A DESIGN PROCESS MODEL WITH MULTIPLE DESIGN OBJECT MODELS

Masaharu Yoshioka, Masahiko Nakamura, Tetsuo Tomiyama, Hiroyuki Yoshikawa

Department of Precision Machinery Engineering
Faculty of Engineering, The University of Tokyo
Hongo 7-3-1, Bunkyo-ku, Tokyo, 113, Japan
telephone: +81-3-3812-2111 (ex. 6481)
internet: yoshioka@zzz.pe.u-tokyo.ac.jp

ABSTRACT

An intelligent CAD system is not merely a set of intelligent design tools, but rather it must be an intelligent, integrated design environment. This requests it must be equipped with a large scale knowledge base in which design knowledge is intensively and systematically stored. To do so, design knowledge must be systematically formalized, made computable, and organized. The present paper investigates fundamental issues of systematization of design knowledge. Design knowledge has two categories; i.e., design object knowledge and design process knowledge. In our previous reports, we have formalized a computable design process model to represent design process knowledge and the concept of metamodel dealing with multiple design object models. This paper proposes a framework to achieve harmonized combination of these two. In this framework, the computable design process model navigates the metamodel mechanism to generate and modify models of the design object. We describe a prototype system and illustrate an example of ship design.

1 Introduction

Intelligent CAD systems aim at serving designers as a knowledge intensive intelligent design environment (Tomiyama, 1991). At the University of Tokyo, we have been working on the IIICAD (Intelligent Integrated Interactive CAD) project to develop such a knowledge intensive intelligent CAD system (Xue *et al.,* 1992) that must be equipped with a large scale knowledge base (Lenat and Guha, 1989). Our experiences with building such a large scale knowledge base (Kiriyama *et al.,* 1992) tells that a process called systematization of design knowledge which is indispensable even only to collect knowledge (Tomiyama *et al.,* 1992). This means that design knowledge must be systematically formalized, made computable, and organized in the knowledge base to achieve flexible, efficient, effective reuse and sharing of knowledge.

There are two categories of design knowledge (Veth, 1987, Tomiyama *et al.,* 1989). One is design object knowledge that is fairly well studied and understood. Examples of this research include geometric modeling and product modeling. The other is design process knowledge that is less clarified than the former, but research effort spent in this area has achieved some significant theoretical results. For example, Yoshikawa (Yoshikawa, 1981) proposed General Design Theory that discusses static aspects of design based on axiomatic set theory. Ullman et al. (Ullman *et al.,* 1988) conducted an empirical study on design and showed that the protocol analysis method was useful.

We have proposed a computable design process model based on analysis of design protocols (Takeda *et al.,* 1992). This model has two levels of inferences; i.e., an action level and an object level. In this model, a design object is described in first order predicate logic in the object level, while a design process is represented and navigated by a sequence of design operations including modification and evaluation of the design object in the action level. In other words, the design process knowledge is operational knowledge about the design object knowledge.

A mechanical engineering design process requires various kinds of design object models, such as geometric model, kinematic model, and finite element model. The designer has to deal with these "aspect" models and to keep consistency among them. We have also proposed a framework for integrating various kinds of aspect models. This framework, called metamodel mechanism, based on a qualitative physics reasoning system handles them and maintains consistency among aspect models (Tomiyama *et al.,* 1989, Kiriyama *et al.,* 1991).

This paper discusses a unified framework to arrive at harmonized combination of the design process knowledge represented by the computable design process model and the design object knowledge organized by the metamodel mechanism. We assume in this paper that the design process knowledge is knowledge to operate aspect models; for example, how to select an appropriate aspect model for evaluation, how to add detailed information to it, how to revise and modify it, and so on. We use the metamodel mechanism that integrates various aspect models, including a functional model, to represent the object level operations in the design process model.

The paper is divided into five sections. Section 2 illustrates a computable design process model and the framework of the metamodel. We also introduce a new technique to deal with functional information. Section 3 discusses an extension of the computable design process model to deal with multiple aspect models. Section 4 demonstrates a prototype system based on the extended design process model. Section 5 compares our approach with other related work and Section 6 concludes the paper.

2 Models of Design Processes and Design Objects

2.1 The Computable Design Process Model

Takeda et al. (Takeda *et al.*, 1990) observed from design experiments that a design object evolves in a step-wise manner. One step consists of five elementary stages; i.e., *awareness-of-problem, suggestion, development, evaluation,* and *conclusion.* These five stages form a cycle called *design cycle.* A design cycle solves a small design problem or divides it into smaller subproblems. This observation from design experiments led to a cognitive model of design processes which is repetition of design cycles.

We have formalized the design cycle in logic (Takeda *et al.*, 1992)(Figure 1). Suggestion in a design cycle is a stage in which the designer suggests new candidate design solutions for a design problem, and it can be logically simulated by a process of *abduction.* Abduction, proposed by Peirce (Peirce, 1932), is a type of inference to find axioms from given consequences that can be derived from the axioms. Development and evaluation are stages in which the designer figures out the properties of a candidate design solution and evaluates it with respect to the specifications. Development is much similar to simulation to see how a candidate behaves under a certain circumstance. We can formalize development and evaluation as processes of *deduction.*

A designer often faces inconsistency during the design process. In most cases, inconsistency means that knowledge is incomplete in that sufficient descriptions about its applicability are not provided: It is practically impossible to describe all applicable situations. Circumscription(McCarthy, 1980) can find implicit descriptions of the knowledge to restrict its applicability and modifies the knowledge to resolve the inconsistency.

The action level inference guides the entire design process and is considered meta level inference. In the action level, actions to be taken in the next step are derived from the current situation of the object level. This computable design process model is shown in Figure 2. Knowledge used in the object level is chosen by the action level inference; i.e., the action level knowledge selects appropriate object level knowledge. Once a design cycle successfully completes, knowledge for the next design cycle is made available by the action level inference.

2.2 The Metamodel Mechanism

A design object can usually be represented with various aspect models. that encompass a wide variety in different dimensions, i.e., domains of concepts for modeling, levels of abstraction, and levels of approximation. For instance, domains include function, geometry, and kinematic. A functional model represents the designer's intention, a geometric model represents the shape, and a kinematic model represents the motion of the design object. On

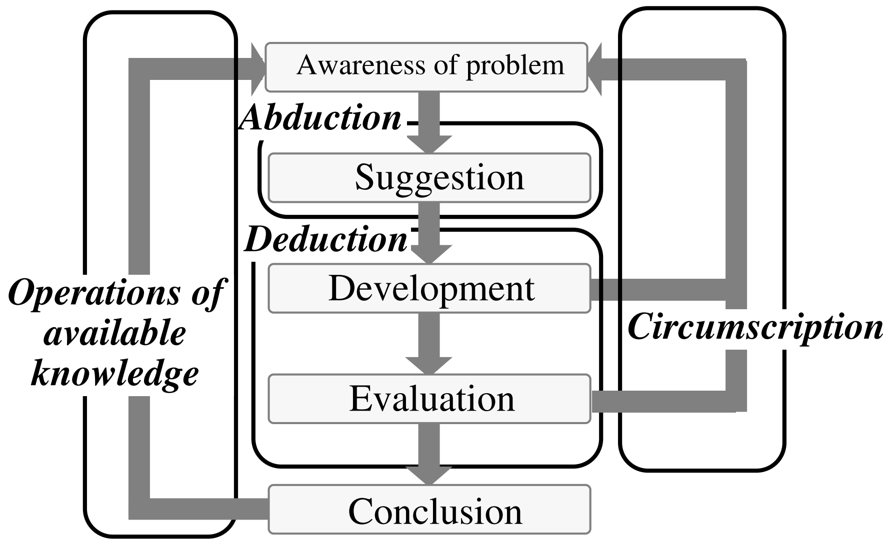

Figure 1: Logical inferences and the cognitive design process model (Takeda *et al.*, 1992)

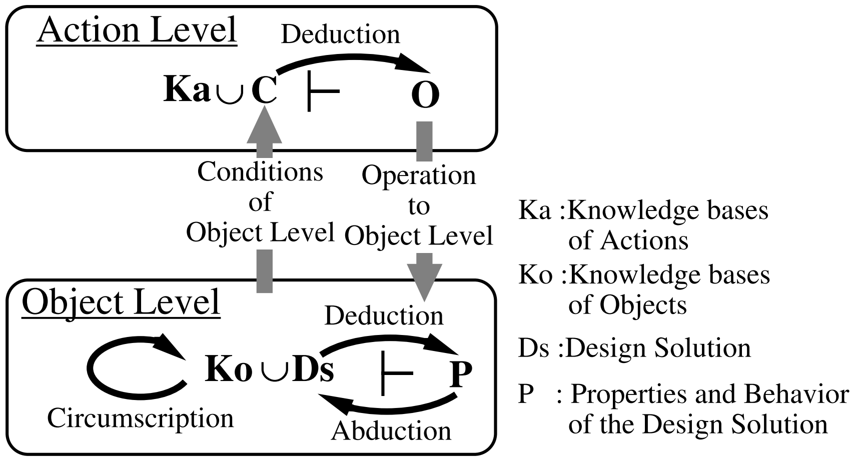

Figure 2: The computable design process model (Takeda *et al.*, 1992)

the contrary, a beam model and a finite element model are used for evaluating strength at different levels of abstraction. A kinematic model with friction and without are used for evaluating motion at different levels of approximation. These aspect models are not independent of each other; therefore, we need to have a mechanism to maintain consistency among aspect models.

The *metamodel mechanism* is a modeling framework for integrating multiple aspect models (Tomiyama *et al.*, 1989, Kiriyama *et al.*, 1991). It has symbolic representation of concepts about physical phenomena and mechanical components (Figure 3). A *metamodel* of a design object is represented as a network of relationships among concepts that appear in the aspect models. Types of relationship include causal dependency among physical phenomena, arrangements of components, and quantitative relationships.

An aspect model is generated from the metamodel by selecting only relevant concepts to the aspect (Kiriyama *et al.*, 1990). Since causal relationships among the generated aspect models are represented in the metamodel, the metamodel mechanism can maintain the consistency among the aspect models.

The metamodel mechanism uses *physical features* for describing a design object in the conceptual design. A physical feature is a piece of qualitative knowledge which represents me-

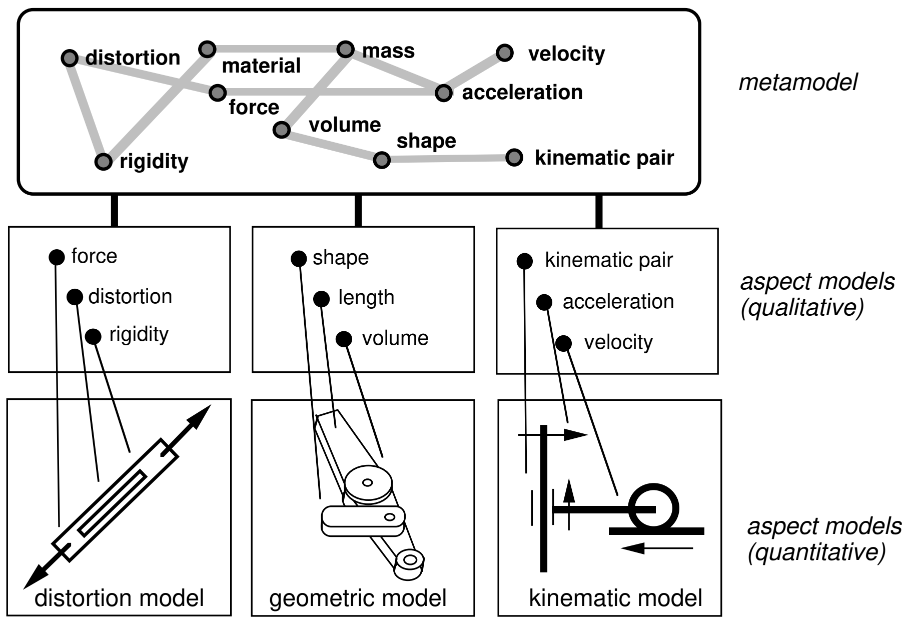

Figure 3: The metamodel mechanism (Kiriyama *et al.,* 1991)

chanical components and physical phenomena causally related to the components.

The designer builds an initial qualitative model of the principal physical behavior and the structure of the design object. We call this initial model a *primary model.* The primary model is an aspect model that represents the designer's mental model about behavior. A metamodel is generated from the primary model through qualitative reasoning.

2.3 Functional Representation

Modeling function of a design object is of particular importance to assist the designer in the conceptual design. A functional model is an aspect model that can represent the designer's intention.

We use the FBS (Function-Behavior-State) diagram (Umeda *et al.,* 1990, Umeda *et al.,* 1992) to represent functions. An FBS diagram denotes two types of relationship. One is relationship between behavior and state. A state is described by entities, attributes, and relations among them. Behavior is a sequence of one or more changes of states. This relationship between behavior and state is called *B-S relationship.* The other is relationship between function and behavior. A function is a subjective description of behavior abstracted through human recognition of the behavior in order to utilize it. This relationship between function and behavior is called *F-B relationship*, and gives physical semantics to a function.

An FBS diagram of a design object is composed of functional prototypes that represent the F-B relationship and physical features that represent behaviors and states of the design object. The scheme of a functional prototype is shown in Table 1 (Umeda *et al.,* 1992).

Table 1: Definition of a functional prototype

Item	Contents
Name	*verb + objectives*
Decomposition	networks of subfunctions
F-B Relationship	physical features

The designer uses the FBS diagram in two ways in functional design. One is function decomposition in which a function is divided into subfunctions. The other is function synthesis in which physical features that can exhibit the desired function are searched for and instantiated.

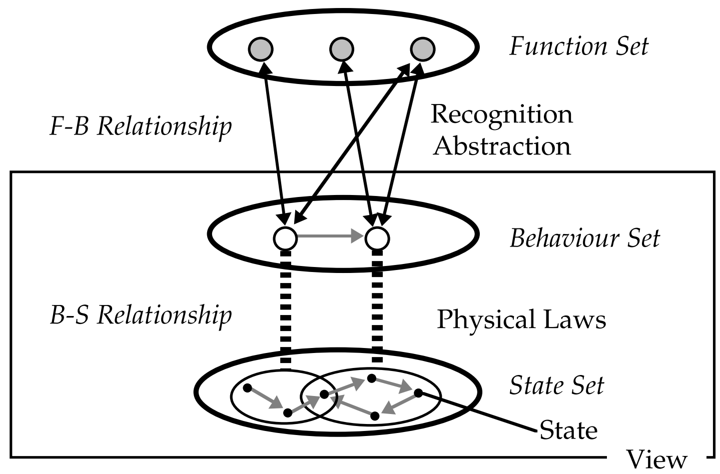

Figure 4: FBS diagram (Umeda *et al.,* 1990)

The metamodel mechanism maintains the consistency of the B-S relationships: Feasibility of functions is tested against the specification by reasoning out behavior of the design object.

3 Model Operations in the Computable Design Process Model

3.1 Extending the Computable Design Process Model

In the design process model described in Section 2.1, we assumed that there was only one aspect model through a whole design process. We extend the computable design process model by incorporating the metamodel mechanism and the FBS diagram as object level knowledge. The metamodel mechanism can consistently deal with various multiple aspect models and the FBS diagram can represent the functional aspect model.

Abduction in the extended computable design process model is realized as operations to the FBS diagram to find out a candidate that exhibits the most appropriate behavior from the given functional specifications. During functional design, the designer builds the functional structure of the design object with physical features from the functional specification. If the specification is too abstract to be associated with physical features, the designer recursively decomposes the specification into subfunctions. The functional structure is completed, when all the decomposed subfunctions are associated with physical features that realize as a whole the specification.

Deduction in the extended design process model performs simulation of behavior and is realized as model operations to the metamodel mechanism. This process involves two steps. One is a process corresponding to development of the design cycle in which a metamodel is generated from the primary model built during the functional design. The other corresponds to the evaluation of the design cycle to evaluate the design object using multiple aspect models. In this sense, model operations include those to build a primary model and those that deal with multiple aspect models.

3.2 Dealing with Aspect Models in Design Process Model

We should extend the action level inference of the computable design process model to include the following five metamodel operations for dealing with aspect models.

In the extended design process model, the designer modifies and evaluates the design object using multiple aspect models. Selection of an appropriate aspect model is equivalent to selection of available knowledge in the design process model. Selection of one aspect model implies changes of the designer's focus of attention in the design object.

We formalized this selection of an aspect model as follows.

1. *Select a domain.*
   Selection of a domain depends on the design context triggered by, for example, execution of a modification operation. For instance, the designer adds new descriptions of the design object using a functional model, and updates the primary model according to the progress of the design process. Therefore, the action level inference makes the metamodel mechanism to generate a metamodel from the modified primary model and all physical phenomena that possibly occur are recomputed. These physical phenomena and the changes of the primary model define a new design context according to which the action level inference selects an appropriate aspect model.

2. *Determine abstraction level for the selected domain.*
   The metamodel mechanism determines most appropriate abstraction level for the selected domain based on the knowledge about building aspect models. The metamodel mechanism suggests concepts included in the determined abstraction level and needed for the selected domain, and the designer has to give abstract descriptions of the design object using these concepts. For example, consider design of a robot arm. If the metamodel mechanism reasons out that distortion is possibly to happen, the metamodel mechanism suggests to describe the arm only using concepts such as beam, load, support, and bending. Then the designer abstracts the arm concept in the domain of geometry as a beam. This abstraction is computationally done by unification that is an operation to create a new instance that delegates the two concepts of shape "arm" and physical feature "beam."

3. *Determine approximation level for the selected domain.*
   The designer determines appropriate approximation level for the selected domain. He/She selects physical phenomena that should be considered in the selected aspect model. In other words, some physical phenomena might be neglected. For instance, the designer can say that any bending is considered while vibration is not of interest in the arm design.

4. *Generate aspect models.*
   The metamodel mechanism generates aspect models of the design object from the metamodel. Suppose in the example of arm design, a bending aspect model is generated. Since aspect models often require numerical information, the metamodel mechanism requests the designer to specify an appropriate aspect model to feed the required numerical information. For instance, since the generated bending aspect model needs dimensions of the beam, the designer specifies the solid modeler as a source to provide geometric information about the arm. Once this information is given, the metamodel mechanism can maintain consistently the relationship between these two models.

5. *Evaluate aspect models.*
   New information about the design object is obtained by evaluating the aspect model in the object level. Obtained information can be propagated to other aspect models for further evaluation.

3.3 Consistency among Aspect Models

Since the designer evaluates aspect models with various independent domain knowledge, he/she is sometimes confronted with inconsistency. For example, a motor seems to rotate qualitatively, though it does not quantitatively because of friction.

In the extended design process model, the metamodel mechanism maintains the consistency of aspect models. All of the aspect model have information about assumptions which are used in building itself. When a new aspect model is generated by the metamodel mechanism, the mechanism maintains the consistency between the new aspect model and other aspect models by checking assumptions of the new aspect model if they contradict with assumptions of other aspect models or not.

In case of the motor example, the metamodel mechanism can generate two types of aspect models; a model in which motor rotates, and otherwise.

4 The Prototype System

4.1 The System Architecture

Based on the extended computable design process model discussed in Sections 2 and 3, we have developed a prototype system that is capable of dealing with multiple aspect models. The system is implemented in Objectworks[1]\Smalltalk on a Sun workstation.

Figure 5 depicts the architecture of the system. Here, the FBS modeler (Umeda *et al.,* 1992) represents an FBS diagram, with which the designer decomposes functions and builds a primary model using the knowledge base of functional prototypes. As a result, the functional hierarchy information, behavior, and structure of the design object are obtained.

Information in the object level is stored in the FBS model and the metamodel, and further used for dealing with multiple aspect models. The current implementation of the prototype system is connected to three external modeling systems; i.e., DESIGNBASE[2] (a solid modeler), MODIFY[3] (an FEM system), and Mathematica[4] (a symbolic processing mathematics system). This system configuration proves that the metamodel system can realize the "pluggable" architecture that permits plugging-in existing external modeling systems.

The action level inference guides the designer to use the FBS modeler and the metamodel mechanism for abductive and deductive reasoning. The knowledge about the action level which does not depend on design objects is written in first order predicate logic. For example, the knowledge about development and

[1] Objectworks is a registered trademark of ParcPlace Systems.
[2] DESIGNBASE is a trademark of Ricoh Company Ltd.
[3] MODIFY is an object oriented finite element analysis system developed by Ohtsubo and Kubota(Ohtsubo *et al.,* 1992) at the University of Tokyo.
[4] Mathematica is a trademark of Wolfram Research Inc.

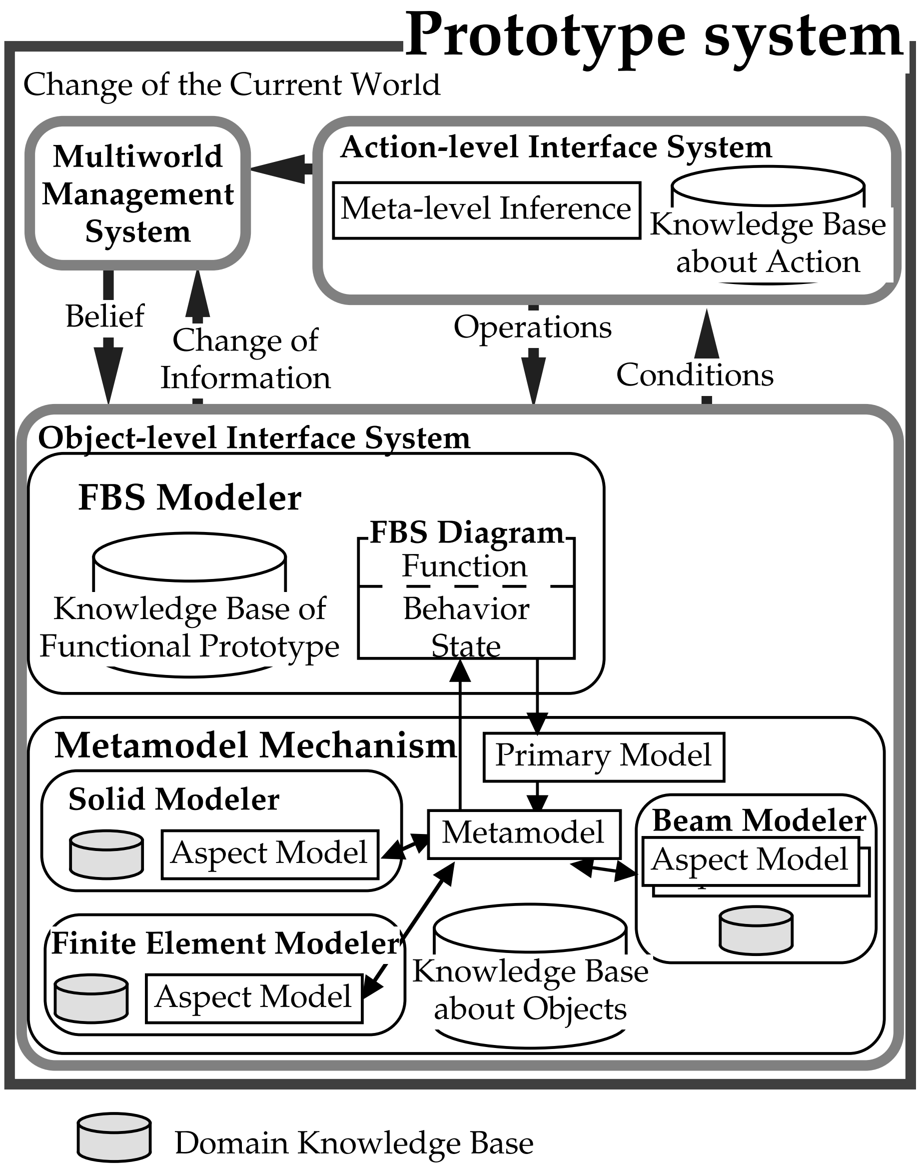

Figure 5: The architecture of the prototype system

evaluation is written below in an abstract but extremely general way.

```
if   (abducted)
          ; if abduction takes place
then (getNewAssumption NA)
          ; then identify modified place in the design solution
     (makeMetamodel NA NewDerived)
          ; and create a metamodel based on modification
     (selectAppropriateDomain NewDerived Domain)
          ; and select an appropriate domain
     (generateAspectModel Domain)
          ; and generate an aspect model.
```

In general, there are three types of knowledge in the action level knowledge base; one for requiring specification, one for suggestion, one for development and evaluation. The action level inference navigates model operations using these three types of knowledge.

4.2 Example

Let us describe an example of designing a hydrofoil boat on the system. Figure 6 illustrates a hydrofoil boat.

We demonstrate how the system performs one design cycle in the design of this boat (Figure 7). Before starting the design cycle, we assume that the designer has already built the functional hierarchy of the hydrofoil boat and decided to use hull structure.

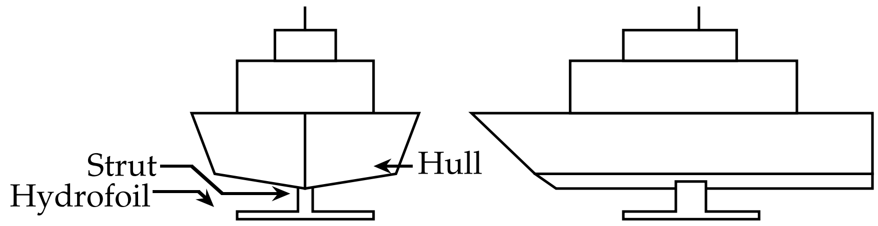

Figure 6: Hydrofoil boat

Figure 8 depicts the FBS diagram of the hydrofoil boat at the beginning of the design cycle. The oval nodes represent functions and the links between the oval nodes represent the functional hierarchy. The rectangular nodes denote physical features that compose the primary model for the metamodel mechanism.

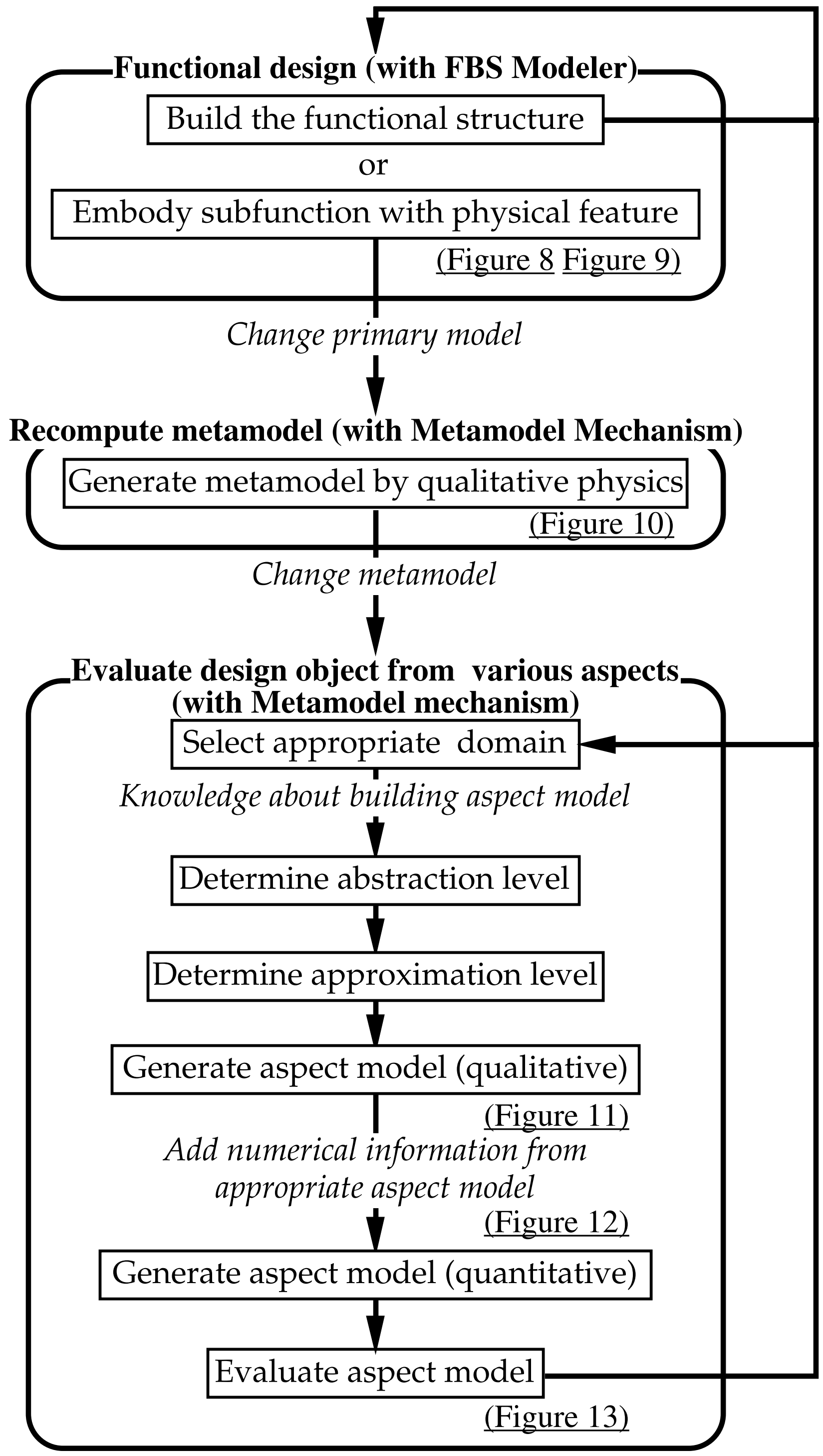

Figure 7: Design cycle of the prototype system

After the designer decided to focus on the function of float, the system suggests two candidates to achieve this function; one is a hydrofoil and the other is an air tank. The designer chooses

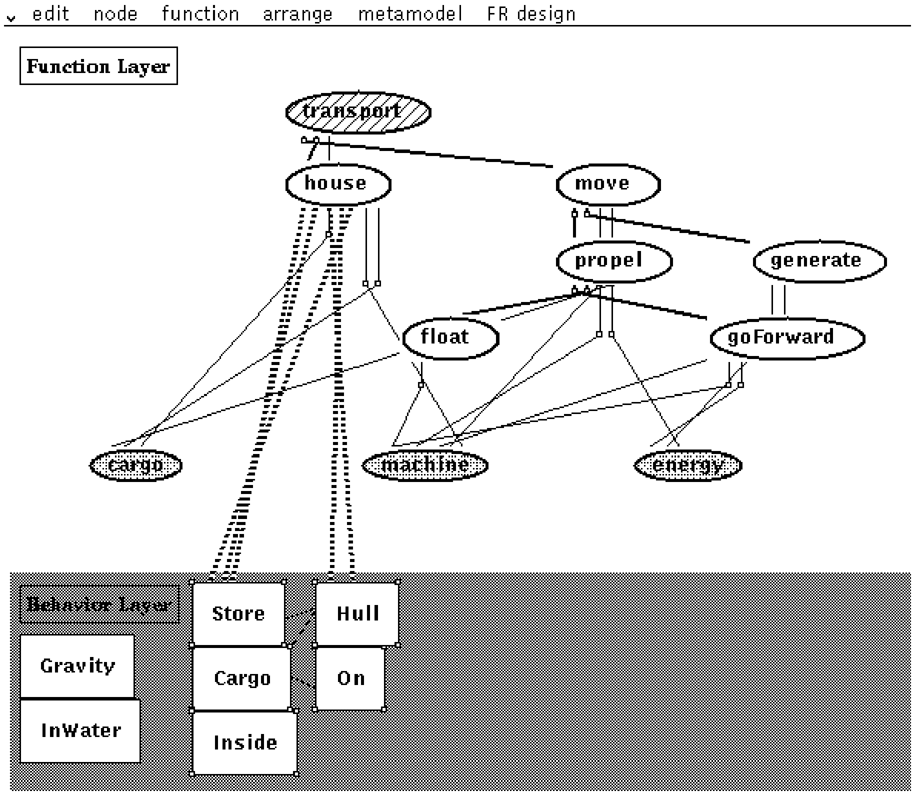

Figure 8: FBS diagram of the hydrofoil boat

the hydrofoil as the candidate. Then the system adds a hydrofoil, and a strut and relations among them to the FBS diagram (Figure 9) from which a metamodel is generated (Figure 10).

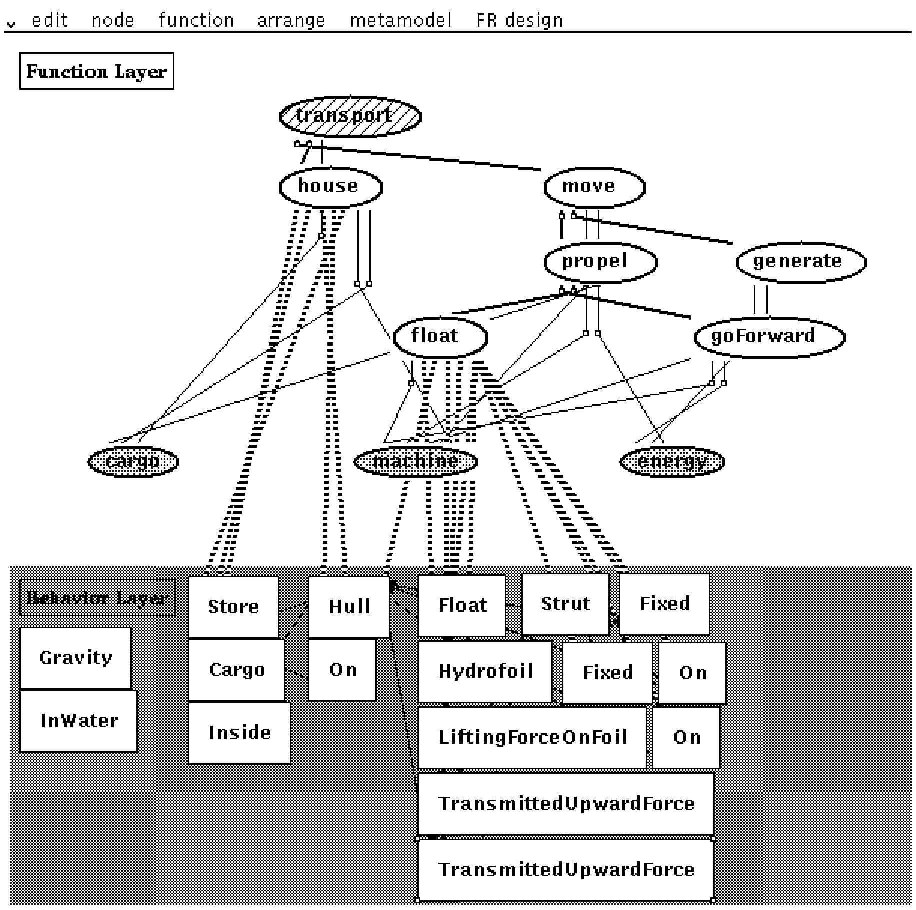

Figure 9: Revised FBS diagram of the hydrofoil boat

For selecting appropriate aspect models, the metamodel mechanism uses qualitative physics reasoning system (Kiriyama *et al.*, 1991) to reason about all possible physical phenomena that the hydrofoil and the strut may cause. In Figure 10, icons with dotted areas represent the derived physical phenomena. Since the lifting force occurred on the hydrofoil and the

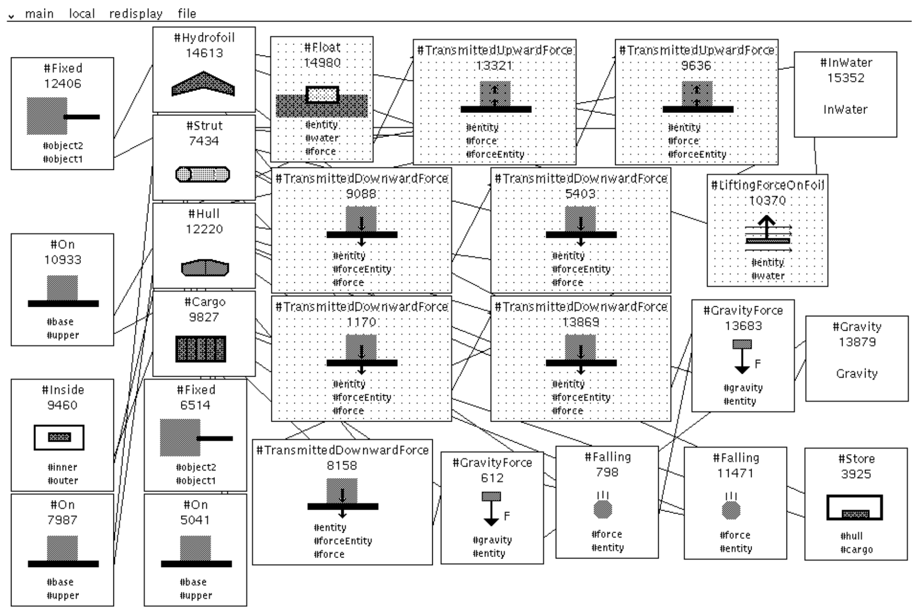

Figure 10: Metamodel of the hydrofoil boat

transmitted force occurred on the strut and the hull, the system selects the aspect of strength of material to analyze deformation of the hydrofoil, the strut, and the hull. Then a beam model and a finite element model can be generated. Note that, at this stage, these models are only qualitative and have no quantitative information.

The designer selects the beam model to analyze deformation of the hull. Then the metamodel mechanism prepares assumptions regarding abstraction of components for building the beam model; i.e., beam, load, support, and so on (Figure 11). Because the transmitted force on the hull equals to the reaction force of its weight, the designer can neglect the transmitted force. However, at this stage there is no numerical information about the structure, the metamodel mechanism cannot generate the beam model. Therefore, the designer provides geometric information by using DESIGNBASE, the solid modeler (Figure 12). Figure 13 depicts the shearing force diagram of the hull computed by Mathematica. If the designer wants to analyze the finite element model with MODIFY, the metamodel mechanism feeds the needed information to MODIFY from other aspect models.

5 Related Work

Xue et al. (1992) developed an earlier version of IIICAD (Intelligent Integrated Interactive CAD) to integrate knowledge about design processes and objects. The integration of IIICAD is, however, less mutual and less closer than the system proposed here, in that the design process manager cannot deal with operations to aspect models.

Nayak (Nayak, 1992) proposed a method to select an adequate model which can explain the given problem. His approach is suitable to generate explanations that arrive at an expected conclusion. However, this is not the case for design, in which no conclusion can be assumed before analyzing behavior of the design object.

Product model is an approach towards integration of various design object models. Most of product modeling systems places a geometric modeler in its core. Therefore, it is not easy to expand the framework if non-geometric information is needed.

PACT (Cutkosky *et al.*, 1993) is another framework for integrating existing external modeling systems. It is similar to our

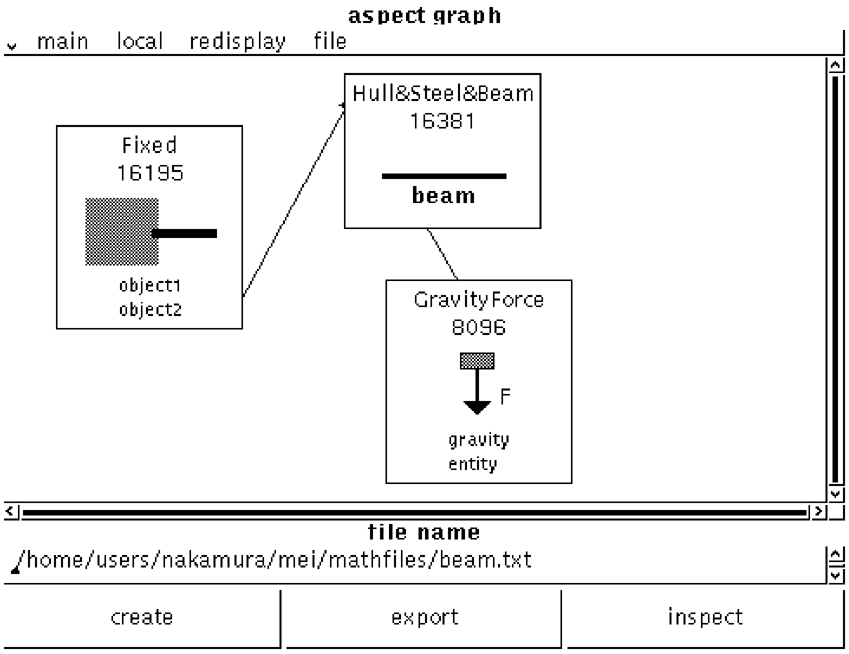

Figure 11: Generated beam model of the hull

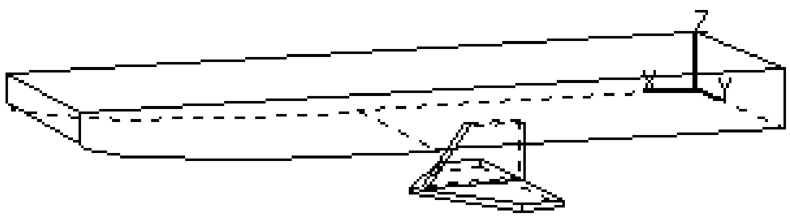

Figure 12: Geometric model of the hydrofoil boat

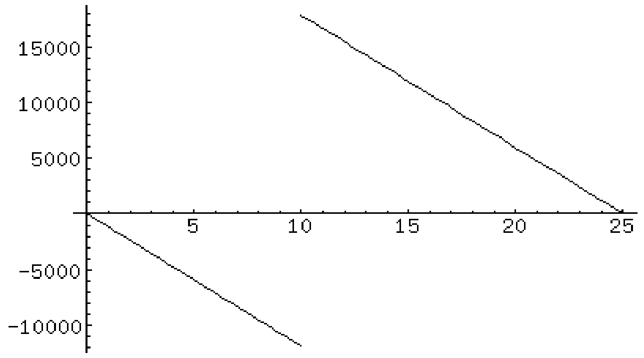

Figure 13: Result of shearing force analysis of the beam model

approach in that integrating models takes place through translating their knowledge to one format; i.e. they use KIF (Knowledge Interchange Format) (Genesereth, 1992) and Ontolingua (Gruber, 1992), we use qualitative process theory (Forbus, 1984) as the most fundamental description. However they do not mention about the design process management, although an intelligent agent oriented architecture is assumed.

6 Conclusions

This paper described how design process knowledge could be captured, formalized, and systematized as meta level operations of design object knowledge. We proposed a design process model that can deal with multiple design object models. The model is an extension of the computable design process model incorporating the FBS modeler and the metamodel mechanism. The object level knowledge is represented and maintained by the FBS diagram and the metamodel. The action level inference navigates a design process and controls operations of multiple design object models. Appropriate aspect models are selected by focusing on physical phenomena that might occur to the design object. The prototype's architecture also realizes the pluggable concept, which can contribute to advances of product modeling environment and CAD architecture.

Future work includes selecting an appropriate abstract level, and pointing out direction of improvement from the result of evaluation.

Acknowledgments

We would like to thank the staff and students of our research group at the University of Tokyo. Especially Dr. Takashi Kiriyama (Research into Artifacts, Center for Engineering) and Dr. Yasushi Umeda (Engineering Research Institute) for their support through discussions and implementations of the prototype system illustrated in Section 4. The example in Section 4 was provided by Professor Akihiro Kubota (Research into Artifacts, Center for Engineering).

References

Mark R. Cutkosky, Robert S. Engelmore, Richard E. Fikes, Michael R. Genesereth, Thomas R. Gruber, William S. Mark, Jay M. Tenenbaum, and Jay C. Weber. PACT: An experiment in integrating concurrent engineering systems. *IEEE COMPUTER*, Vol. 26, No. 1, pp. 28–37, 1993.

K. Forbus. Qualitative process theory. *Artificial Intelligence*, Vol. 24, No. 3, pp. 85–168, 1984.

M.R. Genesereth. Knowledge interchange format. In James Allen, Richard Fikes, and Erik Sandwall, editors, *Proceedings of the Conference of the Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers, 1992.

Thomas R. Gruber. Ontolingua: A mechanism to support portable ontlogies. Technical report KSL91-66, Knowledge Systems Laboratory, Stanford University, Stanford, 1992.

T. Kiriyama, T. Tomiyama, and H. Yoshikawa. Qualitative reasoning and conceptual design with physical features. In *Proceedings of the 4th International Workshop on Qualitative Physics*, pp. 153–160, 1990.

T. Kiriyama, T. Tomiyama, and H. Yoshikawa. The use of qualitative physics for integrated design object modeling. In *Design Theory and Methodology (DTM '91)*. The American Society of Mechanical Engineers(ASME), 1991.

T. Kiriyama, T. Tomiyama, and Yoshikawa.H. Building a physical feature database for integrated modeling in design. In *The Sixth International Workshop on Qualitative Reasoning about Physical Systems*, pp. 124–138, 1992.

D.B. Lenat and R.V. Guha. *Building Large Knowledged-Base Systems*. Addison-Wesley, Reading,MA, 1989.

J. McCarthy. Circumscription — a form of non-monotonic reasoning. *Artificial Intelligence*, Vol. 13, pp. 27–39, 1980.

P. Pandurang Nayak. Automated modeling of physical systems. Technical report STAN-CS-92-1443, Department of Computer Science, Stanford University, Stanford, 1992.

Hideomi Ohtsubo, Akihiro Kubota, and Yasumi Kawamura. MODIFY: Object-oriented finite element modeling system. In *International Conference on Computational Engineering Science*, HongKnog, 1992.

C.S. Peirce. Elements of logic. In C. Hartshorne and P. Weiss, editors, *Collected Papers of Charles Sanders Peirce, Volume II*. Harvard University Press, Cambridge,MA, 1932.

H. Takeda, S. Hamada, T. Tomiyama, and H. Yoshikawa. A cognitive approach of the analysis of design processes. In *The Second ASME Design Theory and Methodology Conference*, pp. 153–160. The American Society of Mechanical Engineers(ASME), 1990.

H. Takeda, T. Tomiyama, and H. Yoshikawa. A logical and computable framework for reasoning in design. In *Design Theory and Methodology (DTM '92)*. The American Society of Mechanical Engineers(ASME), 1992.

T. Tomiyama, T. Kiriyama, H. Takeda, and D. Xue. Metamodel: A key to intelligent CAD systems. *Research in Engineering Design*, Vol. 1, No. 1, pp. 19–34, 1989.

T. Tomiyama. Intelligent CAD systems. In G. Garcia and I. Herman, editors, *Advances in Computer Graphics VI, Images: Synthesis, Analysis, and Interaction*. Springer-Verlag, Berlin, 1991.

T. Tomiyama, D. Xue, Y. Umeda, H. Takeda, T. Kiriyama, and H. Yoshikawa. Systematizing design knowledge for intelligent CAD systems. In G.J. Olling and F. Kimura, editors, *Human Aspects in Computer Integrated Manufacturing, IFIP Transactions B-3*. North-Holland, Amsterdam, 1992.

D.G. Ullman, T.G. Dietterich, and L.A. Stauffer. A model of the mechanical design process based on empirical data: A summary. In J.S. Gero, editor, *Artificial Intelligence in Engineering Design*, pp. 193–215. Elsevier, Amsterdam, 1988.

Y. Umeda, H. Takeda, T. Tomiyama, and H. Yoshikawa. Function, behavior, and structure. In *AIENG '90 Applications of AI in Engineering*. Computational Mechanics Publications and Springer-Verlag, 1990.

Y. Umeda, T. Tomiyama, and H. Yoshikawa. A design methodology for a self-maintenance machine based on functional redundancy. In *Design Theory and Methodology (DTM '92)*. The American Society of Mechanical Engineers(ASME), 1992.

B. Veth. An integrated data description language for coding design knowledge. In P.J.W. ten Hagen and T. Tomiyama, editors, *Intelligent CAD Systems I – Theoretical and Methodological Aspects*, pp. 295–313, Berlin, 1987. Springer-Verlag.

D. Xue, H. Takeda, T. Kiriyama, T. Tomiyama, and H. Yoshikawa. An intelligent integrated interactive CAD-a preliminary report. In D. Brown, M.B. Waldron, and H. Yoshikawa, editors, *Intelligent Computer Aided Design, IFIP Transactions B-4*. North-Holland, Amsterdam, 1992.

H. Yoshikawa. General design theory and a CAD system. In T. Sata and E.A. Warman, editors, *Man-Machine Communication in CAD/CAM, Proceedings of the IFIP Working Group 5.2 Working Conference 1980 (Tokyo)*, pp. 35–58. North-Holland, Amsterdam, 1981.