Integration of Knowledge in Synthesis Process

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 is knowledge for synthesis and how knowledge is used in synthesis process. Variety of knowledge is crucial in synthesis process like design. Such variety of knowledge in turn makes it difficult to collect and integrate knowledge that is the main process of synthesis. We model synthesis process as integration of knowledge units called *design experiences* that is described on different ontologies. We show a formalization for synthesis theory 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. In our formalization, a unit of knowledge that we call *design experience* is a piece of information with objects, knowledge on the objects, properties on the objects, and ontology to describe these types of information. Synthesis process is to integrate a set of knowledge units as follows; (1) collecting design experiences; (2) building a model that can includes the collected design experiences by integrating ontologies of the design experiences, and (3) minimizing an element that designers want to find newness.

Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5) Stockholm, Sweden, August 2, 1999

(V.R. Benjamins, B. Chandrasekaran, A. Gomez-Perez, N. Guarino, M. Uschold, eds.)

http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-18/

1 Introduction

In engineering design, both synthesis and analysis thought processes play an important role. However, the synthesis thought process is not well understood or modeled. This stands in a sharp contrast to the analysis thought process, which is well understood and applied. For example, most of the current CAD systems employ the hierarchical-decomposition strategy that is a form of analysis thought process. Such a strategy can only lead to “existing” design and not lead to “new” design. To solve this problem, understanding the synthesis process is needed. Our project team has already proposed how synthesis can be captured in the framework including multiple viewpoints and the role of abduction [Tom97]. But it is still not clear how knowledge can be obtained and how theories can be integrated. The objective of this study is to understand synthesis better and to propose a preliminary result of formalization of the synthesis process.

The most interesting and also most difficult characteristics of engineering design is that knowledge is engaged to both scientific theories and expertise. It is almost impossible to accomplish engineering design without scientific knowledge on the target domain. On the other hand, scientific knowledge alone is not sufficient to achieve design but much expertise is needed in most cases. It shows that knowledge for design is neither totally universal nor totally personal, but knowledge is actually in between, i.e., it sometimes behaves like universal knowledge and sometimes like personal knowledge.

Our strategy for knowledge for design is as follows;

1. Content of knowledge is theory on object
   Knowledge provides reasons why objects behave in certain situations. In some domains, it is truly scientific theory but in other domain it is more naive theories. Therefore there can be much variety of theories.
2. Unit of knowledge is expertise.
   With much variety of knowledge as mentioned above, it is difficult to assume that all knowledge is applicable. Indeed, designers may select and use

some of them. We call this usage pattern of knowledge *design experience.* It is actually a unit of knowledge that can be used to design some artifacts in the past.

3. Relation among units of knowledge is indicated by ontology
   If units of knowledge are totally irrelevant, there can be no ways to compare or to combine. We assume ontology for each unit of knowledge. It is ontology on which the theory of objects can be described. Comparison and combination between units of knowledge can be done as operations of ontologies.

In our formalization, a unit of knowledge is a piece of information with objects, knowledge on the objects, properties on the objects, and ontology to describe these types of information. Synthesis process is to integrate a set of knowledge units where ontology integration may be required.

2 Three Features of Designing Artifacts

In this section, we discuss the features of design artifacts as synthesis process.

2.1 Synthesis and Analysis

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 [Pei35]. 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 [Tom97] [Tak90a]. 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.

2.2 Features of Newness

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.,

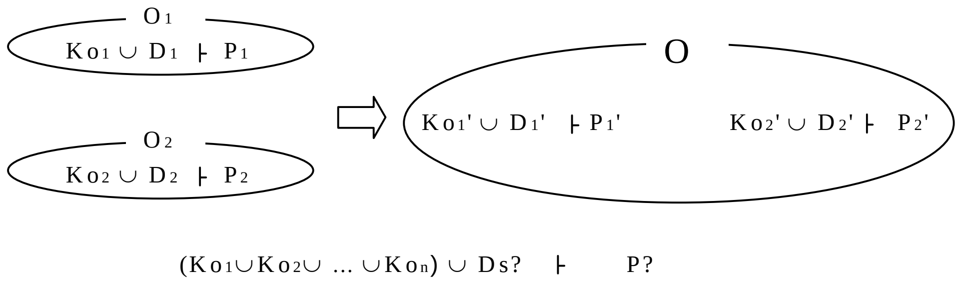

Figure 1: Synthesis Process as Integration of Design Experiences

"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 Formalization of the synthesis process

In this section, we will show our formalization of synthesis that can satisfy the above features, i.e., physicality, unlikeness, and desirability. It is based on formalization of design processes. Although the formalization succeeded in explaining design process as logical process but it not mentioned how knowledge that drives the logical process can be obtained. We expand it by considering knowledge introduction process, i.e., management of design experiences.

3.1 Formalization of Design Process

At the start point, we use our formalization of design process [Tak90][Tak92a]. The primary formula in our formalization is as follows.

$$Ds \cup Ko \models P$$

Where 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. This formalization can explain design process computable, but not explain how to provide knowledge, properties, and description of design object.

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 [Pei35].

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. 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.

3.2 Formalization of Synthesis Process

The above formalization can explain design process in a computational way and also match the practical design processes [Tak90b]. 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. 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 than the second, 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 take into consideration.

Therefore, we model the synthesis process as an intended integration process of design experiences. Figure 1 shows integration 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 consists following three processes:

(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

Table 1: Formalization of "Design Experience"-based Design

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

Processes in case-based design	Formalization	Number of cases
1. Adaptation of cases		
1-1. modification of cases	$Ds' = f(Ds)$	17
1-2. translation of representation	$Ds_N' = Ds\phi, Ko_N' = Ko_N\phi, P_N' = P_N\phi$	15
2. Modification elements of the current design problem by using cases		
2-1. change of design solution		
2-1-1. introduction of candidate solution	$Ds' = Ds \cup Ds_N'$	21
2-1-2. rejection of candidate solution	$Ds' = Ds$	32
2-2. change of design knowledge		
2-2-1. change of design knowledge	$Ko' = Ko \cup Ko_N'$	84
2-2-2. use of existing design knowledge	$Ko' = Ko$	20
2-3. change of specifications		
2-3-1. change of specifications	$Ds' \cup Ko' \models P'$	31
2-3-2. confirmation of specifications	$P' = P$	13
2-4. change of viewpoint	$V' \neq V,\ V', V \in Ko$	15

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.

Ontology consists of vocabulary and conceptualization [4]. We technically assume unique names across ontologies, thus vocabulary is represented as a set of predicates *Ov* and conceptualization as a logical theory (an ontological theory) *Ot* among the predicates. The meaning that a formula *A* is *ontologically* consistent with $O=\langle Ov,Ot\rangle$ is that all predicates in *A* are in the set of predicates of *O* and that *A* is consistent with the ontological theory of *O*. Furthermore we introduce viewpoint $V \subset Ov$ that can restrict use of knowledge. It is represented as a part of vocabulary that can cut off some part of knowledge. We represent a set of design experiences as

$$de_1 = (Ds_1, Ko_1, P_1, Ov_1, Ot_1, V_1),$$
$$\ldots,$$
$$de_n = (Ds_n, Ko_n, P_n, Ov_n, Ot_n, V_n)$$
$$Ds_k \cup Ko_k \models P_k\ ,$$
$$Ds_k, Ko_k, P_k \text{ are ontologically consistent with } O_k$$
$$(1 \le k \le n)$$

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

$$de_1' = (Ds_1', Ko_1', P_1', Ov_1', Ot_1', V_1')$$
$$= de_1\phi$$
$$= (Ds_1\phi, Ko_1\phi, P_1\phi, Ov_1\phi, Ot_1\phi, V_1\phi)$$
$$\ldots$$
$$de_n' = (Ds_n', Ko_n', P_n', Ov_n', Ot_n', V_n')$$
$$= de_n\phi$$
$$= (Ds_n\phi, Ko_n\phi, P_n\phi, Ov_n\phi, Ot_n\phi, V_n\phi)$$
$$Ov = Ov_1\phi \cup Ov_2\phi \cup \ldots \cup Ov_n\phi$$
$$Ot = Ot_1\phi \cup Ot_2\phi \cup \ldots \cup Ot_n\phi$$
$$Ko' = Ko_1\phi \cup \ldots \cup Ko_n\phi$$

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 $O_1 \ldots O_n$ to a single ontology $O=\langle Ov, Ot\rangle$. The new ontology $\langle Ov,Ot\rangle$ is a base to represent the new designing objects, and new knowledge *Ko'* 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

$$\left|Ds\right| < \left|Ds_k'\right| \quad (1 \le k \le n)$$

The other example is to find minimum knowledge. It means to find *O* under the condition that *Ko'* is smallest.

4 Case-based Style Design as Use of Design Experience

In this section, we re-examine our formalization in comparison with case-based design, and then show how this formalization can work by applying interpretation of design protocol data.

4.1 Case-based Design

It is noticeable that design process in our formalization has similarity to so-called *case-based design*. Case-based design is well known approach for computer-aided design that is strongly influenced by case-based reasoning model developed in AI field (see e.g. [Mah95]), and some systems are proposed (see e.g., CADSYN[Mah95] and CADRE[Hua93]). The basic reasoning process of case-based design is simple, i.e., it is iteration of recalling, adapting, and applying cases. The real problem for case-based design is what is case content [Kol93a]. Case descriptions typically capture a problem, its solution, and the outcome of the solution [Kol93b]. But such descriptions are not adaptable to cases in design that has high flexibility.

In our approach, design experiences can play similar roles to cases in case-based design. The difference is that the former includes not only specifications and solutions but also knowledge on objects related to the specifications and solutions. The set of these elements enables flexible operations, i.e., simulation of design process as reasoning on the knowledge and translation of representation as translation of knowledge.

4.2 Re-formalization as Case-based Design

Design experiences can be mapped to design cases in case-based design. But note that the latter just includes design objects as solutions and design specifications as problems but the former includes knowledge on the related objects and its ontology in addition to design objects and specifications. We re-interpret our formalization as case-based design process by regarding design experiences as cases.

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 shows how our formalization can correspond to case-based design process.

4.3 Analysis of Protocol Data

Here, we used the team design of the Delft Protocols [Cro96]. We find 169 cases on the protocol and analyzing each case what the case is used as. The third column of the Table 1 is summery of the analysis of the protocol data.

Some examples can be excerpted from the protocol data. 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 Oz
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 rejection of case, change of design knowledge, and 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.

As this result, we could find that the new knowledge are used as following four types:

1. Suggest new specification

A “case” is applied to change/introduce/remember the knowledge and the knowledge is used to suggest to change description of design object. Usually, this type of knowledge is also used to suggest the reason why the “case” is rejected for design solution. 24 cases of knowledge are categorized in this type.

2. Suggest change of viewpoint

In this type, knowledge leads a new viewpoint. Changing viewpoint is very important role for “new” design. On this analysis, 9 cases of changing viewpoints are this type and this result show changing viewpoint is mainly leaded by obtain new knowledge.

3. Introduce knowledge and apply/use the case

Some introduced knowledge causes modification or translation of new imagined/remembered cases. 12 cases of knowledge used for this type.

4. Knowledge as an example

In this type, knowledge is obtained only as an example. During brainstorming, knowledge that designer imagined/remembered is mainly categorized this type. But some of them are used later as the previous three types.

We also could find that using and introducing new knowledge, evolution of design process is done by repeating change of viewpoint, collecting examples, suggesting new specifications, and applying cases in order.

5 Conclusion

We propose a logical formalization of the synthesis process. Although we show beginning stage, it can explain how to get knowledge that is one of the core part of the synthesis process. We also apply the formalization to case-based design and find how to provide knowledge and used for what. For future work, we plan to make clear the act of formalization in actual design processes by more detailed 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

[Cro96] Cross, N., Christiaans, H., and Dorst, K. editors, “Analysing Design Activities”, John Wiley & Sons, 1996

[Gua98] Guarino, N, "Formal Ontology and Information Systems", In N. Guarino (ed.), Formal ontology in Information Systems. Proceedings of FOIS'98, IOS Press, 1998.

[Hua93] Hua, K.; Faltings, B.: "Exploring case-based Design - CADRE"; Artificial Intelligence in Engineering Design, Analysis and Manufacturing, 1993

[Kol93a] Kolodner, J. L.; Wills, L. M.; "Paying Attention to the Right Things: Issues of Focus in Case-Based Creative Design"; AAAI-93 Case-Based Reasoning Workshop, Washington, D.C., July 1993, pp.19-25.

[Kol93b] Kolodner, J. L.; "Case-Based Reasoning"; Morgen-Kaufman Publishers, Inc., San Mateo, CA.

[Mah95] Maher, M. L.; Balachandran, M. B.; Zhang, D. M.; "Case-Based Design"; Lawrence Erlbaum Associates, 1995

[Pei35] Peirce, C.S.; "Collected Papers of Charles Sanders Peirce", volume 5; Harvard University Press, Cambridge, MA, 1935

[Tak90a] Takeda, H., Veerkamp, P., Tomiyama, T., and Yoshikawa, H., “Modeling Design Processes”, AI Magazine, Vol. 11, No. 4, 1990, pp. 37-48.

[Tak90b] 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

[Tak92] Takeda, H., Tomiyama, T., and 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 -, The American Society of Mechanical Engineers (ASME), 1992, pp. 167-174.

[Tom97] Tomiyama, T., Murakami, T., Washio, T., Kubota, A., Takeda, H., Kiriyama, T., Umeda, Y., and Yoshioka, M., “The Modeling of Synthesis - From the Viewpoint of Design Knowledge”, Proceedings of the 11th International Conference on Engineering Design, Volume 3, Schriftenreihe WDK 25, Tampere 1997, pp. 97-100.