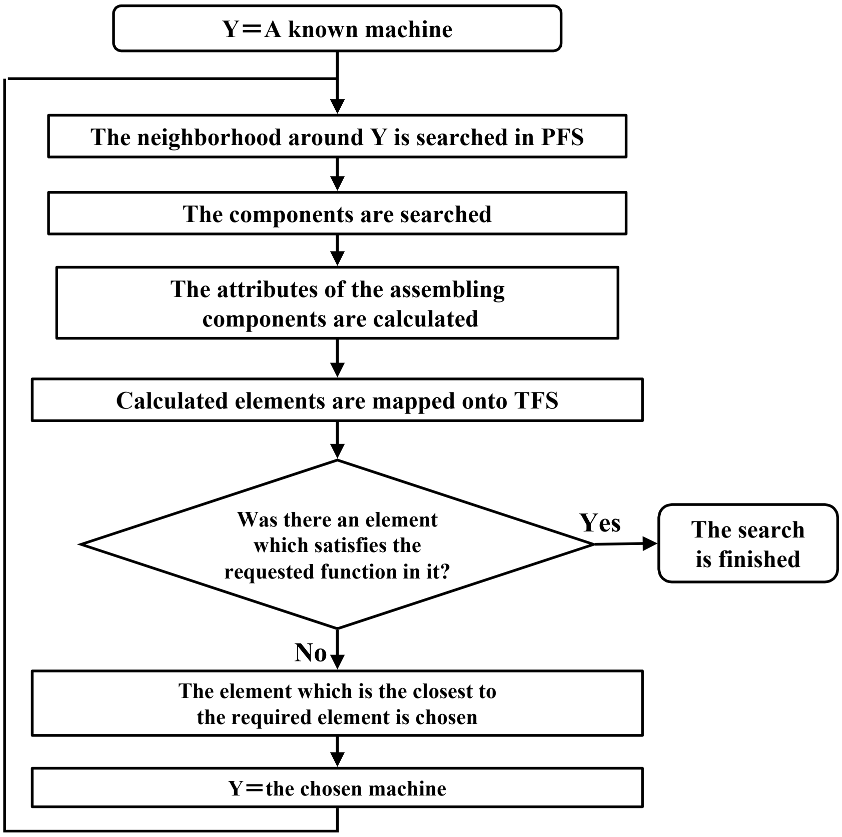

Figure 4. Algorithm of "Gradually Approaching Search"

4. Simulation using "Gradually Approaching Search"

4.1 Set up this simulation

In this computer system, the data of 30 components are stored, and one known machine and the total function of a required machine are inputted, then a combination of components that fills the required total function is outputted. We search a required machine by using "Gradually Approaching Search". The setup of this simulation is as follows.

1. We prepared 30 components of which the mechanisms are simple (gear, belt mechanism, cam, link mechanism, spring, etc…).
2. All of the machines are composed of three components. The total number of combinations of components is 718.
3. Each component has five attributes. Two attributes are for "Input Behavior" and "Output Behavior". The three other attributes indicate a change in transmission speed, the direction of 'Output' from 'Input' on an axis, and the weight of the component.
4. Six kinds of behavior were prepared for "Input Behavior" and "Output Behavior": "Horizontal Straight-Line Movement", "Vertical Straight-Line Movement", "Horizontal Reciprocating Movement", "Vertical Reciprocating Movement", "Rocking Movement" and

"Rotational Movement"..

5. Each function is represented using a function name and its value which is assigned by separating the numerical attributes into three groups. For example, if a machine's numeral value for weight is under –7, the machine has the function that its whole weight is light. If the value is under 7 and beyond -7, the machine has the function that the whole weight is medium. If the value is beyond -7, the machine has the function that the whole weight is heavy.
6. One known machine is prepared in advance. It is the standard machine in the simulation. Its "Input Behavior" is "Horizontal Straight-Line Movement", and its "Output Behavior" is "Vertical Reciprocating Movement", the change in transmission speed is "Constant" throughout the machine, the direction of 'Output' from 'Input' on the X-axis is "Almost Zero", and the machine's weight is "Medium". This machine is composed of component Nos.12, 25, and 29 (Figure 5 and Figure 6).

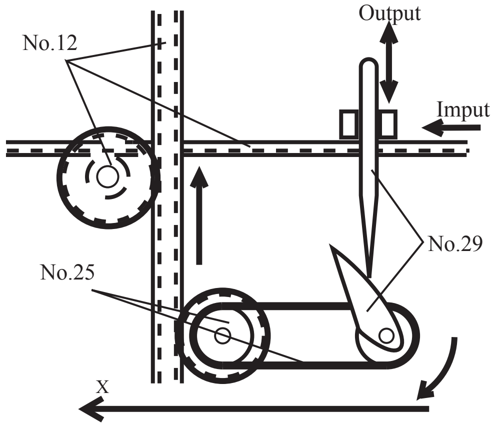

Figure 5. The structure of standard machine

	Function Structure			
	Imput			Output
Behavior	Horizontal Straight-Line Movement	Vertical Straight-Line Movement	Rotational Movement	Vertical Reciprocating Movement
	No.12			
	Speed Change	Increase		
	Direction(X-axis)	Plus		
	Weight	Heavy		
		No.25		
		Speed Change	Constant	
		Direction(X-axis)	Minus	
		Weight	Light	
			No.29	
			Speed Change	Constant
			Direction(X-axis)	Almost Zero
			Weight	Medium

Figure 6. The function structure of standard machine

7. For the required function, its “Input Action” is “Horizontal Perpendicular Movement”, and its “Output Action” is “Vertical Straight-Line Movement”, the change in transmission speed is “Decrease” throughout the machine, the direction of ‘Output’ from ‘Input’ on the X-axis is “Plus”, and the machine’s weight is “Heavy”. The distance between this machine and the standard machine is 0.7 in TFS.
8. The simulation is finished when the one required machine is found. The search efficiency is defined as the reciprocal number of times that the loop is repeated before finding the required machine.
9. In PFS, the functions of 30 components are classified into six classes.
10. PFS is generated at random and for each PFS, fifty simulations are tried and their average is shown.
11. During the search, the degree of similarity conservation between TFS and PFS can be changed since the standard machine changes. Therefore, the average degree of similarity conservation between TFS and PFS through the whole of a search is shown.
12. The maximum number of searching loops is 300. If a required machine is not found in 300 loops, it is considered that a required machine cannot be found.

4.2 Result of simulation

Relationship between the average degrees of similarity conservation of spaces and search efficiency is shown in Figure 7. Figure 7 shows that the conservation of the distance between PFS and TFS throughout the search is related to the improvement in the search efficiency.

When the average degree of similarity conservation of space is the highest, the search was completed in only two loops. After the first search was finished, the machine shown in Figure 8 and Figure 9 was made. The distance between this machine and the standard machine is 0.4 in TFS. After the second search was finished, the machine shown in Figure 10 and Figure11 was made. This machine satisfies the required function.

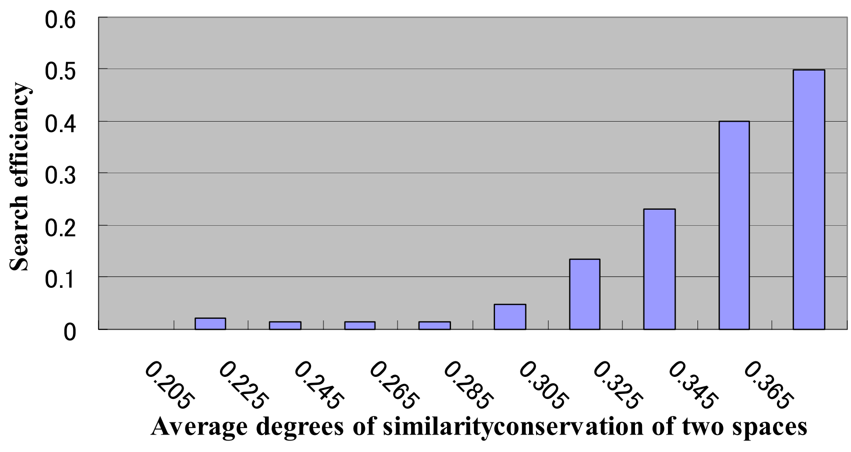

Figure 7. Relation between the average degree of similarity conservation of space and search efficiency

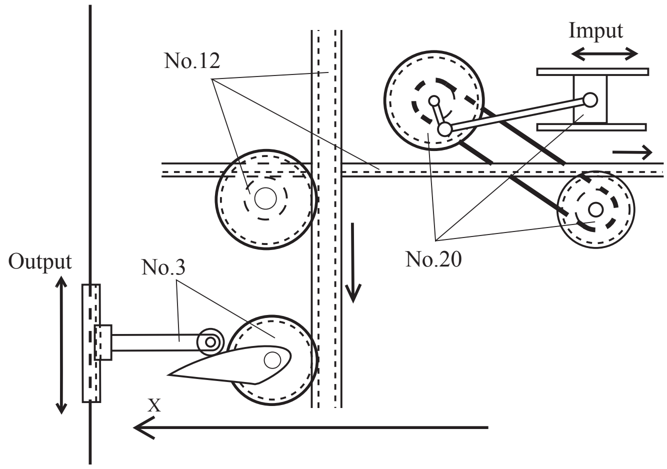

Figure 8. The structure of searched machine after the first search

Function Structure

	Imput			Output
Behavior	**Horizontal Reciprocating Movement**	**Horizontal Straight-Line Movement**	**Vertical Straight-Line Movement**	**Vertical Reciprocating Movement**
	No.20			
	Speed Change	**Constant**		
	Direction(X-axis)	**Almost Zero**		
	Weight	**Heavy**		
		No.12		
		Speed Change	**Increase**	
		Direction(X-axis)	**Plus**	
		Weight	**Heavy**	
			No.3	
			Speed Change	**Increase**
			Direction(X-axis)	**Plus**
			Weight	**Medium**

Figure 9. The function structure of searched machine after the first search

5. Conclusion

It is concluded that the conservation of the similarity between the space for the required function description and the space for the decomposed function description is a key to the space-forming process. This result indicates that forming a suitable function space for finding the design solution efficiently is replaced by the problem of similarity conservation. In other words, the foreseeing nature of design process may be analyzed by the spacial nature. This consideration may extend to analyzing the nature of "trial and error" which is believed to be necessary for design, since a design cannot proceed without information which the designer can know only after they find a design solution.

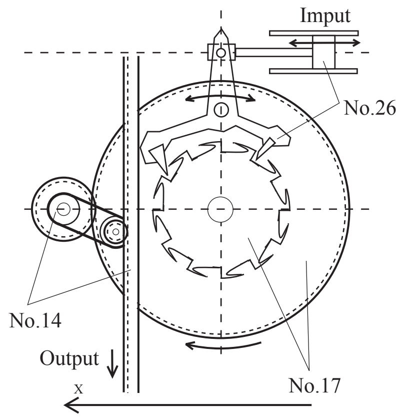

Figure 10. The structure of searched machine after the second search

Function Structure

	Imput			Output
Behavior	**Horizontal Reciprocating Movement**	**Rocking Movement**	**Rotational Movement**	**Vertical Straight-Line Movement**
	No.26			
	Speed Change	**Constant**		
	Direction(X-axis)	**Almost Zero**		
	Weight	**Medium**		
		No.17		
		Speed Change	**Constant**	
		Direction(X-axis)	**Plus**	
		Weight	**Medium**	
			No.14	
			Speed Change	**Decrease**
			Direction(X-axis)	**Almost Zero**
			Weight	**Medium**

Figure 11. The function structure of searched machine after the second search

References

[1] G.Pahl & W.Beitz "Engineering Design – A Systematic Approach" ,Springer-Verlag (1988)

[2] T.Taura "Design Science for Functional Design Process Modeling" ,Proceedings of 10th International Conference on Engineering Design - ICED95- ,pp.456-464 (1995)

[3] N.Suh: "The Principle of Design, Oxford University Press", New York, Oxford (1990)

[4] T.Taura & H.Yoshikawa "A Metric Space for Intelligent CAD", Intelligent Computer Aided Design - Proceedings of the IFIP WG5.2 Working Conference on Int CAD 91 - , pp.133-157 (1992), North-Holland.

For more information please contact:

Toshiharu Taura Graduate School of Science and Technology Kobe University, 1-1 Rokkodai, Nada Kobe, 657-8501 Japan

Tell/Fax: Int +81 78 803 6503, Email: taura@mech.kobe-u.ac.jp,URL http://mi-5.scitec.kobe-u.ac.jp