INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

A GENERAL FRAMEWORK FOR MODELING OF SYNTHESIS - INTEGRATION OF THEORIES OF SYNTHESIS -

Hideaki Takeda, Masaharu Yoshioka, and Tetsuo Tomiyama

Keywords: design theory, knowledge-based engineering, descriptive models of design process, abduction, model-based reasoning

1 Introduction

Engineering design consists of a variety of thought processes such as analysis, synthesis, problem-solving, and decision-making. Among these, the most crucial process in design is “synthesis” or “synthesis-oriented thought process,” because synthesis brings about creativity of design. In spite of the importance of synthesis, compared with analysis, synthesis is less understood and codified as a model.

In this paper, first we clarify the roles of synthesis in design, and then logically define design as a synthesis-oriented thought process in a logical framework as a reasoning process. The core part of the synthesis-oriented thought process is performed by abduction, but the overall process is logically realized by combination of abduction and deduction. Next, synthesis and analysis are formally distinguished.

Second, we introduce the object-dependent (or model-based) approach that has a power to naturally represent objects in the physical world. This combination of logical and object-oriented approaches is commonly used to empower a knowledge representation scheme for the sake of formality and expressiveness. Logic describes inter-model relations and general operations to models, while models describe the nature of design objects and their possible operations. We will redefine synthesis to deal with knowledge about design objects by introducing the concept of model-based reasoning in which modeling operations perform references to an extra-logical world in asserting logical formulae. In our formalism of combination of logical and model-based approaches, this explicit reference to non-logical facts in extra-logical (i.e., model-based) world naturally integrates logic and models.

Based on this concept, we introduce and operate a variety of knowledge on objects in an integrated way in a. reasoning framework for synthesis. The framework has the following three features, viz., duality of abduction and deduction, multiple viewpoints, and duality of logical and model-based reasoning. We also illustrate a prototype system of the reasoning framework of synthesis.

2 Synthesis and analysis

Synthesis can be identified in such activities as scientific discovery, design, and art including writing novels and painting pictures. While synthesis is obviously the core of these activities, its nature is almost unclear. This chapter overviews how we deal with synthesis and defines our approach to synthesis.

Synthesis is often counterposed to analysis and defined as an opposition of analysis, while analysis is defined independently of synthesis. Synthesis and analysis often collocate, but the patterns of occurrence of synthesis and analysis are different. For instance, in scientific discovery, synthesis very often appears after analysis. In contrast, synthesis and analysis appear alternately and repeatedly during design.

Synthesis and analysis are also often discussed logically. A typical interpretation is that analysis is deductive while synthesis is non-deductive; according to Peirce [1], synthesis is often explained by abduction. However, this correspondence might be confusing, because analysis (or synthesis) as human activities can contain activities other than purely logical reasoning. To clarify this, we distinguish *analysis* (or synthesis) *as human activities* from *analysis* (or *synthesis*) *oriented thought process*. Abduction and deduction are mainly used to refer to reasoning processes.

3 Logical formalization of synthesis

To logically deal with synthesis and analysis, we assume that synthesis and analysis are *rational thought processes based on theories*. Here a *theory* is a set of logical correspondence relations (among axioms, facts, and theorems) that gives explanations for a phenomenon described in facts based on axioms and theorems included in the theory. A *thought process based on theories* is a reasoning process in which theories are used to find axioms or theorems that explain given phenomena or to find phenomena as exemplars of axioms and theorems, and *rationality* of any given process means consistency of the statement to theories.

Then, analysis and synthesis can be logically associated with deduction and abduction [2]. First, we consider the following formula.

$$\boldsymbol{A} \vdash \boldsymbol{Th}$$

This formula means that under axioms ***A***, a set of theorems ***Th*** is proven. In this formula, finding theorems from the axioms is deduction, whereas finding axioms from given theorems is abduction. Axioms form the basis of a theory that can explain phenomena and facts, whereas ***Th*** consists phenomena and facts that are observed.

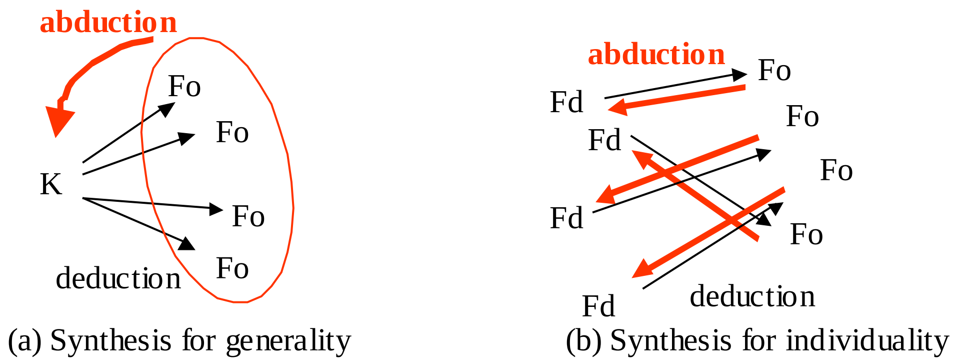

Figure 1: Pattern of Process for Synthsis

Since we usually have distinctions between knowledge and facts, we divide axioms into defined facts ***Fd*** (that should be given prior to reasoning without explanation) and knowledge ***K***.

$$\boldsymbol{K} \cup \boldsymbol{Fd} \vdash \boldsymbol{Th}$$

In contrast to ***Fd***, we can identify facts that should appear in theorem ***Th*** and be explained. They are called *observable facts* ***Fo***. With this distinction of ***Fd*** and ***Fo***, we can categorize abduction more precisely.

1. Finding ***K*** and ***Fd*** from a part of ***Th*** (=***Fo***)

2. Finding ***K*** from a part of ***Th*** (=***Fo***) and ***Fd***

3. Finding ***Fd*** from a part of ***Th*** (=***Fo***) and ***K***

These three different types of abduction play their own roles in different thought processes. For example, a scientific discovery process aims at obtaining knowledge ***K*** that should be *general* and therefore minimal, while given is a part of ***Th*** (=***Fo***) that is *individual* (see Figure 1(a)). Abduction in scientific discovery is types 1 or 2.Hypothetical knowledge (i.e., ***K***) proposed by abduction should be tested against observable facts ***Fo*** and this implies that deduction should be performed more often than abduction.

This stands in a sharp contrast to design in which most of synthesis is abduction of type 3. In a design process, the target is defined facts ***Fd*** that are *individual*, and given are knowledge ***K***

(1) Observation of phenomena
A phenomenon is observed as observations ***O***.
(2) Extraction of facts
Observed facts ***Fo*** are extracted from ***O***.
(3) Formation of hypotheses or selection of axioms
Fo can be used to reason out hypothetical axioms ***Kh***. In obvious cases, a set of known axioms ***Ke*** is selected instead.
(4) Assuming definition facts
Initial definition facts ***Fd*** are assumed. Together with ***Ke*** (or ***Kh***), this will be used to derive theorems ***Th***. Usually, ***Fd*** contain such known facts as boundary conditions and initial conditions.
(5) Derivation of theorems from axioms
Theorems ***Th*** are derived from ***Ke*** (or ***Kh***) and ***Fd*** deductively. It may break down the original problem (i.e., derivation of theorems) into smaller subproblems (the “divide-and-conquer strategy").
(6) Verification of theorems against facts
The derived theorems ***Th*** are tested against the observed facts ***Fo*** to check the explicability of the theorems. If $\boldsymbol{Th} \supseteq \boldsymbol{Fo}$, this test is satisfied. Then the theorems are said to explain the extracted facts and the choice of ***Ke*** (or ***Kh***) was appropriate. If $\boldsymbol{Th} = \boldsymbol{Fo}$, then ***Ke*** is complete. If $\boldsymbol{Th} \supset \boldsymbol{Fo}$, then $\boldsymbol{Th} - \boldsymbol{Fo}$ signifies unobserved facts or undiscovered facts in the future or past. If $\boldsymbol{Fo} - \boldsymbol{Th} \neq \emptyset$, then unexplained facts remain.
(7) Verification of theorems against other known axioms
The derived theorems ***Th*** are again tested against other known sets of axioms ***K’***. This test verifies if the theorems are compatible with ***K’*** or at least if they do not violate ***K’***. If the hypotheses obtained in step (3) pass tests (6) and (7), they become axioms.

(a) Analysis-oriented Thought Process

(1) Describing requirements
Requirements for the synthesis ***R*** are described as theorems.
(2) Extraction of requirements of interest
From ***R***, we only focus on interesting facts as ***Fo***.
(3) Selection of axioms
Axiom to be used is selected. Synthesis requires, various viewpoints to be considered. This means that the number or cardinality of ***K*** tends to be large.
(4) Derivation of solutions from requirements and axioms
Solutions ***Fd*** are derived as facts from ***K*** and ***Fo***. The basic reasoning is abduction logically, but other algorithms to arrive at solutions can be also used. The “divide-and-conquer strategy” might be used, but since the number (or cardinality) of ***K*** could be larger than analysis, trade-off and negotiation among different solutions are important.
(5) Derivation of theorems from axioms and facts
Theorems ***Th*** are derived from ***K*** and ***Fd*** deductively. This is the same as in the analysis oriented thought process. Deduction and the divide-and-conquer strategy are central.
(6) Verification of theorems against requirements
The derived theorems ***Th*** are tested against the requirements of interest ***Fo*** to check if the derived ***Th*** subsume the initial requirements ***Fo***; (i.e., $\boldsymbol{Th} \supseteq \boldsymbol{Fo}$). By doing so, we can check if the solutions ***Fd*** are satisfactory.
(7) Verification of theorems against other known axioms
The derived theorems are again tested against other known sets of axioms ***K’***. This test verifies if ***Fd*** (and accordingly ***Fo***) is compatible with not only ***K*** but also ***K’***.

(b) Synthesis-oriented Thought Process

Figure 2: Formalization of synthesis- and analysis-oriented thought processes

and observable facts ***Fo*** that is a part of ***Th***. Abduction must be performed as many as deduction (see Figure 1(b)), because both hypothetical facts (i.e., ***Fd***) and observable facts (***Fo***) are individual. Knowledge is also different in these two processes. While minimum knowledge ***K*** is desired in the scientific discovery process, a variety of knowledge ***K*** is required for design, because a variety of defined facts ***Fd*** need to be found by abduction.

From the discussions above, we can now formalize the synthesis-oriented and analysis-orient thought processes in Figure 2 [3]. While the analysis-oriented thought process (AOTP) is deduction-dominant, the synthesis-oriented thought process (SOTP) emphasizes the role of abduction. We can characterize SOTP in the following three dimensions in comparison with AOTP.

(1) Arbitrariness in problem definition: Enumeration of the requirements is less constraining than observation.

(2) Arbitrariness and complexity of viewpoints: Fewer axioms are preferred in AOTP while more axioms are preferred in SOTP.

(3) Complicated relationship between abduction and deduction: Both AOTP and SOTP need abduction and deduction. For instance, AOTP needs abduction to form hypotheses, while SOTP needs deduction to derive theorems in testing the facts.

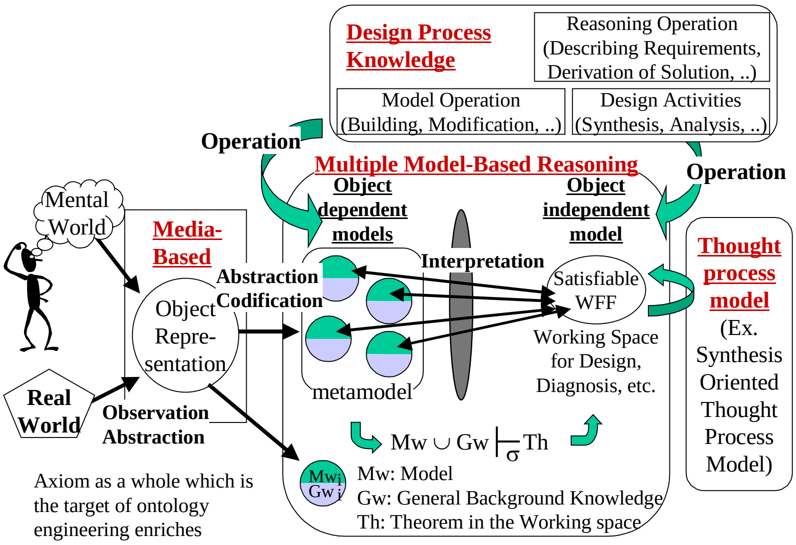

Figure 3: A computational Framework of Synthesis

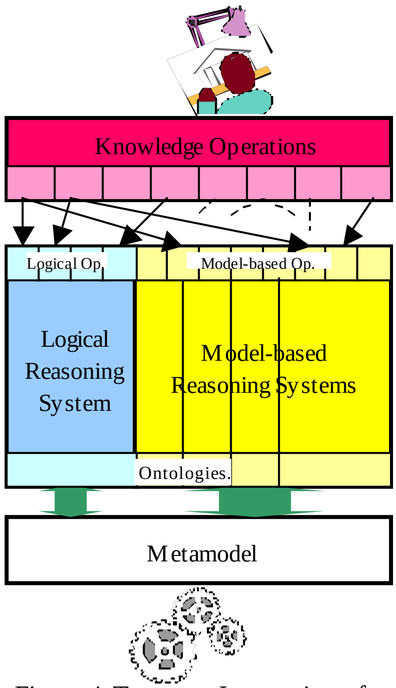

Figure 4: Two-way Integration of logical and model-based aspects

4 A computational framework for synthesis

The above discussion leads to the following two requirements for the framework to model synthesis.

1) Explicit controls for abductive and deductive reasoning processes over facts and knowledge are needed.

2) A variety of knowledge should be provided.

We propose a *computational framework of synthesis* to satisfy these requirements (see Figure 3). It has two main components; an object-independent model or a logical inference workspace in which abduction and deduction are explicitly controlled, and object-dependent models or a model-based reasoning workspace in which rich modeling knowledge is provided. The former is introduced to satisfy the first requirement, and the latter the second. Knowledge in a logical form has a clear syntax and therefore a sufficient computational power, but it lacks semantics that is fulfilled in real world. Model-based reasoning can offer such knowledge that is usually difficult to describe in logic.

The logical and model-based approaches are integrated as follows (see Figure 4). First, the metamodel mechanism manages multiple model-based reasoning systems and submits information about design objects to a common, logical workspace. This ensures smooth transfer of design object information between logical representation and models. Second, the designer's activities are represented by a set of knowledge operations that are decomposed into logical operations and modeling operations. The logical operations operate the logical inference workspace, while the modeling operations manipulate design object models through the metamodel mechanism. The multiple model-based reasoning performs assertion of logical formula in the logical workspace. In other words, a logical formula in the logical workspace is given a truth value only by referring to extra-logical models, if they are satisfiable. (Mathematically, this is called *model-theoretic* view as opposed to *proof-theoretic* view.) This explicit reference to non-logical models naturally integrates logic and models.

5 Knowledge operations

This paper is based on a knowledge-centric view of design activities. We provide a set of operations in the knowledge level to describe designers' activities. These knowledge operations are decomposed into logical and modeling operations, so that they become computable assuming the reasoning framework depicted in Figure 3. For example, *analysis activity* is a typical use of knowledge, and *activity of exchanging information with other designers* is a knowledge handling activity for knowledge acquisition.

As operations in the logical reasoning system, we define the following operations. First, we provide two types of object-level reasoning based on object knowledge; i.e., (l-1) deduction of properties of objects from objects (design solutions), and (l-2) abduction of objects from properties of objects. Here, requirement specifications are included in the properties of objects. Second, we also define meta-level reasoning as one up higher operations with operands such as objects and properties of objects in the object level; i.e., (l-3) setting of objects, (l-4) setting of requirement specifications, (l-5) setting of design knowledge, (l-6) consistency checking of knowledge, and (l-7) operations on the current set of design knowledge.

We provide the following eight operations as modeling operations. Among these eight, five operations concern individual models; i.e., (m-1) building a model, (m-2) reasoning with the model, (m-3) modification of knowledge, (m-4) modification of the model, and (m-5) reference to the model. The rest three are operations for maintenance of multiple models; i.e., (m-6) introduction of model-based systems, (m-7) selection of model-based systems, and (m-8) maintenance of consistency among different models.

We define the following seven knowledge operations as combinations of logical and modeling operations. First, to manage knowledge, there are five operations; i.e., (k-1) knowledge/information acquisition, (k-2) knowledge/information reorganization, (k-3) information confirmation, (k-4) conflict resolution, and (k-5) knowledge/information revision. Second, to utilize knowledge, there are two operations; i.e., (k-6) solution synthesis and (k-7) object analysis. Below, we describe these knowledge operations in detail.

(k-1) *Knowledge/Information acquisition:* The objective is to acquire knowledge and information related to the problem. There are two types of knowledge acquisition. One is to introduce a knowledge source and is formalized as introducing a new model-based system (m-6). The other is to add a new piece of knowledge to a particular model-based system and is formalized as modification of the knowledge base of a model-based system (m-3). The corresponding logical operation is the operation on the current set of design knowledge (l-7).

(k-2) *Knowledge/Information reorganization*: The objective is to reorganize knowledge and information for a task. Knowledge reorganization is a process to reorganize and maintain a set of model-based systems that are used for a task. So, this process is formalized as selection of (one or more) model-based systems (m-7) or maintenance of models in different model-based systems (m-8). The corresponding logical operation is an operation on the current set of design knowledge (l-7).

(k-3) *Information confirmation*: The objective is to confirm information in one knowledge source by testing it against another information source. In this operation, the designer confirms information in the logical level through mappings between models in the object dependent level and the logical level. The metamodel mechanism should provide such mappings and it is formalized as reference to the model (m-5), and setting of design knowledge (l-5), and consistency checking of knowledge (l-6).

(k-4) *Conflict resolution*: The objective is to resolve conflict among different model-based systems; for example, one modeling system says that value of an attribute of the design object should be 10, while another says 10.5. The logical reasoning system detects this type of conflicts by integrating mode-based systems and solves by modifying relationship among different mode-based systems or models in particular model-based systems (see k-5). It corresponds to maintenance of consistency among different models (m-8) and consistency checking of knowledge (l-6).

(k-5) *Knowledge/Information revision*: The objective is to revise knowledge or information to keep consistency with multiple models. It corresponds to modification of knowledge in model-based systems (m-3) or modification of models (m-4), setting of design knowledge (l-5) and setting of requirement specifications (l-4).

(k-6) *Solution synthesis*: It is synthesis in the narrow sense or abduction, i.e., to suggest a new solution for the problem. Following SOTP in Figure 2 (b), first, the selection of axioms for synthesis is formalized as selection of a model-based system (m-7) and description of the problem is formalized as building a model (m-1). Then, the designer proposes new solution candidates by reasoning about a model (m-2). From the logical aspect, it corresponds to setting of requirement specification (l-4), setting of design knowledge (l-5), and abduction of objects from properties of objects (l-2).

1. Derive the neighborhood system of a solution candidate from one axiom set A_1.
   a. Set requirements that can be treated by axioms A_1 as theory (Th_1) in formula (1).
      i. $A_1' \cup F_1 \mid\!-\sigma\ Th_1$
      ii. $\{e_1 \rightarrow p_1, e_1 \rightarrow p_2, .., e_2 \rightarrow f_1, e_2 \rightarrow f_2, ..\} \cup F_1 \mid\!-\sigma\ \{p_1, .., f_1, ..\}$
      where e_i is an abstract entity concept, p_i is an attribute concept, and $f_{i,}$ is a function concept.
   b. Derive $F_1=\{e_1, e_2, \ldots\}$ by abduction (purely logically) with the closed world assumption.
   c. Analyze the neighborhood system of F_1 in the attribute space and the function space by deduction using a modeler that corresponds to Axioms A_1. This will enrich Th_i.
2. Apply previous procedures with another set of axioms and make the attribute information and function information richer.
3. Compute $F_n \cap F_{n+1} \cap \ldots$ for narrowing the solution space to reach a solution.

Figure 5: Algorithm for model-based abduction

(k-7) *Object analysis*: After proposing new solution candidates, the designer should test the candidates against other knowledge sources. This operation follows AOTP shown in Figure 2 (a). Model-based operations for this are the same to those for solution synthesis (k-6). The difference is the reasoning mode in model-based systems, i.e., deductive or abductive. From the logical aspect, the first and second steps are the same to solution synthesis (k-6), and the third step is to deduction of properties of objects (l-1) instead.

6 Model-based abduction

The logical and model-based approaches are mutually integrated by knowledge operations that can associate operations in the both workspaces. This integration enables new types of reasoning smoothly combining different kinds of ontologies that are embedded in various model-based reasoning systems. Here, logical level abduction is enhanced by model-based abduction. Model-based abduction is an inference mode in which models are operated and inferred to incorporate new extra-logical statements into logical reasoning. In other words, various knowledge bases that are based on different ontologies cooperate each other to arrive at richer design solutions than pure logical level abduction. Model-based abduction can provide various methods based on particular models that are used as heuristics in design such as generate and test, catalog retrieval, case-based reasoning, computational model, and optimization techniques instead of pure logical abduction.

The process of model-based abduction is formalized as iterative exploration of candidates by applying different theories (see Figure 5). In order to apply this algorithm, we first need relationships between the ontology of each model-based system and the metamodel ontology to translate representation of a solution in one aspect to another. We also need semantic categorization of facts into entities, functions, and properties in those ontologies to interpret knowledge on models as formula shown in Figure 5.1.a.ii, and distribute representation of a solution into either facts or theorem. We discussed the details of model-based abduction in [4].

7 Conclusions

We discussed the nature of synthesis in design and showed a model that combined both logical and model-based representations. By doing so, we could clarify the over-all structure of the synthesis-oriented process as well as a general computational framework for modeling of synthesis that performs new types of inferences, such as knowledge integration by model-based abduction.

Synthesis is not mere abduction but an appropriate combination of abduction and deduction. Since actual abduction and deduction are deeply dependent on domain knowledge, we should model synthesis with knowledge operations. The proposed computational framework allows integration of logic level reasoning and model-based reasoning that have different knowledge bases with different ontologies.

We conducted case studies about how our framework can be applicable to actual design processes [5][6]. We could explain core parts of the processes including analysis and synthesis, as well as preparatory parts such as knowledge acquisition. Based on this result, we are now building a prototype system and testing it [7][8]. The system is an environment to provide designers with various kinds of design knowledge and abductive capabilities. This research is supported by the Research for the Future Program of the Japan Society for Promotion of Science in the project of “Modeling of Synthesis,” JSPS-RFTF96000701

References

[1] Hartshorne, C. and Weiss, P. eds., 1931-1935, “The Collected Papers of Charles Sanders Peirce, ” Vol. I-VI, Harvard University Press.

[2] Takeda, H., Veerkamp, P., Tomiyama, T. and Yoshikawa, H. “Modeling design processes”, AI Magazine, 1990, Vo.11, No. 4, pp.37-48.

[3] 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,” ICED ’97, 1997, pp. 97-100.

[4] Yoshioka, M. and Tomiyama, T., “Model-based Abduction for Synthesis”, Proc. 2000 ASME Design Engineering Technical Conference & Computers and Information in Engineering Conference, ASME, DETC2000/DTM-14553 (CD-ROM), 2000

[5] Tsumaya, A., Takeda, H. and Tomiyama, T., "A Synthesis-Related Analysis Method of Design Protocol," ICED ’99, 1999, Vol. 3, pp. 1949-1952.

[6] Tsumaya, A., Nomaguchi, Y., Yoshioka, M., Takeda, H., Murakami, T. and Tomiyama, T., “Verification of a Model of Synthesis -The methods for verification and results-“, Proc. ICED 2001, (to appear)

[7] Yoshioka, M. and Tomiyama, T. “Toward a Reasoning Framework of Design as Synthesis,” Proc. 1999 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, ASME, DETC99/DTM-8743 (CD-ROM), 1999.

[8] Yoshioka, M., Nomaguchi, Y. and Tomiyama, T, "Proposal of an Integrated Design Support Environment Based on the Model of Synthesis", 27th Design Automation Conference in 2001 ASME Design Engineering Technical Conference & Computers and Information in Engineering Conference, 2001 (to appear).

Hideaki Takeda
National Institute of Informatics, 2-1-2 Hitotsubashi Chiyoda-ku Tokyo 101-8430, Japan, Tel: +81-3-4212-2543, Fax: +81-3-3556-1916, E-mail: takeda@nii.ac.jp, URL http://www-kasm.nii.ac.jp