

1(leave this here)

Design Knowledge Collection by Modeling

M. Yoshioka

Research and Development Department

National Center for Science Information Systems, Japan

Ootsuka3-29-1, Bunkyo-ku, Tokyo, 112-8640, Japan.

Telephone: +81-3-3942-7003 Fax: +81-3-5395-7064

yoshioka@rd.nacsis.ac.jp

T. Sekiya, T. Tomiyama

Research into Artifacts, Center for Engineering (RACE)

The University of Tokyo

Komaba 4-6-1, Meguro-ku, Tokyo 153-8904, Japan

Telephone: +81-3-5453-5888, Fax: +81-3-3467-0648

{sekiya, tomiyama}@race.u-tokyo.ac.jp

Abstract

Since design is a process in which designers use various kinds of knowledge and it is an open-ended problem, it is difficult to prepare all necessary knowledge before design. Also, due to advances of technology, design knowledge is largely fragmental and compartmentized. This makes the communication among design experts or design agents difficult. Due to these, a knowledge sharing mechanism among agents based on a common ontology is needed even for collecting design knowledge. In this paper, first we analyze the nature and structure of design knowledge. Then, we propose a knowledge collection methodology through design object modeling and verify the method with an example of the suspension design for a motorcycle. Finally, we identify knowledge that should be prepared before the design process and that can be collected during modeling processes.

Keywords

**design knowledge acquisition, computational methods of design,
knowledge reuse**

1 INTRODUCTION

In a design process, a designer uses various kinds of design knowledge, from common sense knowledge about the physical world to domain specific knowledge like theories in physics. In addition, since design is an open-ended problem, it is difficult to predict what kind of design knowledge is required before starting a design process. Therefore, to support a design process with knowledge base technologies, the system is required to have a knowledge collection mechanism through modeling of a design object.

Also, the advances of technology resulted in a set of sophisticated, but compartmentized and fragmental bodies of design knowledge. For instance, it is difficult for an average engineer to understand the entire knowledge assumed in a Finite Element Analysis (FEM) system for structural analysis. Because a large body of design knowledge is embedded in these modeling systems, such a system becomes a blackbox for designers and should be regarded as a design agent. Therefore, the system we discuss in the paper has to facilitate communication among these design agents based on a common ontology.

There are many research efforts on design knowledge acquisition from the design process (AIEDAM, 1996). However, most of them only focused on improving quality of design knowledge and they do not address issues about collecting and cooperatively using design knowledge. In contrast, there are several research efforts on knowledge sharing such as CYC (Guha, 1994) and PACT (Cutkosky, 1993), that address these issues. However, they assume that knowledge should be prepared before design and the collection issues are missing.

We have been working on a project of building a large scale engineering knowledge base (Ishii, 1995) as a kernel of the Knowledge Intensive Engineering Framework (KIEF) to support knowledge intensive engineering (Tomiyaama, 1994). Knowledge intensive engineering is a new way to conduct a variety of engineering activities, including design, manufacturing, operation, maintenance, and recycling, in which knowledge is used in a flexible and integrated manner. Figure 1 shows the architecture of KIEF. In this architecture, the concept base provides vocabulary commonly used for the entire knowledge base and further the metamodel mechanism to manage a variety of models.

This paper describes our approach to collecting design knowledge through modeling of design objects on KIEF that has a collaborative design agent architecture. In Section 2, we discuss the nature of design knowledge based on the result of the project Modeling of Synthesis (Tomiyaama, 1997) and propose a model of its layered structure. Section 3 briefly reviews our previous work by illustrating KIEF and the pluggable metamodel mechanism. The nature of design knowledge (knowledge being a blackbox as a result of compartmentization) requests that deep, domain dependent knowledge can only be dealt with by design agents, i.e., external modelers to be plugged into the metamodel mechanism. These agents communicate and share knowledge with each other based on the common ontology. To do so, Section 4 discusses our methodology to collect design

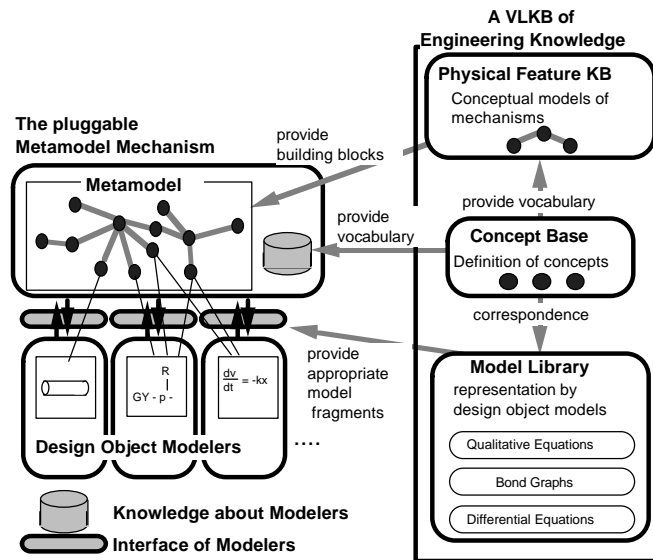


Figure 1 The Architecture of the KIEF

knowledge. However, due to the agent architecture, the knowledge we need to collect boils down to the ontology level knowledge and the modeling knowledge. The former and half of the latter must be collected and organized before design. The rest of the latter can be collected during design. Thus, we propose in this section “design knowledge collection by modeling.” In Section 5, we demonstrate the proposed methodology with a case study. We also discuss the reusability of the collected knowledge and the ontological knowledge that should be prepared before collecting the modeling knowledge. Section 6 concludes this paper.

2 DESIGN KNOWLEDGE

In this section, we discuss the nature of design and propose a layered structure model of design knowledge.

2.1 Nature of Design Knowledge

In the project Modeling of Synthesis, we aim to build a scientific model of synthesis (Tomiya, 1997). This project, funded by the Japan Society for the Promotion of Science (JSPS-RFTF 96P00701), is a five years project and it started in 1996. Our first result from the project is a hypothetical reasoning model of synthesis (Figure 2).

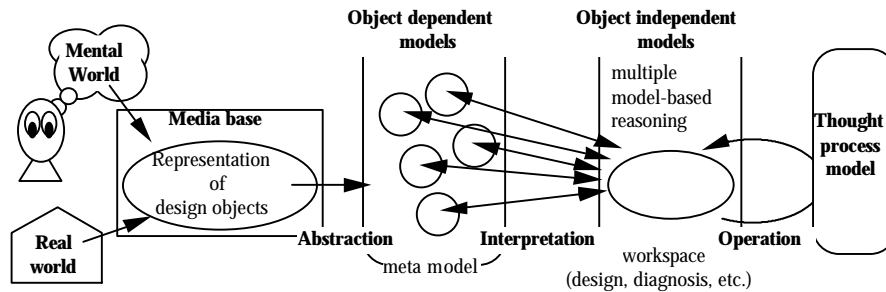


Figure 2 A Hypothetical Reasoning Model of Synthesis

One important feature of the model is “multiple model-based reasoning”. In a design process, it is better to generate more design candidates to justify that a selected candidate is the best solution. To do so, it is necessary to handle a wide variety of design knowledge in the model of synthesis. However, in the design process, designers often suggest new solutions which are different from the well-known solutions. This implies that typically design is an open-ended problem that addresses the frame reference problem. It is difficult to predict what kind of knowledge is required before starting a design process.

For handling a wide variety and a huge amount of design knowledge, traditionally engineering was organized as micro theories capable of generating and verifying a new solution. These micro theories tend to be dealt with just as a black box. For example, in the design of a robot, a micro theory about sensors can be introduced. In this case, however, designers do not need to know details about behavioral principles of sensors. All they should know for using a sensor is the relationship between its input and output.

Therefore, the knowledge base system for KIEF is requested to have a capability of introducing multiple micro theories. Each of these micro theories must be associated with a model and knowledge about how to use them. The metamodel mechanism we proposed (Yoshioka, 1997) can deal with multiple micro theories by incorporating multiple design object modelers.

2.2 Structure of Design Knowledge

The discussion above is based on the assumption that design knowledge has a layered structure. Figure 3 depicts its model. The lower layer describes the syntax of the knowledge. The semantic layer is divided into three sublayers. The ontology layer provides the common vocabulary for describing the design knowledge and forming a basis for design object modelers as agents to communicate each other. The middle modeling knowledge layer contains knowledge about modelers and model building knowledge to describe, e.g., how to use each of those modelers, such as input and output conditions. In the upper layer, each modeler encapsulates its knowledge.

Based on this knowledge structure, it is necessary to describe the syntax layer, ontology layer, and concept description layer for cooperatively using multiple design object modeling systems.

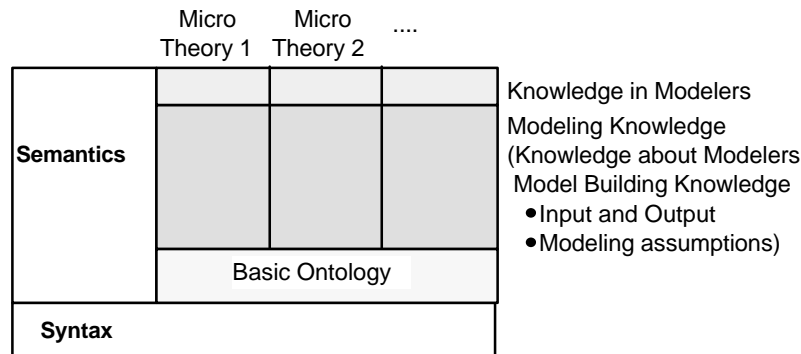


Figure 3 The Model of Design Knowledge Structure

3 THE KNOWLEDGE BASE SYSTEM FOR KIEF

In this section, we summarize our preliminary research (Ishii, 1995) to build a knowledge base system for KIEF. First, we describe the architecture of the knowledge base. Then we describe the multiple modeling environment of KIEF.

3.1 Architecture of Knowledge Base

Figure 1 depicts the three component architecture of the knowledge base system for KIEF. The middle component, called "concept base," contains physical concepts. This corresponds to the basic ontology layer in Figure 3. In the concept base, physical concepts are categorized into the following five types.

- Entity

An entity represents an atomic physical object. Entities include such as mechanical parts and electric devices, and are organized in an abstract-concrete hierarchy. For example, a "worm gear" is a subclass of a "gear." The hierarchy allows multiple inheritance. In addition, entities have descriptions of "has" relationships with related physical properties.
- Relation

Relations represent relations among entities to denote static structure. They include relations between physical objects such as "connection" and "on." They are also organized in an abstract-concrete hierarchy.
- Attribute

An attribute is a concept attached to an entity and takes a value to indicate the state of the entity, such as "position" and "temperature." Attributes also have a description about differential relationships with other attributes (e.g. "velocity" is a differential of "position").

- Physical property
A physical property represents generic characteristic of entities such as “elastic” and “magnetized.” A physical property is associated with a set of attributes that indicates degree of the property.
- Physical phenomenon
A physical phenomenon designates physical laws or rules that govern behaviors. A physical phenomenon is defined by the following slots.
 - Name of the phenomenon.
 - Super (or abstract) physical phenomena as described in Entity.
 - Related physical phenomena, entities, and attributes with respect to the phenomenon.
 - Physical rules for representing relationships among attributes. These rules are correspondent to the lower component in the three components architecture in Figure 3.

Figure 4 is an example description about gear transmission. The left side contains a “physical feature,” such as a worm gear pair, which represents a combination of a set of entities and relations among the entities, and physical phenomena causally related to the entities. Physical features are used as building blocks for a physical model on this system. Physical concepts in the concept base provide a vocabulary to build the physical feature in this basic ontology layer.

The right side component contains physical laws. This component corresponds to the modeling knowledge layer in Figure 3. The names of physical laws are stored in the *physical rule KB* with the relationship with attributes to consider. Model fragments for building a model with various model representations are stored in the *model libraries* with the relationship to physical laws.

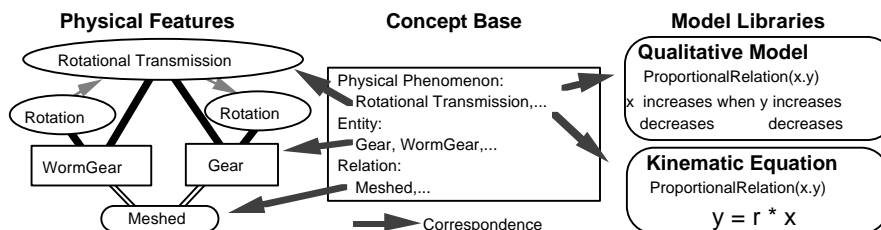


Figure 4 Describing the Knowledge about Gear Transmission in KIEF

3.2 Pluggable Metamodel Mechanism

We have already proposed the idea of a multiple modeling environment called a *pluggable metamodel mechanism* (Yoshioka, 1997). The metamodel mechanism maintains relations and consistencies among these design object models by using a *metamodel*, which represents relationships of physical concepts used in various design object models. This framework also can deal with existing external modelers (e.g., solid modeler).

The modeling process on the metamodel mechanism has the following four steps.

1. Constructing a primary model.
The designer constructs a primary model of the design object by combining physical features.
2. Computing a metamodel.
The metamodel mechanism finds out possible physical phenomena that may occur on the design object with qualitative reasoning.
3. Generating an external model.
The designer generates an aspect model for evaluation by an external modeler. First, the metamodel mechanism selects related concepts to the aspect and relations among the concepts from the metamodel. By doing so, a conceptual model for generating an external model is constructed. After this, the metamodel mechanism translates the conceptual model to an external model by using model fragments and attribute data represented in the metamodel.
4. Evaluating the external model.
Finally, the designer evaluates the generated aspect model with the external modeler.

4 APPROACH TO COLLECTING DESIGN KNOWLEDGE

This section presents our approach to collecting design knowledge by modeling. First we discuss the kind of knowledge to be collected and then we propose a methodology to collect design knowledge.

4.1 Discussion about Knowledge to be Collected

As discussed in Section 2.1, since design is an open-ended problem and it requires a large amount of knowledge, it is difficult to predict and prepare whole knowledge needed for a design before the design. Section 2.2 pointed out that there are two types of knowledge we can find out in commercial computational tools developed and used for design. One is a type of knowledge about how to use these tools and how to build models for them. The other is a type of knowledge embedded within these tools. The former corresponds to the modeling knowledge in Figure 3, while the latter to the knowledge in the modelers and does not have to be collected or described.

The former knowledge (i.e., modeling knowledge) has descriptions about models. More specifically, it includes knowledge about physical concepts that a modeler can deal with (knowledge about modelers) and about the model building process (model building knowledge), using the ontology level knowledge in the concept base. However, to collect such a type of knowledge is inevitably dominated by modeling “know-hows.” This requests skills and experiences and cannot be done easily.

The metamodel mechanism can basically be regarded as a collaborative multiple agent architecture. Guha and Lenat (1994) wrote that there were two forms of collaborative multiple agent architecture:

- *exo-*: Impose rigid standards “from the outside.”

- *endo-*: Inject at least partial understanding of the meaning of the information which each agent is working on, and emitting.

Obviously, it is difficult to set standards as in the former approach and they selected the latter. For the same reason, we also selected the latter form for collaboration. This approach allows the designer to describe, during modeling, information about what kind of physical concepts is used in a model for partial understanding of other models. This led us to a new design knowledge collection method, “design knowledge collection by modeling,” which is discussed next.

4.2 Knowledge Collection Method by Modeling

As described above, we focus on the knowledge collection of modeling knowledge. This modeling knowledge includes two types. One type is knowledge about modelers. For example, suppose a motion analysis system that can analyze motion of mechanisms. While actual concepts that the system can deal with are mathematical equations, what we think we deal with are positions, velocities, and forces of various mechanical components, such as linkage and cam that are described in the concept base. This suggests that we need to have mapping knowledge in the model libraries between mathematical parameters and kinematic concepts. In Figure 4, model fragments play the role of the mapping knowledge. The entire set of model fragments for a particular modeler defines the knowledge about the modeler. To collect this type of knowledge, we need to define concepts and model fragments that can be dealt with the modeler in the concept base as well as data accessing methods.

The other type of modeling knowledge describes more heuristic oriented knowledge about the model building process. This knowledge includes, for instance, relationships between input and output parameters of the modeling process, and modeling assumptions. Modeling assumptions are knowledge about, for example, minor parameters that can be neglected during the modeling process. This type of knowledge is also written with concepts in the concept base and stored in the metamodel as mapping rules. Before collecting this type of knowledge, we need to define model libraries that corresponding to the knowledge about modelers in Figure 3.

These two types of knowledge are largely dependent on specific modeling and also heuristics. Therefore, these can only be collected at the time of modeling, and recording a modeling process becomes equivalent to design knowledge collection. However, such knowledge does not have enough reusability. For instance, we should check the use of existing physical concepts in the collected knowledge to measure the reusability of the collected knowledge. New knowledge has more chances to reuse, only if the collected new knowledge is associated with existing concepts. If the new knowledge is associated with problem specific concepts, abstract concepts should be created to increase their reusability.

5 CASE STUDY

This section illustrates a case study to demonstrate the feasibility of our approach to collecting design knowledge by modeling. The system was implemented in Objectworks\Smalltalk running on a Sun workstation. The case study is a design of suspension for a motorcycle.

5.1 Design of a Suspension for Motorcycle

Before conducting the case study, we have collected physical concepts to be stored in the concept base as listed in Table 1.

Table 1 Knowledge in the Concept Base (Ishii, 1995)

Physical concepts	Number
Entities	200
Relations	40
Physical phenomena	150
Attributes	280
Physical properties	80
Physical laws	300

In modeling the motorcycle, 31 physical concepts were needed to build a primary model, a metamodel, and an external model for evaluating dynamic behavior. During this process, four new physical concepts and one physical feature were added. These concepts were “damper” (Entity), “damping” (Physical Phenomenon), “damping coefficient” (Attribute), and “damping law” (Physical Laws). A physical feature, DamperMechanism, was created to represent the way to use a damper.

In addition, we created a geometric model of the motorcycle with DESIGNBASE¹ which is a commercial solid modeler (Figure 5). To deal with the information in the solid modeler, we needed to add modeling knowledge for the solid modeler. In this case, we added a new physical concept “solid” (Attribute) to the concept base. We also defined the knowledge about solid modelers as follows.

- Knowledge about the solid modeler.
The solid modeler can deal with “solid” and other geometric attributes. These concepts were stored in the concept base, as well as associated operations of the modeler.
- Relationships between input and output parameters.
Geometric attributes such as “length” can be calculated as distance between two points by using knowledge in the solid modeler.
- Modeling assumption.

¹ DESIGNBASE is a trademark of RICOH Company.

To use the information in the solid model, the designer input the information about the correspondence between each solid in solid model and the entity concept in metamodel.



Figure 5 Solid Model of the Motorcycle

Since geometric attributes like “length” had already been defined in the concept base, data in the solid model could be shared with other modeling systems. For instance, the kinematic equation modeler (Figure 6) could obtain “length” information from the solid modeler.

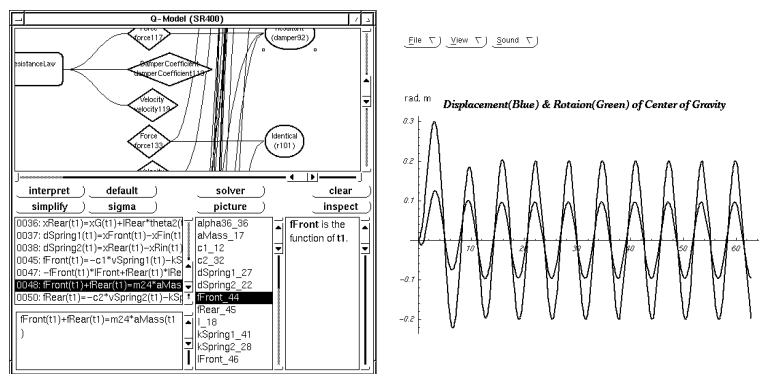


Figure 6 The Kinematic Equation Modeler and the Damper Behavior

5.2 Discussion

In the example of suspension design, following two types of knowledge collection activities were conducted.

- Collection of ontological knowledge
For adding a new entity concept to the concept base, it is requested to define the relevant physical phenomena, physical laws, attributes, and physical features with relationships with the existing concepts. In this case study, “damping law” represents the relationship between the new attributes such as “damping ratio” and the existing attributes such as “force.” “Damping law” was used to simulate the behavior of suspension combined with existing knowledge such as “Hooke's law.” This knowledge collection activity was conducted at the ontology level to collect more or less generic knowledge to be stored in the concept base.
- Collection of modeling knowledge.
In the case study, a new geometric model of the motorcycle was generated. This requested to add a new concept “solid” in the concept base and data

accessing methods to the geometric modeler were defined in the model libraries. In addition, we defined a method that translates the new added attribute “solid” to existing attributes such as “length” to share information with other existing modeling systems. Also, in translating the geometric data to the kinematic data, special mapping rules had to be added to the metamodel. By doing so, we created a modeling assumption that the wheels of the motorcycle were modeled as disks (i.e., a special case of the cylinder) in the geometric model by using the “matching browser.”

From this case study, it is clearer that collecting ontological knowledge can also be done during modeling. Knowledge such as “Hooke's law” is general enough to be reused. In contrast, obviously modeling knowledge which is model-specific can only be defined and collected at the time of modeling.

While this case study demonstrated the feasibility of design knowledge collection by modeling, it is desirable to prepare as much ontology level knowledge as possible. At this moment, we have collected physical laws based on a book about physical laws and effects (Hix, 1958) and we believe that such an effort makes it possible to cover sufficient basic concepts.

6 CONCLUSION

In this paper, we proposed a methodology to collect design knowledge by modeling during design. This methodology was derived from discussion about the nature and structure of design knowledge. Design knowledge can be structured into three layers, i.e., ontological knowledge, modeling knowledge, knowledge in modelers. Modeling knowledge can be further categorized into knowledge about modelers and knowledge about model building. While it is possible to collect ontological knowledge and knowledge about modelers can deal with before design, knowledge about model building can only be collected during design. We also validated the method by demonstrating a case study of suspension design for a motorcycle.

For future work, we plan to collect more knowledge through example design for investigating scale effects and to plug more commercial modeling systems into KIEF. The system level standardization based on knowledge interchange format such as KIF (Genesereth, 1992) is also a further research issue, when the number of external modeling systems increases and knowledge must be well organized.

ACKNOWLEDGEMENT

The authors would like to thank Dr. Takashi Kiriyama, Dr. Yasushi Umeda, Dr. Yoshiki Shimomura and other colleagues in our group at the University of Tokyo who helped us in the discussions and implementations of the ideas presented in this paper. This research was partially funded by the “Modeling of Synthesis” project of the Japan Society for the Promotion of Science’s Research for the Future Program under the contract number, JSPS-RFTF 96P00701.

REFERENCES

- AIEDAM (1996) Special Issue: Machine Learning in Design, in *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, Vol. 10, No. 2.
- Cutkosky, M.R., *et al.* (1993), PACT: An experiment in integrating concurrent engineering systems. *IEEE Computer*, Vol. 26, No. 1, 28-37.
- Genesereth, M.R., and Fikes, R.E. (1992) Knowledge interchange format, version 3.0 reference manual. Technical report logic-92-1, Department of Computer Science, Stanford University, Stanford.
- Guha, R.V. and Lenat, D.B. (1994) Cyc: Enabling agents to work together. *Communications of the ACM*, Vol. 37, No. 7, 126-142.
- Hix, C.F. and Alley, R.P. (1958) *Physical Laws and Effects*. John Wiley & Sons, London.
- Ishii, M., Sekiya, T., and Tomiyama, T. (1995) A very large-scale knowledge base for the knowledge intensive engineering framework, in *Toward Very Large Knowledge Bases*, IOS Press, Amsterdam, Oxford, Tokyo, 123-131.
- Sekiya, T. and Tomiyama, T. (1997) Case studies of ontology for the knowledge intensive engineering framework, in *Knowledge Intensive CAD*, Vol. 2, Chapman & Hall, London, 139-156.
- Tomiyama, T. (1994) From general design theory to knowledge-intensive engineering, in *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, Vol. 8, No. 4, 319-333.
- Tomiyama, T., *et al.* (1997) The Modeling of Synthesis - From the Viewpoint of Design Knowledge. *Proceedings of International Conference on Engineering Design, ICED 97*, Vol.3, 97-100.
- Yoshioka, M., Tomiyama, T. (1997) Pluggable Metamodel Mechanism: A framework of an Integrated Design Object Modelling Environment, in *Proceedings of CACD'97*, 57-70, Lancaster University, Lancaster.

BIOGRAPHY

Masaharu Yoshioka has been a Research Associate at National Center for Science Information Systems (NACSIS) in Tokyo, since 1996. He received his doctor's degree in precision machinery engineering from the Graduate School of the University of Tokyo in 1996. His research interest includes design process modeling, functional modeling, and knowledge intensive engineering.

Dr. Tetsuo Tomiyama is Professor at RACE of the University of Tokyo. He was an Associate Professor at the Department of Precision Machinery Engineering, the University of Tokyo, from 1987 to 1998. From 1985 to 1987, he worked at the Centre for Mathematics and Computer Science in Amsterdam. He received his doctor's degree in precision machinery engineering from the Graduate School of the University of Tokyo in 1985. His research interest includes design theory and methodology, knowledge intensive engineering, applications of qualitative physics, large scale engineering knowledge bases, soft machines (self-maintenance machines and cellular machines), and manufacturing paradigms.