Function Modeling: Confluence of Process Modeling and Object Modeling

Hideaki Takeda
Nara Institute of Science and Technology*
Yoshiki Shimomura
Mita Industrial Co., Ltd.†
Yasushi Umeda and Tetsuo Tomiyama
The University of Tokyo‡

Abstract

Function is a key concept to integrate object modeling and process modeling in design. In this paper function is defined as a part of FBS (Function-Behavior-Structure) Diagram, where function is a description of behavior abstracted through recognition of behavior for utilization. Function defined above is then used in FEP (Functional Evolution Process) to represent design processes. In FEP, function can be evolved in three ways, i.e., *decomposition evolution, causal evolution,* and *"patch" evolution.* We show a FEP model of design of a weighing scale to illustrate our approach.

1 Introduction

Function is a key concept in design because ideally design is a process in which object is realized from its functionality (see [4]). Although it is well known concept, its definition has been vague yet. In our approach, function is defined using structure and behavior (FBS: Function-Behavior-Structure modeling)[5]. Then it is used in design process where function is gradually evolved (FEP: Functional Evolution Process). In the designers' standpoint of view, function is used firstly as modeling language by which they can compose and develop their requirements. It also serves as object representation which can connect requirements and objects in the middle stage of design. After construction and deliberation of object representation, function representation is again used to evaluate object representation.

In the following section, we will show our model of function and a test case of its application. We will explain FBS modeling in Section 2 and FEP modeling in Section 3. Then we will show a result of analysis of design process by FEP modeling.

*8916-5, Takayama, Ikoma, Nara 630-01, Japan, Fax: +81-7437-2-5269, Email: takeda@is.aist-nara.ac.jp

†Tamatsukuri 1-2-28, Chuo-ku, Osaka, Japan, Fax: +81-6-764-3309, simomura@mita.co.jp

‡Yayoi 2-11-16, Bunkyo-ku, Tokyo 113, Japan, Fax: +81-3-3813-5772, {umeda, tomiyama}@zzz.pe.u-tokyo.ac.jp

2 Function-Behavior-Structure Modeling

There are many approaches to represent function, but there is a common problem, i.e., function and behavior are confused and mixed. Behavior can be directly derived from structure and environment of object, while function is related to not only structure and environment of object but also related to perception of object by designers. For example, suppose function of a car. Some people may say one of its function is "moving", others "carrying", and others "trampling", even if they observe the same behavior. We distinguish function, behavior, and structure levels in object representation (see Figure 1).

Structure level is represented by entities and relations among entities. Entities are identifiers of objects, and relations represent attributes of entities, structures composed by entities, and states of entities. Then behavior is defined as "sequential change of states of objects." In the physical world, changes of states of objects are governed by physical laws. We call this set of definitions of structure and behavior *aspect* which is a basic unit of object representation. Aspect consists of definition of terms and entities (structure) and rules (physical laws). Designers have many kinds of aspects from well-defined aspects (e.g., rigid body dynamics) to ill- or vaguely- defined aspects (e.g., manufacturability).

While behavior is grounded on structure and within the scope of aspect, function is indirectly related to structure and not in the scope of aspect. We define function "a description of behavior abstracted through recognition of behavior for utilization." Function is defined on a chunk of behavior (or behavior itself). There are a lot of possible chunk of behavior. But

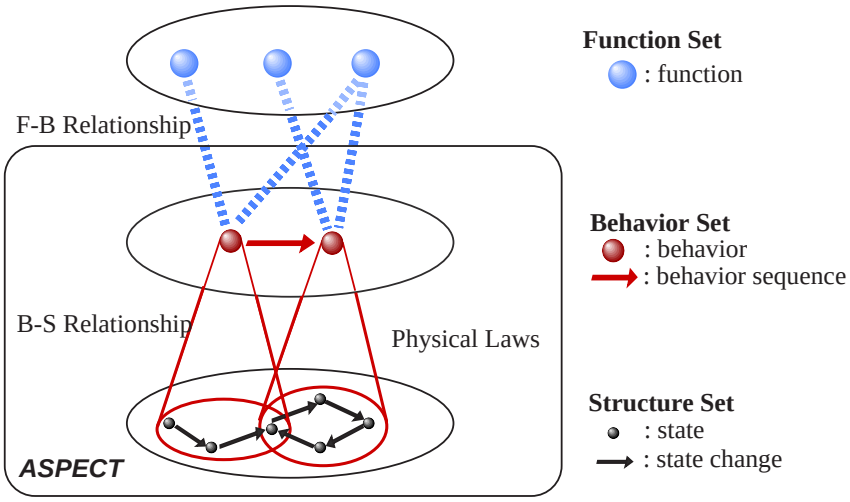

Figure 1: Relationship among Function, Behavior, and Structure

only some of them are meaningful for designers when they recognize and design objects.

Although function is not included in aspect, most of functions are associated to aspects, because behaviors which a function is based on, are in a single aspect. In other words, an aspect has a set of associated functions which is a description of the aspect in the view point of utilization.

3 Functional Evolution Process

In this section, we show function representation in design processes.

Function in the early stage of design is a language to describe requirements. Requirement is not complete in the most of design. Detail of requirement is realized according to detailing of object description, i.e., function is also detailed in design processes. We call it *functional evolution process*.

In order to represent functional evolution process, we provide description of function and its relationship.

A function is represented as combination of a *function body objective entities* and *functional modifiers*. A function body is a symbol which carries meaning of the function. A typical function body is a verb word in sentences like "move" and "carry." An objective entity is an entity which function occurs on or to. It should be realized as an object in structure level until the end of design. A functional modifier is a symbol which restricts functionality in order to match functionality with designers' intention. A typical functional modifier is an adverb word like "precisely" or "firmly."

We provide three types of relations among functions.

decompose It is a typical process for designers to divide a function into sub-functions. This relation should be transferred to behavior and structure level.

be-caused-by It means that new function B is needed to exist in order to realize function A. B is necessary condition for A. This relation should be supported causal relation in behavior level.

be-reinforced-by It means that new function B is recommended to exist in order to realize function A properly. Since B is not necessary condition for A, A alone can exist. But A with B would accomplish its functionality more properly. This relation would be generated as a result of interpretation of functional modifiers.

Functional evolution is to generate functions and relations among functions. According to these three types of functional relations, functional evolution has three different ways.

decomposition evolution Designers try to find sub-functions from a function. Then they try to find either sub-sub-functions or behaviors associated to sub-functions (see Figure 2). For example, Function "to visualize weight" is decomposed into Function "to make weight into displacement" and Function "to convert weight and visualize".

causal evolution Designers try to find functions linked by causal relation. This relation is found through behavior level. First they would find behavior associated to the given function. Then they would find causal behaviors to the behavior by using causal simulation (e.g., Qualitative Simulation[1]). Finally they would obtain functions associated to these behaviors (see Figure 3). For example, Function "to translate weight into

displacement" invokes Structure "spring". But by mental simulation designers find that a new structure like "plate spring" is needed "to guide" spring. Function "to guide" is found through behavior and structure levels.

"patch" evolution Designers would find new functions by consulting functional modifiers. Then designers would combine and test behaviors associated to the given function and derived function in order to know whether the derived function would support the given function according to the modifier (see Figure 4). For example, Function "to enumerate rotation" invokes Structure "rotation plate". Then designers examine how rotation plate can realize Function "to enumerate rotation" with modifier "as large as possible", here the modifier is criteria to evaluate realized function. Then designers find another function "to enlarge indicator" is needed to accomplish the function properly. This function can not be derived in behavior and structure levels only, but functional evaluation can generate it.

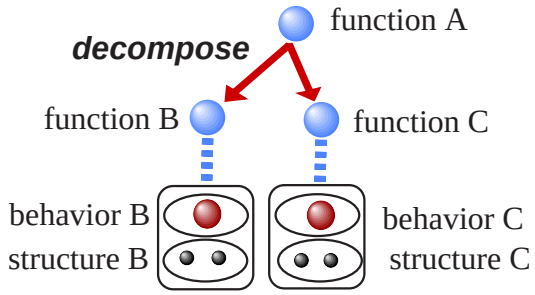

Figure 2: Decomposition Evolution

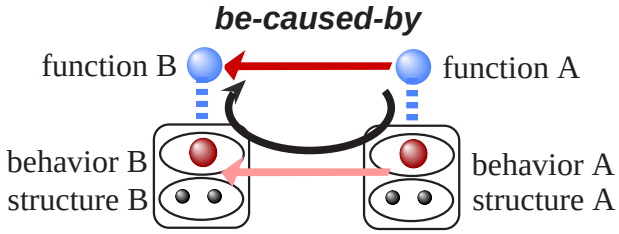

Figure 3: Causal Evolution

Through these processes, functional representation is gradually detailized.

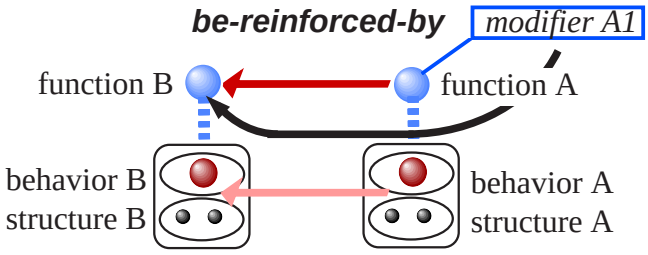

Figure 4: "Patch" Evolution

4 Applicability

We are now testing our representation with some examples. Figure 4 shows a functional evolution process in design of a weighing scale[1].

This figure explains how both function and structure of a weighing scale are developed in this design. At the beginning of the design process, there are a few functions like "to visualize weight" which are directly derived from the given specifications. But as design proceeds, there appeared other functions which are derived as evolution of the given functions. Some of them are internal functions for a scale, but others are external, that is, more detail specifications of a scale. It is an advantage of our approach, because function and structure are developed separately in traditional approaches (for example [2]).

We are also analyzing design processes of PPC copiers with the same method.

References

[1] K.D. Forbus. Qualitative process theory. *Artificial Intelligence*, 24:85–168, 1984.

[2] G. Pahl and W. Beitz. *Engineering Design*. The Design Council, London, 1984.

[3] H. Takeda, S. Hamada, T. Tomiyama, and H. Yoshikawa. A cognitive approach of the analysis of design processes. In *Design Theory and Methodology – DTM '90 –*, pages 153–160. The American Society of Mechanical Engineers (ASME), 1990.

[4] H. Takeda, P. Veerkamp, T. Tomiyama, and H. Yoshikawa. Modeling design processes. *AI Magazine*, 11(4):37–48, 1990.

[5] Y. Umeda, H. Takeda, T. Tomiyama, and Y. Yoshikawa. Function, behaviour, and structure. In J.S. Gero, editor, *Applications of Artificial Intelligence in Engineering V*, volume 1, pages 177–194, Berlin, 1990. Springer-Verlag.

[1] Detail of the design process is shown in [3].

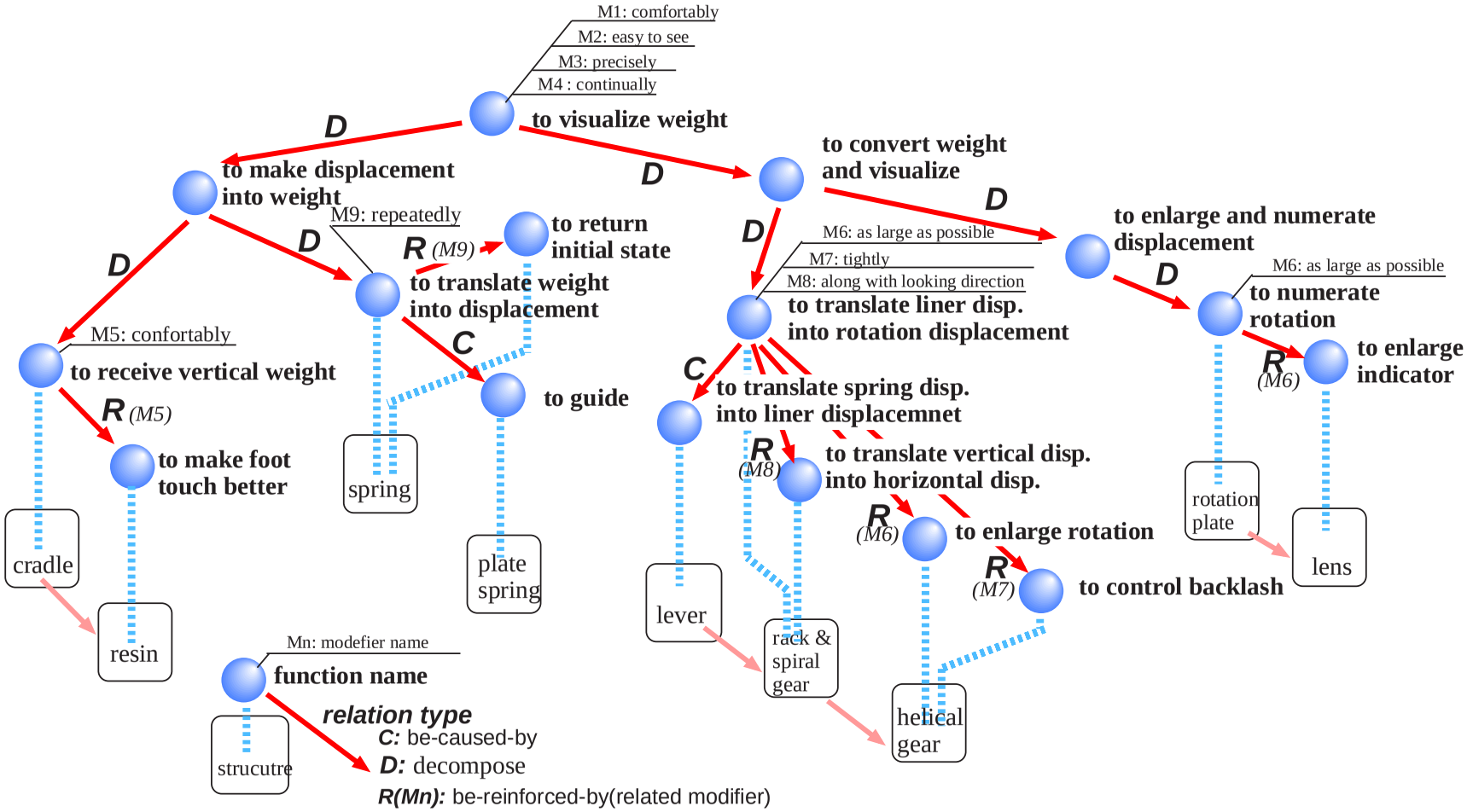

Figure 5: Functional evolution in design of a weighing scale