Synthesis Thought Processes in Design

Hideaki Takeda
Graduate School of Information Science
Nara Institute of Science and Technology (NAIST)
8916-5, Takayama, Ikoma, Nara 630-0101, Japan
takeda@is.aist-nara.ac.jp

Akira Tsumaya
The Modeling of Synthesis Project
The University of Tokyo
Ishikawa Building 11th floor, Hongo 5-25-16, Bunkyo-ku, Tokyo 113-0033, Japan.
tsumaya@modelsyn.t.u-tokyo.ac.jp

Tetsuo Tomiyama
Research into Artifacts, Center for Engineering (RACE)
The University of Tokyo
Komaba 4-6-1, Meguro-ku, Tokyo 153-8904, Japan
tomiyama@race.u-tokyo.ac.jp

Abstract: In this paper, we discuss what properties the synthesis process should have as a thought process and show a tentative synthesis theory. Instead of universality and minimality for knowledge for analysis, knowledge for synthesis in design needs *physicality*, *unlikeness*, and *desirability*. Physicality ensures possibility of existence, while unlikeness and desirability ensure newness, i.e., unlikeness is for ensuring that an artifact is different from other existing artifacts, and desirability for ensuring that it has reason to create newly. Then we show a formalization for synthesis theory that has the above characteristics. It is based on our design process theory in which a design process is an iterative logical process of abduction and deduction on design solution, its properties and behaviors, and knowledge on objects. Synthesis theory for design is defined as reconstruction process of design experiences, each of which contains the logical design process. It consists of three steps, i.e., (1) collecting design experiences; (2) building a model that includes the collected design experiences, and (3) minimizing an element that designers want to find newness.

Keywords: synthesis, analysis, knowledge, design, abduction

1. INTRODUCTION

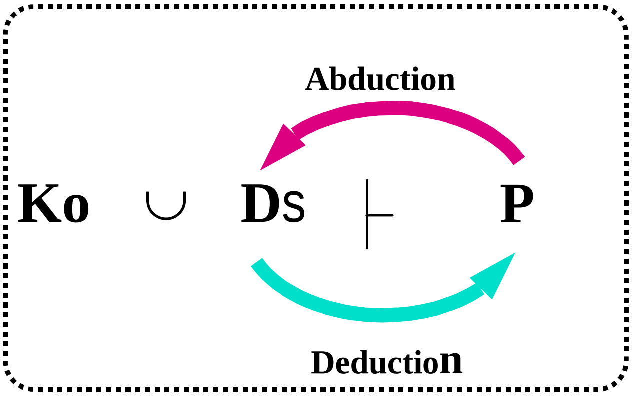
In this paper, we discuss what properties the synthesis process should have as a thought process and propose a preliminary result of logical formalization of the synthesis process.

Considering the synthesis process as a knowledge-based process, the following two questions are important to clarify the synthesis process; the first question is how knowledge is used in the synthesis process, and the second one is what is knowledge for synthesis. The first question is often mentioned as showing difference between synthesis and analysis. One of the clearest answers is abduction proposed by C.S. Peirce [Peirce, 1935]. But the second question is often missing. We tend to assume implicitly that knowledge for synthesis is similar to knowledge for analysis. But the aims of synthesis and analysis are so different that it is reasonable to question this assumption.

The aim of analysis is to clarify characteristics of objects. To clarify objects means to explain different objects in the same manner. In order to apply different objects as much as possible, characteristics should be universal and minimum. It implies that requirements for knowledge for analysis are also universality and minimality.

On the other hand, the aim of synthesis is to create objects having necessary characteristics. In this case, it is not required that characteristics are universal and minimum, rather they should *not* be. In order to capture human desire for objects, characteristics should be as rich as possible to represent various desires. Thus requirements for knowledge for synthesis are not universality and minimality but rather individuality and diversity. The last statement indicates that the assumption underlying the traditional logical approach is not appropriate because it is to capture our world with minimum and universal axioms.

We have already proposed how synthesis can be captured in a logical framework that includes abduction [Tomiyama et al., 1997] [Takeda et al., 1990a]. It is based on logical theory, but we introduce multiple theories in a logical framework. The heart of our formalization is that synthesis is not applying logical theories but extending and composing logical theories enough to represent human desire to create new artifacts. It is denying the assumption of the traditional logical approach that we mentioned above, but it is not clear in these papers how theories can be combined. In this paper, we extend our formalization by including requirements for knowledge for synthesis.



Ds: Design Solution

P: Properties and Behavior of Design Solution

Ko: Knowledge on Objects

Figure 1; *A Logical Framework of design process*

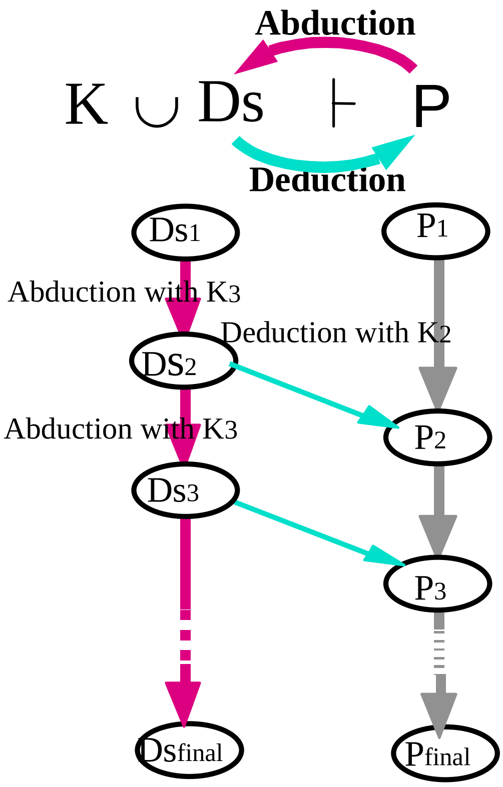


Figure 2; *Iteration of abduction and deduction*

2. THE THREE FEATURES OF DESIGNING ARTIFACTS

In the previous section, we mentioned that knowledge for synthesis is related to represent our desires to create new artifacts. Then the question is what is necessary to create “new” artifacts. In other words, it is a question what conditions are required to agree that “it is new”.

We propose three features to ensure “newness” for artifacts. One is *physicality* that ensures artifacts existing in the world. It is necessary for all existing artifacts that are not even new. But it is better for designers to understand physicality because they can use the relations between the physical characteristics to realize their intention. Knowledge for synthesis should include knowledge for analysis in this sense. But it is not all about knowledge for synthesis. Knowledge for analysis tends to be minimum and universal, while knowledge for synthesis needs much variety to represent requirements for artifacts. It is the key issue for synthesis to provide not minimum but enough knowledge for the given requirements.

The second feature is *unlikeness*, which means that it is a unique artifact. There are two problems in unlikeness. The first one is which set of artifacts we should take into consideration to ensure unlikeness. It is impossible to compare the designing artifacts with all artifacts in the world. What we can do at most is to collect artifacts that we have designed or encountered, learned. We call such an artifact as a design experience. We should collect design experiences to judge unlikeness. The second problem is how to compare the designing artifacts with others. It is often difficult to compare the designing artifacts with some other design experience because every design experience is so different in which situation we encountered it that its representation may be different from the designing artifacts. It is the problem on ontological integration.

The third feature is *desirability*. Even if physicality ensures that the designing artifact can exist in our world and unlikeness ensures that it has no similar artifacts, it is not enough reason to realize it in our world. For example, there are no reasons to create a new artifact that is much more complicated to the existing artifacts with the same functions. We need criteria to ensure that is has reasons to create new artifacts. It is a very difficult feature because most of reasons are implicit requirements from the society. We can point out an example for it, i.e., “Occam's razor”. To make something minimize is a good reason to create new artifacts. To minimize user operations is a good reason to design a new consumer product.

3. REQUIREMENT FOR FORMALIZATION OF THE SYNTHESIS PROCESS

In this section, we verify our formalization of design process [Takeda et al., 1990a][Takeda et al., 1992] in the light of the above three features of the synthesis process.

The primary formula in our formalization is the following one (see Figure 1).

$$Ds \cup Ko \models P$$

Here *Ds* is a set of logical formula that represents description of design objects. *P* is a set of logical formula that represents description of properties of design objects. *Ko* is a logical formula that represents knowledge on objects. This formula means that description of design objects with knowledge on objects can deduce description of properties of design objects. To keep this formula is to keep physicality condition.

The ideal design process is defined as the inference process in which description of design objects can be inferred abductively from knowledge on objects and a part of properties of design objects as design requirements. It is not deduction process but abduction process [Peirce, 1935].

In the actual design process, design requirements and available design knowledge cannot be determined in advance but be defined during design. So the real design process is defined as process repeating abduction and deduction (see Figure 2).

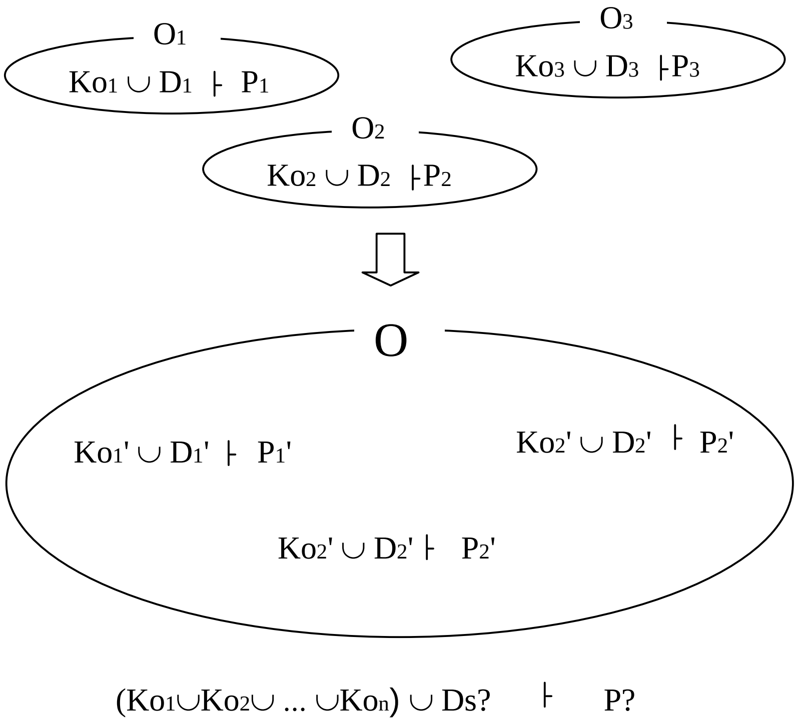

O_1
$Ko_1 \cup D_1 \vdash P_1$
O_3
$Ko_3 \cup D_3 \vdash P_3$
O_2
$Ko_2 \cup D_2 \vdash P_2$
O
$Ko_1' \cup D_1' \vdash P_1'$
$Ko_2' \cup D_2' \vdash P_2'$
$Ko_2' \cup D_2' \vdash P_2'$
$(Ko_1 \cup Ko_2 \cup \ldots \cup Ko_n) \cup Ds? \vdash P?$

***Figure 3**: Integration of Design Experiences*

Description of properties of design objects can be inferred deductively from description of design objects and knowledge on objects. Then it can help further decision of design requirements. Description of design objects can be inferred abductively from description of properties of design objects and knowledge on objects. It can in turn suggest what kind of knowledge is needed for the further inference.

This formalization can explain design process in a computational way and also match the practical design processes [Takeda et al., 190b]. But the formalization has three unsolved issues. The first one is evaluation of design solutions. Abduction process can theoretically infer many solutions without any order[1]. We need some schemes to compare them. This issue corresponds to the desirability condition. The second is how to provide knowledge. We assumed that knowledge could be provided a priori, but providing knowledge is also an important process in design. The third issue, which is more fundamental one, is how to provide representation of description of design objects, properties of design objects, and knowledge on objects. The second and third issues correspond to the unlikeness condition, i.e., what ontology and knowledge we should

[1] Some logical methods for evaluation of abduction have been proposed. For example, to minimize propositions in hypothesis (solution) and links from hypothesis to observation [Ng and Mooney, 1990]. But there are no reason why we should do such a way.

take into consideration.

4. THE FIRST ATTEMPT TO FORMALIZE THE SYNTHESIS PROCESS

We model the synthesis process as an intended integration process of design experiences. The basic idea is that designers can evolve not only their designing artifacts but also their ontology and knowledge by using their design experiences. We extend our formalization of design process to capture this process. It consist of the three processes as follows (see Figure 3):

(1) Collecting design experiences
(2) Forming a model to represent the collected design experiences
(3) Minimizing an element that designers want to make it new.

(1) Collecting design experiences

The issue for collecting design experiences is how to represent design experiences. What we consider artifacts during design process means not only to remember an artifact itself, e.g., to enumerate its attributes, but to remember or imagine what purpose it has and why it comes to have such attributes. In other words, we can imagine its design process. So we model a design experience as a quadruplet of Ds, Ko, P, and O, i.e., description of design objects, knowledge on objects, properties of design objects, and ontology to represent these three formulae respectively. We represent a set of design experiences as

$$de_1 = (Ds_1, Ko_1, P_1, O_1), \quad \ldots, \quad de_n = (Ds_n, Ko_n, P_n, O_n)$$
$$Ds_k \cup Ko_k \models P_k \ , Ds_k \in O_k, Ko_k \in O_k, P_k \in O_k \quad (1 \le k \le n)$$

where de_k denotes k-th design experience.

(2) Forming a model to represent the collected design experiences

Although we can collect a set of design experiences, they are probably based on different ontologies so that we should integrate design experiences. It is to find a substitution ϕ of predicates that can map all the given ontologies O_1 ... O_n to a single ontology O.

$$de_1' = (Ds_1', Ko_1', P_1', O_1') = de_1\phi = (Ds_1\phi, Ko_1\phi, P_1\phi, O_1\phi)$$
$$\ldots$$
$$de_n' = (Ds_n', Ko_n', P_n', O_n') = de_n\phi = (Ds_n\phi, Ko_n\phi, P_n\phi, O_n\phi)$$
$$O = O_1\phi \cup O_2\phi \cup \ldots \cup O_n\phi$$
$$Ko' = Ko_1\phi \cup \ldots \cup Ko_n\phi$$

Table 1: A Cognitive model for case-based design

States in case-based design	**Formalization**
The current design problem:	(Ds, Ko, P, O)
A design case:	(Ds_N, Ko_N, P_N, O_N)
The next design problem:	(Ds', Ko', P', O')

Processes in case-based design	**Formalization**
1. Adaptation of cases	
1-1. modification of cases	$Ds' = f(Ds)$
1-2. translation of representation	$Ds_N' = Ds\phi,\ Ko_N' = Ko_N\phi,\ P_N = P_N\phi$
2. Application of cases	
2-1. change of design solution	
2-1-1. use of cases	$Ds' = Ds \cup Ds_N'$
2-1-2. rejection of cases	$Ds' = Ds$
2-2. change of design knowledge	$Ko' = Ko \cup Ko_N'$
2-3. change of specifications	$Ds' \cup Ko' \models P'$

The new ontology *O* is a base to represent the new designing objects, and new knowledge *K* represents the current available knowledge.

(3) Minimizing an element that designers want to make it new.

For example, to find a simplest solution is to minimize *Ds*, i.e., to find *Ds* where $|Ds| < |Ds_k'|$ $(1 \le k \le n)$. The other example is to find minimum knowledge. It means to find *O* under the condition that *Ko'* is smallest.

5. A COGNITIVE MODEL FOR CASE-BASED DESIGN

We consider how this formalization can apply actual design processes. The straightforward application is case-based design process. Applying the formalization to case-based design leads a new cognitive model for it.

The basic process of case-based design is iteration of recalling, adapting, and applying cases for the current design objects. In our context, adaptation of cases consists of two processes, i.e., modification of cases within their ontologies and translation of their ontologies. The former includes operations like changing of parameters and configuration. The latter happens when we are applying cases to the design objects in different domains. In such cases we should map concepts in a domain to those in the other domain.

Application of cases means not only changing of the current design objects, but also changing of knowledge and specifications. In our formalization, cases do not means either design objects themselves or pairs of specifications and design objects but design experiences, i.e., sets of object properties including specifications, design objects, and design knowledge that can reason how they are designed. When a case is applied to the current design objects, knowledge for it is also applied to knowledge for the current design objects. New knowledge can be valid even if the case is abandoned to apply. New knowledge may suggest new specifications. Table 1 summaries a cognitive model for case-based design process and correspondence to our formalization.

Some examples can be excerpted from the protocol data [Cross, 1996]. In this protocol, the task for design is to design a carrying device that can fasten a backpack on the mountain bike. During the design process, the designers discussed where the device should be located. One idea was as follows;

I: handlebars? yeah try that
J: or off this handlebar stem even because that's fixed but if it's off the handlebars you know it's like an old bike basket that way like the Wizard of O
K: heavy to steer tends to

The idea "old bike basket" was instantly rejected but considering this idea invoked knowledge that “power of steering bike should be reasonable”. The specification "driving a bike" now includes "good steering". In this example, the case is used as 2-1-2 (rejection of case), 2-2 (change of design knowledge), and 2-3 (change of specification). The other example is as follows;

K   ooh well I've done a lot of lake touring and I've done front panniers and I've done rear panniers and em
...
I   boy that is one pannier
K   yeah front panniers you could you could set it up so you could have one of these on each side there's no guarantee you'd always have two but it's actually

The case is almost as same as the first example, i.e., "good balance is needed for driving a bike" was introduced but the case itself was refused to use.
But the other idea that expanded to this one was suggested.

J   what if it's split
K   not as bad as you'd think to have just one that's off
J   what if the em the back backpack folds

```
I   they already have a hold in the middle so they could uh yeah
K   oh like add a joint right here
I   yeah
J   so yeah so it's sorta like
K   so it's more like a pannier
```

The case was translated and then applied to the current design problem. This translation is to extract a basic structure of "pannier" (two connecting components) and to map one pannier to one compartment of the backpack. This use corresponds to 1-1 (modification of cases) and 1-2 (translation of representation).

6. CONCLUSION

We show the first step of a logical formalization of the synthesis process. It is a new approach to formalize the synthesis process in a logical framework because it can explain how we can obtain knowledge that is often missing point when discussing logical frameworks. But there are many unsolved problems; e.g., we could not show how to achieve desirability. We are planning how this formalization can work in actual design processes by testing it in protocol analysis.

ACKNOWLEDGEMENT

This research is supported by the *Modeling of Synthesis Project* (JSPS-RFTF 9600701) under the *Research for the Future Program* of the Japan Society for the Promotion of Science.

REFERENCES

[Cross et al., 1996] Cross, N.; Christiaans, H.; Dorst, K. editors; "Analysing Design Activities"; John Wiley & Sons, 1996

[Ng and Mooney, 1990] Ng, H.T.; Mooney; R.J.; "On the role of coherence in abductive explanation"; In Proceedings of AAAI-90, pp. 337-342, 1990

[Peirce 1935] Peirce, C.S.; "Collected Papers of Charles Sanders Peirce", volume 5; Harvard University Press, Cambridge, MA, 1935

[Takeda et al., 1992] Takeda, H.; Tomiyama, T.; Yoshikawa, H.; "A logical and computable framework for reasoning in design"; In D.L. Taylor and L.A. Stauffer, editors, Design Theory and Methodology - DTM '92 -, pp. 167-174; The American Society of Mechanical Engineers (ASME), 1992

[Takeda et al., 1990a] Takeda, H.; Veerkamp, P.; Tomiyama, T.; Yoshikawa, H.;

"Modeling design processes"; AI Magazine, Vol. 11, No. 4, pp. 37-48, 1990

[Takeda et al., 1990b] Takeda, H.; Hamada, S.; Tomiyama, T.; Yoshikawa, H.; "A cognitive approach of the analysis of design processes"; In Design Theory and Methodology - DTM '90 -, pp. 153-160; The American Society of Mechanical Engineers (ASME), 1990

[Tomiyama et al., 1997] Tomiyama, T.; Murakami, T.; Washio, T.; Kubota, A.; Takeda, H.; Kiriyama, T.; Umeda, Y.; Yoshioka M.; "The modeling of synthesis -from the viewpoint of design knowledge"; In WDK 25, Proceedings of the 11th International Conference on Engineering Design in Tampere 199, volume 3, pp. 97-100, Tampere, Finland, 1997; Laboratory of Machine Design, Tampere University of Technology