ABDUCTION FOR DESIGN

Hideaki Takeda

This paper describes a logical design process model and in particular abduction in it. We present our model of design processes that consists of abduction, deduction, circumscription, meta-level inference, and multi-world mechanism. In this model, design is a process of refining descriptions of design objects. Abduction generates object descriptions as a hypothesis, deduction examines validity of the object descriptions proposed by abduction, circumscription maintains knowledge used in abduction and deduction by resolving inconsistency, meta-level inference provides knowledge for abduction, and multi-world mechanism represents evolution of the object descriptions. Since abduction is crucial part to realize synthesis in design, we clarify the role of abduction in design and develop an abductive inference for design. Here we define abduction as a process making integrated hypotheses and theories to explain the given facts. It means that (i) not only hypotheses but also theory to explain facts with hypotheses are important part of explanation, and that (ii) integration should be kept both in hypotheses and in theories. To achieve above features of abduction in a knowledge-based framework, we introduce a *segment of explanatory coherence* and *superposition of proposition.* The former shows connectivity of explanations and the latter realizes integration of hypotheses. Since abduction contributes to integration of knowledge, the framework based on abduction gives an answer to deal with variety of knowledge in CAD systems.

1. Introduction

Recently CAD systems are largely used in industries, and there is no doubt that CAD systems contribute to increasing design quality and decreasing designers' routine work like drafting. Although current CAD systems, in which geometric modeling is centered, have been developed rapidly and used widely, the next generation CAD systems, which can support designers from the beginning of design, remain obscure in their concept and realization. This is mainly because of lack of formal theory for design that can explicate design and also serve as foundation to realize design activities in computers.

The purpose of this paper is to propose a model of design that is not only descriptive but also computable (Finger and Dixon, 1989). In order to establish such a model, we adopt logic as the basic framework, and model design processes as logical reasoning. We have proposed the framework to model design processes in which design is modeled as abduction, deduction, circumscription, and meta-level inference (Takeda, Tomiyama and Yoshikawa, 1992; Takeda, Veerkamp, Tomiyama and Yoshikawa, 1990; Takeda, Tomiyama, Yoshikawa and Veerkamp, 1990). In this paper, we focus on abduction in this model. Abduction is crucial part to realize synthesis in design. We clarify the role of abduction in design and develop an abductive inference for design.

In the next section, we will discuss our design process model which includes various types of inferences in the logical framework. After we will discuss characters of abduction in Section 3, then we will show how such characters can be realized as a knowledge-based inference in Section 4. We will demonstrate how this inference works in design by using the design simulator that we implemented to evaluate the design process model in Section 5. We will discuss future directions for this research in Section 6, then summarize the paper in Section 7.

2. Logical Design Process Modeling

2.1. THE LOGICAL FRAMEWORK FOR DESIGN

In order to describe design processes in the logical framework, we should clarify what we should represent in logic. Although many factors are complexly related to design, we use three factors which are prerequisite to describe design processes. These factors are required specifications, design solutions (design objects), and knowledge. And we interpret design as logical inference among them.

It may seem natural to take the *deductive framework* to describe design processes in logic. In this approach, we can formalize design as follows;

$$S \cup K \vdash Ds$$

where S, K, and Ds are sets of formulae that denote required specifications, knowledge used in design, and design solutions, respectively. Here solutions are derived from specifications and knowledge as the results of deduction. In short, this approach adopts the "design is deduction" paradigm.

Many works which explain design or design processes in logic are based on this framework in principle. For example, Treur (1991), and Dietterich and Ullman (1987) took this approach, and we also took it in Ref. (Takeda, Tomiyama and Yoshikawa, 1990).

This "design as deduction" approach may be suitable for routine design, but it cannot offer a sufficient framework for other more flexible and complicated design. Although solutions and knowledge are always incomplete in design, it requires solid and absolute knowledge and solutions.

Then we can use the second framework — the *abductive framework*. In this case, specifications can be derived from design solutions and knowledge.

$$Ds \cup K \vdash S.$$

Here design is abduction with knowledge and specifications.

Coyne (1988) and RESIDUE system (Finger and Genesereth, 1985) stand for this approach for design formalization.

Knowledge represented in this framework is knowledge about objects themselves, i.e., knowledge about object properties and behaviors, because formulae in this framework should be prepared to deduce properties and

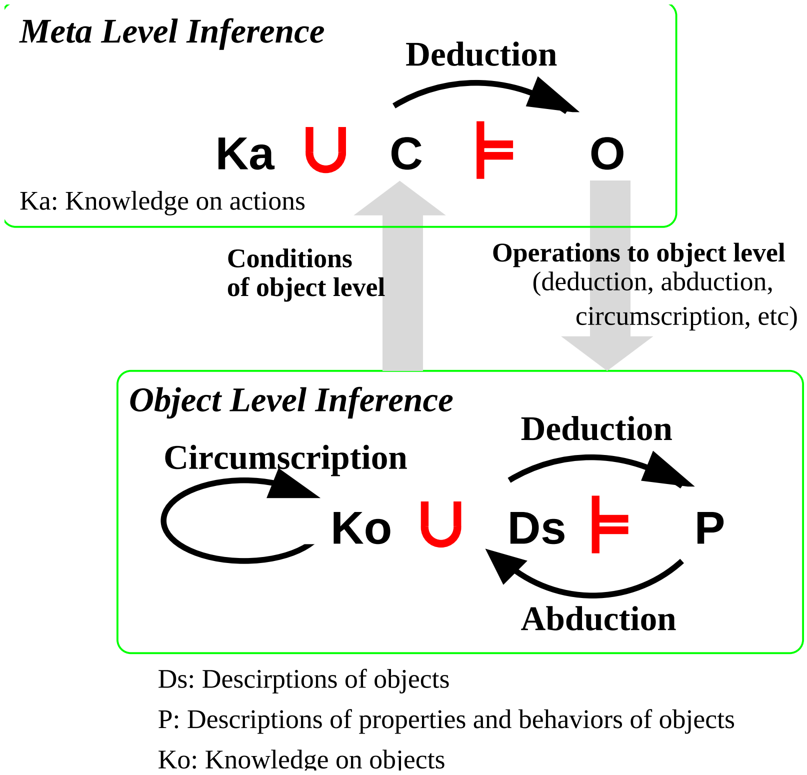


Fig. 1. The logical design process model

behaviors of objects from descriptions of objects themselves. It is more desirable than knowledge representation in the deductive framework where knowledge is about how to design.

Furthermore solutions the abductive inference can generate are, by definition, not definite solutions but feasible solutions.

Therefore, we adopt the abductive framework as the framework of the logical formalization of design.

2.2. The Logical Inference Model for Design Processes

The inference model we propose is illustrated in Figure 1. We define the design process model as a logical inference model.

Here there are two levels in the model, one is the object level and the other is the action level. The object level contains descriptions of design objects (design solution) Ds, knowledge about objects Ko, and descriptions of object properties and behaviors P. P can include required specifications.

The basic design process is interpreted by iteration of abduction and deduction that evolve design objects and their properties and behaviors, and circumscription is invoked to resolve inconsistency.

The action level contains knowledge about actions (knowledge about how to design) Ka, and the meta-level inference is performed to proceed design by specifying inferences in the object level and operating directly the contents of Ds, Ko, and P.

Changing of design objects (Ds) is managed by the multi-world mechanism based on a type of modal logic. Every state of design objects in design processes corresponds to a possible world in modal logic so as to manage multiple solutions and operations to design processes themselves.

2.2.1. Iteration of Abduction and Deduction as the Basic Process

We regard a design process as an evolutionary process, that is, the design objects are refined in step-wise manner. We call each state of step-wise refinement as a *design state.* In each state, the following three types of descriptions hold; The first one is descriptions of the current design solution which is denoted by Ds. It consists of identifiers of design objects which are components of the current design solution, and properties and relations which are *necessary* to identify the objects. The second one is descriptions of properties and behaviors of the current design solution, P. It consists of all kinds of properties and behaviors that the current design solution has. Required specifications are included in P. The third one is knowledge that is available at the current state, Ko. These descriptions are kept consistent to satisfy the following formula;

$$Ds \cup Ko \vdash P.$$

Given design knowledge Ko and the required properties P as the specifications, the designer tries to find a candidate by abduction, hence, the current descriptions of the design objects are formed. Then deduction is performed to obtain all the properties of the current solution with respect to the current available knowledge. It is performed (i) to see what properties the solution has and (ii) to see whether the solution does not contradict with the given specifications and knowledge. Then again abduction is performed to evolve the solution more — new descriptions for the next state are formed. If the solution does not satisfy the specifications or can not evolve any more, the designer either tries an alternative solution or modifies the design knowledge and the specifications.

This iteration of abduction and deduction continues until the descriptions of the objects become fully detailed ones that are suitable to hand the next process (e.g., manufacturing) or reach a situation where no more evolutions are possible.

2.2.2. Circumscription for Resolution of Inconsistency

Inconsistency in design has not only negative effects in design but also positive ones. Most cases of inconsistency in design does not mean that knowledge has wrong information essentially, but that knowledge is used in a wrong manner. Knowledge is used beyond situations where it is expected to be used. But it is not impossible to describe all applicable situations in advance, because it is the nature of knowledge in design that boundary of applicability is vague.

Here we assume that inconsistency comes from such incompleteness of the knowledge description. Then resolution of inconsistency is to find implicit descriptions of knowledge which restrict applicability of knowledge. One solution to accomplish this process is *circumscription* (Lifschitz, 1985; McCarthy, 1980).

Circumscription is a type of commonsense reasoning and has been developed to deal with *exceptions.* In circumscription, exceptions for given contexts can be determined by minimizing logical extensions of the predicates which represent *abnormality* with keeping the whole context consistent. Here abnormality is the implicit description of each piece of knowledge.

When abduction is performed with the modified knowledge by circumscription, we can obtain different results from before. Thus use of circumscription not only solves inconsistency but also helps design to proceed more by modifying knowledge.

2.2.3. Meta-level Inference for Actions

We defined the basic design process as iteration of abduction and deduction on descriptions of objects, knowledge about objects, and descriptions of object properties and behaviors. We also introduced circumscription to resolve inconsistency.

Although they explain what the designer can do with given knowledge and specifications, they can not deal with change of knowledge or specifications because such actions require change of axioms or theorems of the logical system and therefore it is out of a logic system.

In order to solve this problem, we introduce a meta-level inference architecture. Metal-level inference architectures for reasoning are suggested by many researchers. For example, Weyhrauch (1980) proposed FOL which is a meta-level reasoning system based on first-order logic. Usually the relation between axioms and theorems in the object-level logical system corresponds to the atomic formula in the meta-level logical system. In our approach, what the meta-level system treats as its atomic formula is the relation among descriptions of objects, available knowledge about objects, and descriptions of object properties and behaviors in the object-level system. We can represent this as follows;

$$Ds \cup Ko \vdash_{L_O} P \Leftrightarrow \vdash_{L_M} design(Ds, Ko, P).$$

where $\vdash_{L_O}$ and $\vdash_{L_M}$ denote derivativeness in the object-level system and in the meta-level system respectively.

The current condition of the three elements in the object level systems is constantly reported to the meta-level system, and the results of inference in the meta-level system are reflected to the object-level system. The reflection is the change of the condition of the object level system, i.e., either specification of the next inference or modification of the contents of the three elements in the object-level system.

Knowledge about how to design can be described as formula in the meta-level system. For example, a rule "if you are designing a certain object g, you should use knowledge base K_g" can be described as follows;

$$design(Ds, K, P) \wedge g \in Ds \rightarrow design'(Ds, K \cup K_g, P)$$

We can thus describe knowledge like design rules and design procedures in this level.

2.2.4. Multi-worlds for Representation of Changing

Each element of the object level system (descriptions of objects, available knowledge about objects, and descriptions of object properties and behaviors) is changed dynamically by either the object-level inference or the meta-level inference. We introduce the multi-world mechanism based on modal logic to manage this changing.

Since a designer accumulates her or his decisions as the design solution in step-wise refinement processes, it is crucial to distinguish what is already determined from what is not determined yet. It is suitable to represent such situations by partial semantics. Therefore we can use *data logic* to represent them. Data logic (Landman, 1986; Veltman, 1981) is intuitively a version of modal logic based on partial semantics. There are three truth values, i.e., t, f, and u. The third value can be interpreted as *undecided* . Among these values, a partial-order $\sqsupseteq$ is defined where $t \sqsupseteq u$ and $f \sqsupseteq u$ are hold. In this logic, we can access the other possible worlds from a certain possible world, if the value of every proportion in the world is not lower than that in the original world with respect to partial-order $\sqsupseteq$. This means that the *next* world is more determined one than the current world. The truth value u is thus expected to fall into either t or f at last.

Since changing of descriptions of object properties and behaviors (logically it means a set of derivable formulae) is monotonic, we can use possible worlds and accessibility to represent design states and their relations.

In this formalization, if a designer obtains two new different object descriptions from a single solution, two possible worlds are created as descendants of the current possible world. Revision and retraction of the design solution means backtracking to the desirable world (the latest world which does not contradict with the new object descriptions) and creating a new world as its descendant.

3. Characters of Abduction in Design

We have presented our model of design processes that consists of abduction, deduction, circumscription, meta-level inference, and multi-world mechanism. Abduction is crucial part of this model, because it should represent synthesis in design. Abduction generates object descriptions as a hypothesis, while other types of reasoning assist this process. Deduction examines validity of the object descriptions proposed by abduction, circumscription maintains knowledge used in abduction and deduction by resolving inconsistency, meta-level inference provides knowledge for abduction, and multi-world mechanism represents evolution of the object descriptions.

Although we have shown the function of abduction in design, we have not discussed mechanism how abduction should be performed. We discuss nature of abduction in design in this section, and then discuss the mechanism to involve such nature of abduction as an inference in the next section.

3.1. ABDUCTION IN COMPUTER SCIENCE

C.S. Peirce introduced abduction as the third kind of reasoning in logic in addition to deduction and induction.

One of important characters of abduction he argued is that direction of inference in abduction is opposite to that in deduction. For example, he demonstrated abduction as follows (Peirce, 1935);

> The surprising fact C is observed,
> But if A were true, C would be a matter of course;
> Hence, there is reason to suspect that A is true.

Many logical formalizations for abductive reasoning have been proposed recently, including Levesque (1989), Poole (1988), Cox and Pietrzykowski (1986), and Finger and Genesereth (1985), but their definitions for abduction are basically similar, i.e., abduction for an observation O with a theory T is to find a hypothesis A which consists of (ground instances of) possible hypotheses and satisfies both $A \cup T \vdash O$ and $A \cup T$ is consistent. This definition is logically sound and suitable to represent the character of abduction mentioned above.

Unfortunately, this definition of abduction fails to capture another important character of abduction. Abduction is *ampliative* reasoning, while deduction is merely *explicative* reasoning. In ampliative inference the conclusion introduces new ideas into our store of knowledge, but it it does not follow from the premises with necessity (Fann, 1970). In explicative inference the conclusion explicates what is stated in the premises and follows form the premises necessarily.

Hypotheses generated by the above definition are *definitely* all what can deduce the given observation with the given theory, and ampliativity is realized just by enumeration of multiple hypotheses.

This *clear and definite* abduction is unattractive in design because of complexity and quantity of object structures and knowledge. Since it translates ampliative ability of abduction into enumeration of multiple hypotheses, it would generate an enormous number of hypotheses. We need the other way to interpret ampliative ability of abduction.

The problem lies in the following two issues. One issue is that they put abduction into a traditional problem solving scheme. It should include not only problem solving but also problem formation to some extent. Although abduction may generate hypotheses by using reasoning like *reversed deduction*, it does not imply that the whole process of abduction is such reasoning. The other issue is lack of structures in hypotheses and the background theory. They assume simple and uniform structures that hide crucial problems in abduction like composition of hypotheses.

In the following discussion, a problem given to abduction to solve is called an *observation.* It represents facts in the target world and it is what we should find explanation for. Knowledge which is used to find explanation is call a *background theory.* A *hypothesis* is an idea conjectured by abduction.

Then we will discuss what kind of characters are needed for abduction in design.

3.2. EXPLANATION = HYPOTHESIS + THEORY

When a set of facts is given as an observation, abduction is to make hypotheses that explain the given facts with a background theory. It is important to show not only proposed hypotheses but also how the background theory is used to explain the facts. One of the requirements that the proposed hypotheses should satisfy is to show that they can deduce the given facts. In this deduction, at least the background theory that is used in abduction should be included in the axiom.

But there are no reasons that the background theory used in abduction is identical to the whole background theory that exists before abduction. It is natural to assume that the background theory used in abduction can be extracted from the whole background theory. We call this used background theory *an explanatory theory.* Then abduction is to make a hypothesis and an explanatory theory of which combination can explain the given facts.

The content of explanatory theories need not be created newly. To create new rules or laws is another abduction problem in a higher level. An explanatory theory consists of formulae that are selected from the existing background theory.

3.3. Integration of Explanations

As we mentioned, the first requirement for hypotheses is to derive the observation with the background theory. But it is not sufficient to restrict generation and selection of hypotheses, and some criteria are proposed as methods, for example, specificity (Pople, 1973) and creditability (Hobbs, Stickel, Martin and Edwards, 1988).

In this paper, we use *integration of explanations* for criteria for generating and selecting explanations, It means how parts of explanation are integrated together as an explanation, and this criteria is thus to ensure that an explanation is valid to explain observation as a whole. Integration of explanations has two meanings; One is coherence of explanations that is integration of the way how they explain the observation. It represents plausibility of hypotheses as explanation. We will discuss this problem in Section 4.3. The other is integration of hypotheses that is how parts of a hypothesis are integrated together. It represents plausibility of hypotheses themselves. It is related to the problem how abduction can yield new ideas. We clarify it in the next subsection.

3.4. Creation of New Ideas

Creation of new ideas in abduction can be realized as new combinations of propositions in hypotheses. Even if every proposition in a hypothesis is well-known, combination of these propositions can express a new idea if it is a new and meaningful combination. Therefore, it is crucial for abduction to provide meaningful combinations in a hypothesis.

In order to obtain meaningful combinations, members of a hypothesis should be related to each other. A set of irrelevant propositions does not carry any new ideas as a hypothesis. Integration of hypotheses to yield new ideas is therefore to conglomerate propositions which are able to carry new ideas.

In design, an important relationship among propositions is sharing of entities. Combination of propositions in a hypothesis is more important in design if they share entities, because it forms a description of an object.

Integration of hypotheses in design is, so far, to form conglomerate of information around entities. More propositions share entities in a hypothesis, more integrated it is to be able to carry new ideas.

This problems is significant when an explanation needs new entities. Since each proposition in a hypothesis is proposed as explanation of some propositions in the observation, it can have its own new entities if the explanation requires entities which are different from entities in the observation.

If different propositions in the hypothesis can share such newly required entities, we can obtain a more integrated hypothesis. It means to identify entities which are introduced independently and therefore have different properties. We call identification of different entities *superposition*. It is a fusion of different concepts and such fusion can produce a new concept. In abduction this process is performed in order to obtain integrated hypotheses. We will discuss how to realize it in Section 4.2.

3.5. ABDUCTION AS PROBLEM FORMATION

Abduction can work not only to solve problems but also to generate information to define problems.

Since design consists of a lot of ill-defined problems, it is difficult in design to define space for solution and domain of knowledge (heuristics) for each problem in advance. In such domains, abduction can serve to define space for solution and extent of knowledge to solve the problem preliminarily.

As we have shown in the design process model, abduction in design is used iteratively and develops hypotheses gradually. Each abduction stops its inference by using some criteria, and generates hypotheses and explanatory theories gathered from the background theory. Since each hypothesis is not so definite, it does not indicate that the hypothesis is a solution exactly, but that there would exist a solution around the hypothesis. Thus we can use it as preliminary definition of space for solution. An explanatory theory also indicates preliminary definition of knowledge. Because the explanatory theory is collected just to explain the observation, it is not all information we can solve the problem. But at least it is related and maybe needed to solve the problem. It serves, therefore, the first definition of extent of knowledge.

After we obtain preliminary definitions of space for solution and extent of knowledge, we can use more definite methods like deduction.

Furthermore characters of abduction we have explained can generate other information to define problems. We will discuss how these types of information can be used in a knowledge-based inference in Section 4.4.

4. Formalization of Abduction as a Knowledge-based Inference

We have discussed general characters of abduction in design in the previous section. In this section we focus on how to realize abduction in the current knowledge-based framework.

Here we provide a first-order language $\mathcal{L}$, and explanatory hypotheses, observations, and background theory are written in the first-order predicate language. We can define abduction as follows;

DEFINITION 1. *Explanation.*
An explanation of an observation O with a background theory K_0 is $\langle A, K\rangle$, a tuple of an explanatory hypothesis A and an explanatory theory K which satisfy the following conditions;

- $K \subseteq K_0$,
- $K \cup A$ *is consistent,*
- $K \not\models O$,
- $A \cup K \models O$, *and*
- *there are no* $E \subset A \cup K$ *that stratifies* $E \models O$.

We can say that a hypothesis A explains an observation O by an explanatory theory K. In this paper, we restrict both observations and hypotheses to ground formulae, i.e., no variables are appeared in them. Furthermore observations are given as a set of literals (atomic formulae or negation of atomic formulae).

This definition may seem identical to the definition in Section 3.1, but an explanation is not a hypothesis but combination of a hypothesis and an explanatory theory, and the whole background theory is not required to use in abduction. But we have not defined how an explanatory theory is taken from the background theory, and what an explanatory hypothesis consists of.

Under this definition, we then define structures of a background theory and hypotheses to realize integration of explanation.

4.1. STRUCTURE IN THEORY

When designers design even a single object, they usually use various kinds of knowledge which come from some different backgrounds. They can manage to combine and use such kinds of knowledge which sometimes seem to be inconsistent to each other. It is important for design to deal with such variety of knowledge.

Instead of assuming a single background theory, we here assume a set of background theories, i.e., the background theory is divided into separate *aspect theories* each of which has its own perspective of description. The perspective of an aspect theory is how to represent phenomena or concepts as propositions in laws or rules. It is definition of vocabulary for the aspect.

Furthermore we assume clusters of knowledge in an aspect theory. We often use some part of an aspect theory instead of the whole aspect theory. A cluster of knowledge is a unit to handle aspect theories. An aspect theory consists of a set of clusters.

We can define aspect theories as follows;

DEFINITION 2. *Aspect.*
An aspect A_i *consists of vocabulary* V_i, *an aspect theory* K_i, *and clustering of knowledge* $KC_i = \{kc_i^j\}$. *The aspect theory is a set of formulae written with the vocabulary, and divided into knowledge clusters, i.e.,* $KC_i = \bigcup_{j \in \Lambda_C} kc_i^j$ *where* Λ *is a set of identifiers for knowledge clusters.*

We need knowledge to connect different aspect theories in order to use them together. We call it an *inter-aspect* theory. Since different aspect theories may represent the same phenomena or concepts differently, the inter-aspect theory holds relations among such representations. Then we can define the background theory.

DEFINITION 3. *Background theory.*
The background theory K_0 *is union of aspect theories* K_i *and the inter-aspect theory* K_I, *i.e.,*

$$K_U = \bigcup_{i \in \Lambda} K_i \cup K_I,$$

where Λ *is a set of aspect identifiers.*

Then an explanatory theory can be defined as collection of clusters of knowledge to explain the given observation (see Figure 2).

DEFINITION 4. *Explanatory theory.*
An explanatory theory K *for the background theory* K_0 *is union of knowledge clusters taken from aspect theories in the background theory.*

Of course this definition satisfies $K \subseteq K_0$.

4.2. SUPERPOSITION IN HYPOTHESES

Then we can also divide an explanatory hypothesis as follows;

$$A = A_{TH} \cup A_I$$

Here, A_{TH} is the derivative hypothesis that can be derived from the background theory and the observation. A_{TH} is a set of formulae with vocabulary $V_0 = \bigcup_{i \in \Lambda} V_i$. A_I is the connective hypothesis that integrates members of the derivative hypothesis (see Figure 2). A derivative hypothesis A_{TH} alone can satisfy derivativeness of the observation O, i.e.,

$$A_{TH} \cup K_0 \models O.$$

Since the hypothesis is generated from combination of different aspect theories, it may be merely a set of hypotheses each of which is generated from an aspect theory. To ensure integration of the hypothesis, we need the connective hypothesis which combines parts of the derivative hypothesis together. We realize this connective hypothesis as superposition of entities.

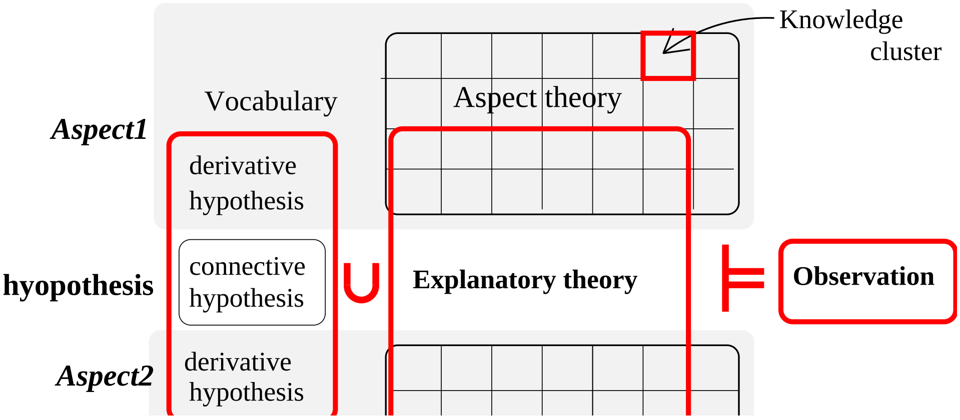


Fig. 2. Hypothesis and explanatory theory

4.2.1. Instantiation of Entities

An observation is a description about entities, and a hypothesis is another description about entities appeared in the observation. But in synthesis one should consider not only entities presented in the observation, but also other entities needed to solve the given problem. We can call these entities instantiated entities.

Introduction of new entities should be careful because it changes the degree of integration of explanation. It is one of important criteria to create and evaluate hypotheses.

Suppose $\boldsymbol{a} = \langle a_1, ..., a_i \rangle$ a tuple of constants appeared in the observation, $\boldsymbol{i} = \langle i_1, ..., i_j \rangle$ a tuple of instantiated constants appeared in the hypothesis A, and $\boldsymbol{x} = \langle x_1, ..., x_i \rangle$ a tuple of variables.

Suppose that there are no constants in the explanatory theory.

We can get $A(\boldsymbol{x})$ by substituting each constant in A, $A(\boldsymbol{x})$ itself can explain the observation too, i.e.,

$$\forall \boldsymbol{x} A(\boldsymbol{x}) \cup K \models O.$$

Since we need hypotheses of ground formulae, we find a substitution θ to all variables in $A(\boldsymbol{x})$ so that $A(\boldsymbol{x})\theta = A$ (Lloyd, 1984).

We can also represent O as $O(\boldsymbol{y})\gamma_a$ where $\gamma_a = \{a_1/y_1, ..., a_i/y_i\}$ is a substitution. Then,

$$\forall \boldsymbol{x} A(\boldsymbol{x}) \cup K \models \forall \boldsymbol{y} O(\boldsymbol{y}).$$

Since the observation is given as $O(\boldsymbol{y})\gamma_a$ not as $O(\boldsymbol{y})$, terms which satisfy every predicate in O should be restricted to constants used in the substitution

γ_a. It means that $A(\boldsymbol{x})\theta \cup K \cup O$ should be minimal with respect to each predicate in O. Minimality with respect to a predicate is that the extension of the predicate (a set of tuples which satisfy the predicate) is minimal (Davis, 1980). The extension of a predicate in O for $A(\boldsymbol{x})\theta \cup K \cup O$ should be the same to the extension for O. This restriction can find a substitution θ_a for $A(\boldsymbol{x})$. Abductive procedures with the resolution principle can find this substitution. But still $A(\boldsymbol{x})\theta_a$ can have free variables. Then these free variables in $A(\boldsymbol{x})\theta_a$ are assigned either to instantiated constants or to constants in O. Here θ_s stands for a substitution from variables to variables, θ_i for a substitution from variables to instantiated constants. Then $A = A(\boldsymbol{x})\theta_s\theta_i\theta_a$. θ_s represents identification between different terms, i.e., the way which entities in hypotheses should be identified.

For example, suppose

$$\{is\_alive(x) \wedge has(x,y) \wedge wing(y) \wedge is\_feather(y) \rightarrow bird(x),\\ has(x,y) \wedge wing(y) \wedge is\_big(y) \rightarrow fly(x)\}$$

as K and $\{bird(a),\ fly(a)\}$ as O. If there are no ideas to identify entities, both

$$A_1 = \{is\_alive(a),\ has(a,b),\ wing(b),\ is\_feather(b),\ has(a,c),\\ wing(c),\ is\_big(c)\}$$

and

$$A_2 = \{is\_alive(a),\ has(a,b),\ wing(b),\ is\_feather(b),\ is\_big(b)\}$$

can be hypotheses. The former seems redundant, but both hypotheses are minimal because $A_1 \not\supset A_2$ and $A_1 \not\subset A_2$. The difference is the way how to introduce entities in hypotheses.

4.2.2. Minimality of Entities in Explanation

One of criteria to integrate hypotheses is minimality of entities. Domain circumscription (McCarthy, 1980) can be used to achieve minimality of entities in explanations. Domain circumscription finds models that have minimal domains to hold given formulae. In this case $A(\boldsymbol{x})\theta_a \wedge K \wedge O(\boldsymbol{a})$ is a formula to circumscribe. But using domain circumscription without any restrictions will make undesirable results. For the above example, we can get

$$\{is\_alive(a),\ has(a,a),\ wing(a),\ is\_feather(a),\ is\_big(a)\}$$

as a hypothesis with domain circumscription. This hypothesis seems unnatural, because we have knowledge about what kind of entities can be unified or not. In this case, entities which can satisfy $wing(x)$ and $bird(x)$ should be different, while entities which can satisfy $wing(x)$ can be unified to each other[1].

Superposition is identification between entities, but it is specified by two propositions which have entities to be identified.

Although it is impossible to describe all possible unifiable entity relations in knowledge[2], we can postulate at least consistency of aspect theories. Relations among predicates in an aspect are all what are written in the aspect theory. If two proposition have predicates in the same aspect, they are not allowed to identify unless these predicates are the same.

Suppose

$$K_1 = \{is\_alive(x) \wedge has(x,y) \wedge wing(y) \wedge is\_feather(y) \rightarrow bird(x)\}$$

$$K_2 = \{part(x,y) \wedge lift\_force\_device(y) \rightarrow fly(x)\}$$

$$K = K_1 \cup K_2$$

$$O = bird(a) \wedge fly(a)$$

where K_1 and K_2 are aspect theories. We can get a hypothesis

$$\begin{aligned} A = &\{is\_alive(a),\ has(a,b),\ wing(b),\ is\_feather(b),\ part(a,c),\\ &\ lift\_force\_device(c)\}. \end{aligned}$$

If we assume superposition

$$\{has(x,y),\ part(x,y)\}$$

and

$$\{wing(x),\ lift\_force\_device(x)\},$$

then the hypothesis is

$$\begin{aligned} A' = &\{is\_alive(a),\ has(a,b),\ part(a,b),\ wing(b),\ is\_feather(b),\\ &\ lift\_force\_device(b)\}. \end{aligned}$$

Notice such superposition is also a hypothesis, and validity of the superposition is examined by deduction and further abduction from the whole or part of the hypothesis A'. In particular, part of the hypothesis which includes identified entities is important in further abduction and deduction in order to realize how the superposition is feasible. In this example, it is $\{wing(b),\ is\_feather(b),\ lift\_force\_device(b)\}$.

---

[1] It is not a matter of course. If there are more than two entities which satisfy the same predicate, each of such predicates can be related to different entities.

[2] It is the frame problem to enumerate all combinations among predicates (McCarthy and Hayes, 1969).

4.3. EXPLANATORY COHERENCE

Ng and Mooney (1990) proposed *explanatory coherence* as the primary measure to evaluate the quality of an explanation. Explanatory coherence computes the degree of connectivity of a hypothesis as follows;

$$C = \frac{\sum\limits_{1 \leq i \leq j \leq l} N_{i,j}}{Nl(l-1)/2}$$

where l is the total number of the observation, N is the total number of nodes in the proof graph, and $N_{i,j}$ is the number of distinct nodes n_k in the proof graph such that there is a sequence of directed edges from n_k to n_i and also n_k to n_j where n_i and n_j are elements of the observation.

This quantity may be useful to compare some tightly connected hypotheses, but we need a more qualitative scale to evaluate coherence of explanations where connectivity is not so tight, and finding connectivity of explanation itself is one of purposes of abduction.

Here we introduce a *coherent segment of explanation* to evaluate explanations.

DEFINITION 5. *When an explanation* $\langle A, K\rangle$ *for an observation* O *is given, a partial explanation* $\langle A(O'), K(O')\rangle$ *for the observation* $O' \subset O$ *is defined as follows;*

$A(O')$ *and* $K(O')$ *are both minimal sets of formulae that satisfy* $A(O') \subseteq A$, $K(O') \subseteq K$ *and* $A(O') \cup K(O') \models O'$.

In case of multiple partial explanations, we denote $A(O')[i]$ and $K(O')[i]$.

DEFINITION 6. *Given an explanation* $\langle A, K\rangle$ *for* O, $O_1 \subset O$ *and* $O_2 \subset O$ *are directly connected to each other if and only if* $\bigcup_{i,j}(A(O_1)[i] \cap A(O_2)[j]) \cup \bigcup_{i,j}(K(O_1)[i] \cap K(O_2)[j])$ *is non empty. This set is called "direct connection of explanation between* O_1 *and* O_2*".*

Direct connection of an explanation corresponds there exists n_k for specified n_i and n_j in Ng and Mooney's definition.

DEFINITION 7. *Given an explanation* $\langle A, K\rangle$ *for* O, $O_1 \subset O$ *and* $O_2 \subset O$ *are indirectly connected to each other if* O_1 *and* O_2 *are directly connected to each other or there is* $O_3 \subset O$ *that is indirectly connected to both* O_1 *and* O_2.

DEFINITION 8. *Given an explanation* $\langle A, K\rangle$ *for* O, *if every element of* $O' \subseteq O$ *is indirectly connected to other element in* O', *and any element in* $O - O'$ *is not indirectly connected to element in* O', $\langle A(O'), K(O'), O'\rangle$ *is a coherent segment of explanation.*

If the number of coherent segments for an explanation is 1, the whole explanation is connected. If the number is more than 1, the explanation includes some explanations that are not related to each other. This coherent segments are calculated by tracing dependency of members of hypotheses.

4.4. INFORMATION FOR NEXT INFERENCES

Abduction can generate useful results for next inferences.

As discussed, an explanation consists of not only a hypothesis but also an explanatory theory. Furthermore we can obtain superposition of propositions to identify entities, and coherent segments of the explanation.

These types of information are used by two different ways, i.e., for successive inference steps, and for inferences to solve similar problems.

Explanatory theories and explanatory coherence are useful for successive inferences. An explanatory theory is a primary theory for deduction that is used to confirm the hypothesis proposed by abduction, i.e., deduction should use at least this theory. And it is also a reference theory for the next abduction. Clusters of knowledge in an explanatory theory are most plausible candidate theories for the next abduction.

Explanatory coherence helps us to determine the extent of next problems. If there are some coherence segments, it seems reasonable to divide the current problem. Even if there is only one coherent segment, we can form sub problems by direct or indirect connections among members in the observation and the hypothesis.

Explanatory theories and superposition of propositions are useful to solve similar problems. The sets of clusters that are used in abduction are examples of adherence among knowledge. And superposition of propositions is also examples of connections among predicates in different aspect theories. By gathering such information, we can develop the inter-aspect theory, or add more descriptions in meta-level knowledge (see Section 2.2.3).

5. Design simulation

We have been developed a prototype system called *design simulator* to evaluate our inference model discussed in Section 2 and Section 4. This system has two purposes. One to show computability of the proposed model, and the other is to show adaptability to actual design processes. Figure 3 is the outline of the design simulator and Figure 4 is a snapshot of the system. This system is implemented in Lucid/Allegro Common Lisp, CLX (Common Lisp X interface), and X11 on Decstation/Sparcstation.

We show how abduction can explain designers' processes by using a simple example taken from protocols in design experiments (Takeda, Hamada,

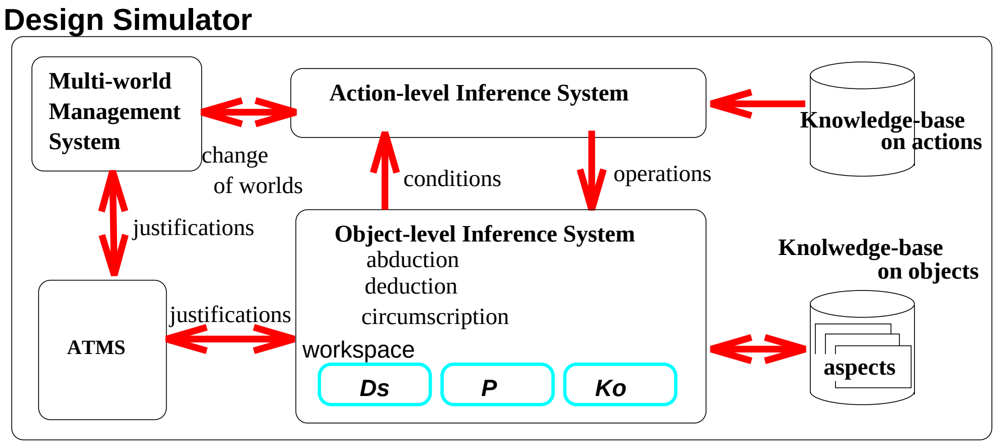
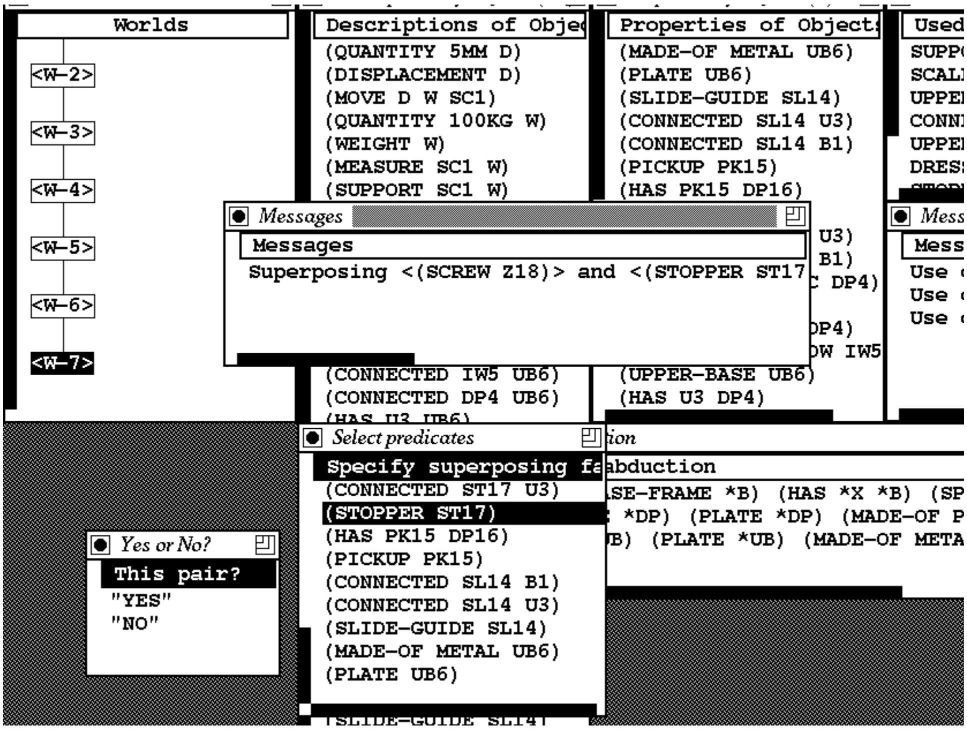


Fig. 3. The architecture of the design simulator



```
Descriptions of Objec
(QUANTITY 5MM D)
(DISPLACEMENT D)
(MOVE D W SC1)
(QUANTITY 100KG W)
(WEIGHT W)
(MEASURE SC1 W)
(SUPPORT SC1 W)
(CONNECTED IW5 UB6)
(CONNECTED DP4 UB6)
```

```
Properties of Object
(MADE-OF METAL UB6)
(PLATE UB6)
(SLIDE-GUIDE SL14)
(CONNECTED SL14 U3)
(CONNECTED SL14 B1)
(PICKUP PK15)
(HAS PK15 DP16)
U3)
B1)
DP4)
DP4)
DW IW5
(UPPER-BASE UB6)
(HAS U3 DP4)
```

```
Used
SUPP
SCAL
UPPE
CONN
UPPE
DRES
```

```
Messages
Superposing <(SCREW Z18)> and <(STOPPER ST17
```

```
Mess
Use
Use
Use
```

```
Specify superposing fa
(CONNECTED ST17 U3)
(STOPPER ST17)
(HAS PK15 DP16)
(PICKUP PK15)
(CONNECTED SL14 B1)
(CONNECTED SL14 U3)
(SLIDE-GUIDE SL14)
(MADE-OF METAL UB6)
(PLATE UB6)
```

```
abduction
SE-FRAME *B) (HAS *X *B) (SP
*DP) (PLATE *DP) (MADE-OF P
B) (PLATE *UB) (MADE-OF META
```

```
This pair?
"YES"
"NO"
```

Fig. 4. A snapshot of the design simulator

Tomiyama and Yoshikawa, 1990)[3]. We interpret this design session as an inference by knowledge. We pick up pieces of protocol (verbal protocol and figures), and represent them as logical formulae. Inference procedures in the system has been described in Ref. (Takeda, Tomiyama, Yoshikawa and Veerkamp, 1990).

This session is an interactive process, i.e., every time the system finishes one step of its inference, it shows results and possible actions to do next, and then the user specifies one of them.

In this session, we assume five aspects, i.e., *scale aspect, exterior-design aspect, support-motion aspect, translate-motion aspect*, and *manufacturing aspect.* These aspects have rules that are representation of designers' knowledge in this design session.

The specification is to design a scale. It means to design an object that can support and measure given weight (see Figure 5(a)).

Then designers suggest a structure of typical scales(see Figure 5(b)). Revered lines are newly added members of the hypothesis.

Since "`(support sc1 w) (measure sc1 w) (weight w) (quantity 100kg w) (move d w sc1) (displacement d) (quantity 5mm d)`" have not been abduced yet and other members of the hypothesis are connected to each other at this moment, there are eight coherent segments. Then they abduce "`(support sc1 w)`"and get a hypothesis using the following rule;

```
(support *s *w) <-
          (upper-frame *u)(has *s *u)(base-frame *b)(has *s *b)
          (slide-guide *sl)(connected *sl *u)(connected *sl *b)
          (pickup *pk)(has *pk *sc)
          (stopper *st)(connected *st *u)(connected *st *b))
```

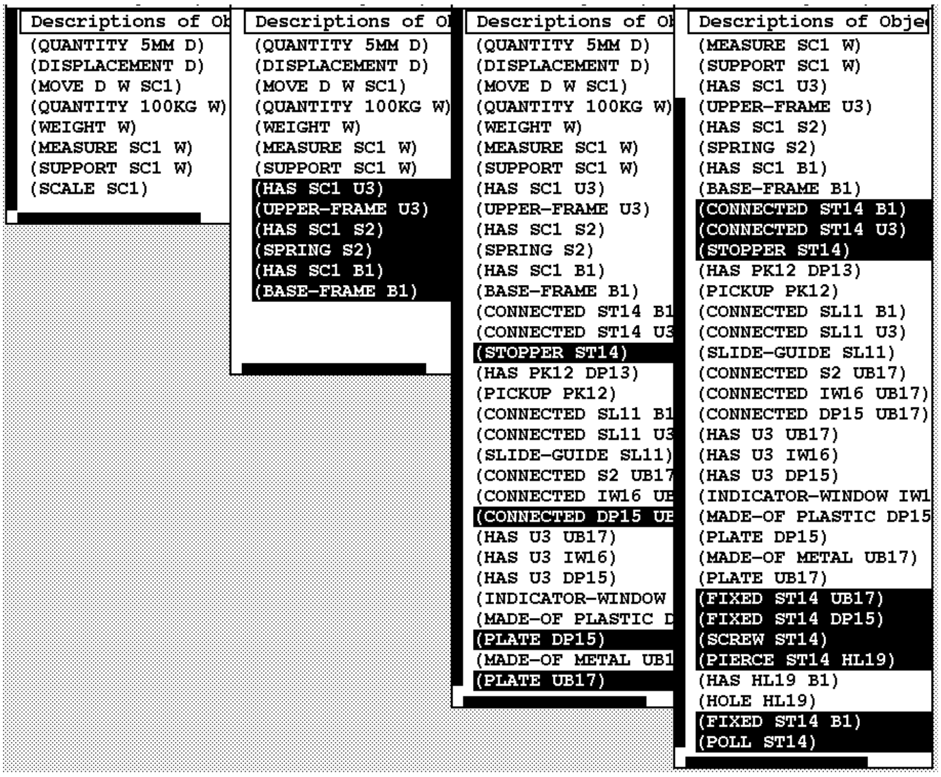
This rule introduces "`(stopper st14)`" and "`(slide-guide sl11)`', and also makes these propositions connect to the segment which includes "`(upper-frame u3)`" and "`(base-frame b1)`". Thus the number of coherent segments is decreased. Using the exterior-design aspect, they get a new hypothesis as design descriptions shown in Figure 5(c).

Furthermore they decide to connect the plastic cover and the upper base by screws. "`(connected dp15 ub17)`" is abduced to "`(screw z18)(fixed z18 ub17)(fixed z18 dp15)`" by a rule in manufacturing aspect, Then they notice these screws can be used as the stopper of the vertical movement. They identify "`(screw z18)`" and "`(stopper st14)`". Figure 4 is a

---

[3] This example is taken from other experiments that we used in Ref. (Takeda, Veerkamp, Tomiyama and Yoshikawa, 1990). According to names in Ref. (Takeda, Hamada, Tomiyama and Yoshikawa, 1990), we use experiment No. I–1 in this paper, while we used No. I–3 in Ref. (Takeda, Veerkamp, Tomiyama and Yoshikawa, 1990).

snapshot of the design simulator when superposition of propositions is asked to users. Using some other rules, they can get a hypothesis shown in Figure 5(d). In this hypothesis, revered lines are descriptions of the entity which should play two roles in this design. One problem they should solve next is to develop and examine descriptions of this entity.



```
Descriptions of Ob
(QUANTITY 5MM D)
(DISPLACEMENT D)
(MOVE D W SC1)
(QUANTITY 100KG W)
(WEIGHT W)
(MEASURE SC1 W)
(SUPPORT SC1 W)
(SCALE SC1)
```

```
Descriptions of O
(QUANTITY 5MM D)
(DISPLACEMENT D)
(MOVE D W SC1)
(QUANTITY 100KG W)
(WEIGHT W)
(MEASURE SC1 W)
(SUPPORT SC1 W)
(HAS SC1 U3)
(UPPER-FRAME U3)
(HAS SC1 S2)
(SPRING S2)
(HAS SC1 B1)
(BASE-FRAME B1)
```

```
Descriptions of Ob
(QUANTITY 5MM D)
(DISPLACEMENT D)
(MOVE D W SC1)
(QUANTITY 100KG W)
(WEIGHT W)
(MEASURE SC1 W)
(SUPPORT SC1 W)
(HAS SC1 U3)
(UPPER-FRAME U3)
(HAS SC1 S2)
(SPRING S2)
(HAS SC1 B1)
(BASE-FRAME B1)
(CONNECTED ST14 B1
(CONNECTED ST14 U3
(STOPPER ST14)
(HAS PK12 DP13)
(PICKUP PK12)
(CONNECTED SL11 B1
(CONNECTED SL11 U3
(SLIDE-GUIDE SL11)
(CONNECTED S2 UB17
(CONNECTED IW16 UB
(CONNECTED DP15 UB
(HAS U3 UB17)
(HAS U3 IW16)
(HAS U3 DP15)
(INDICATOR-WINDOW
(MADE-OF PLASTIC D
(PLATE DP15)
(MADE-OF METAL UB1
(PLATE UB17)
```

```
Descriptions of Obje
(MEASURE SC1 W)
(SUPPORT SC1 W)
(HAS SC1 U3)
(UPPER-FRAME U3)
(HAS SC1 S2)
(SPRING S2)
(HAS SC1 B1)
(BASE-FRAME B1)
(CONNECTED ST14 B1)
(CONNECTED ST14 U3)
(STOPPER ST14)
(HAS PK12 DP13)
(PICKUP PK12)
(CONNECTED SL11 B1)
(CONNECTED SL11 U3)
(SLIDE-GUIDE SL11)
(CONNECTED S2 UB17)
(CONNECTED IW16 UB17)
(CONNECTED DP15 UB17)
(HAS U3 UB17)
(HAS U3 IW16)
(HAS U3 DP15)
(INDICATOR-WINDOW IW1
(MADE-OF PLASTIC DP15
(PLATE DP15)
(MADE-OF METAL UB17)
(PLATE UB17)
(FIXED ST14 UB17)
(FIXED ST14 DP15)
(SCREW ST14)
(PIERCE ST14 HL19)
(HAS HL19 B1)
(HOLE HL19)
(FIXED ST14 B1)
(POLL ST14)
```

(a) (b) (c) (d)

Fig. 5. Changing of descriptions of design objects

6. Abduction as Fundamental Reasoning in Future CAD

It is important for future CAD systems to keep integration of knowledge. Nowaday we have been putting models and knowledge from various backgrounds in computer in order to support design. But there are no unified methods to integrate them. Since design is not archived with a single model, but various perspectives should be taken into account, integration of various models and knowledge is crucial to realize future CAD systems.

Furthermore future CAD systems should have not only ability of exchanging information in various models to each other, but also ability of guiding use of various models. That is, static and dynamic integration of models are required.

Methodology to exchange information among models tends to deal with objects and aspects ontologically, while integrated use of models in design tends to deal with objects and aspects teleologically. If every relation among models be clear enough, there would be no reason to be teleological. Since we could not expect such a situation in design, designers assume relations under their purpose. Then such relations are very vulnerable and should be examined by experiment and manufacturing. But if it turns out that they are true and useful relations, then they can be included into ontological knowledge.

As an approach to model integration, Tomiyama, Kiriyama, Takeda and Xue (1989) proposed the concept of metamodel for a new modeling framework for design objects. The metamodel is used as (1) as a central modeling mechanism to integrate models, (2) as a mechanism for modeling physical phenomena, and (3) as a tool for describing evolving design objects. Each model in CAD systems is connected only through metamodel where physical phenomena as concepts are used to describe objects. They also proposed a metamodel system based on qualitative physics (Xue, Takeda, Kiriyama, Tomiyama and Yoshikawa, 1991). Here qualitative physics plays an inter-aspect theory among models.

The metamodel based on qualitative physics can provide basic and common connections among aspect models, but it is not appropriate to represent and describe integration generated in design processes. Integration of aspects differs in every design, and furthermore it is also the goal of design because objects should be represented as integration of aspects. Abduction we discussed here is an approach to realize to create such integration.

Abduction can support designers to achieve dynamic integration of knowledge. As mentioned in Section 4.4, both explanatory theories and superposed propositions represent dynamic integration of knowledge. An explanatory theory is an example of combination of knowledge. Superposed propositions is also an example how descriptions in different aspects can be related to each other. Accumulation of these examples in successful design processes shows how knowledge is used in design processes. This information is useful to re-organize knowledge-bases and aspects as well as to create new relationship among different aspects.

Thus corporation of two types of integration of knowledge could make CAD systems more flexible and more designer-oriented.

7. Conclusion

In this paper, we explained the logical framework for design processes in which iteration of abduction and deduction is main process to proceed design. Since abduction is a key inference in this framework, we investigated abduction in design, and proposed an abductive inference for design.

Unlike other definition of abduction, we assumed variety of background theory, and defined abduction as reasoning to integrate theories to obtain explanation. It means that abduction serves not only to obtain hypotheses, but also to propose integration of background theory. We also demonstrated how this abductive inference works in design processes.

Since integration of knowledge is a key issue for future CAD systems, this framework gives an answer to deal with variety of knowledge in CAD systems.

Acknowledgements

The author would like to thank Prof. Øyvind Bjørke for his advice and support. The author would also like to thank Prof. Hiroyuki Yoshikawa and Prof. Tetsuo Tomiyama of the University of Tokyo. The motivation and background material were developed while the author was in the University of Tokyo. The help provided by all former colleagues is gratefully acknowledged.

References

Cox, P. and Pietrzykowski, T.: 1986, Causes for events: their computation and applications, *Lecture Notes in Computer Science 230*, Springer-Verlag, Berlin, pp. 608–621.

Coyne, R.: 1988, *Logic Models of Design*, Pitman Publishing, London.

Davis, M.: 1980, Notes on the mathematics of non-monotonic reasoning, *Aritifical Intelligence*.

Dietterich, T. and Ullman, D.: 1987, FORLOG: A logic-based architecture for design, *Rep. no. 86-30-8*, Computer Science Department, Oregon State University.

Fann, K.: 1970, *Peirce's Theory of Abduction*, Martinus Nijhoff, The Hague, The Netherlands.

Finger, J. and Genesereth, M.: 1985, RESIDUE: A deductive approach to design synthesis, *Technical report stan-cs-85-1035*, Stanford University.

Finger, S. and Dixon, J.: 1989, A review of research in mechanical engineering design. part I: descriptive, prescriptive, and computer-based models of design processes, *Research in Engineering Design* **1**(1), 51–67.

Hobbs, J., Stickel, M., Martin, P. and Edwards, D.: 1988, Interpretation as abduction, *Proceedings of 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, NY, pp. 95–103.

Landman, F.: 1986, *Towards a Theory of Information: the Status of Partial Objects in Semantics*, Foris, Dordrecht.

Levesque, H.: 1989, A knowledge-level account of abduction, *Proceedings IJCAI-89*, Detroit, pp. 1061–1067.

Lifschitz, V.: 1985, Computing circumscription, *Proceedings IJCAI-85*, Los Angles, CA, pp. 121–127.

Lloyd, J.: 1984, *Foundations of Logic Programming*, Springer-Verlag, Berlin.

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

McCarthy, J. and Hayes, P.: 1969, Some philosophical problems from the standpoint of artificial intelligence, *Machine Intelligence* **4**, 463–502.

Ng, H. and Mooney, R.: 1990, On the role of coherence in abductive explanation, *Proceedings AAAI-90*, pp. 337–342.

Peirce, C.: 1935, *Collected Papers of Charles Sanders Peirce*, Vol. 5, Harvard University Press, Cambridge, MA.

Poole, D.: 1988, A logical framework for default reasoning, *Artificial Intelligence* **36**, 27–47.

Pople, Jr., H. E.: 1973, On the mechanization of abductive logic, *Proceedings IJCAI-73*, Stanford, CA, pp. 147–152.

Takeda, H., Hamada, S., Tomiyama, T. and Yoshikawa, H.: 1990, A cognitive approach of the analysis of design processes, *Design Theory and Methodology – DTM '90 –*, The American Society of Mechanical Engineers (ASME), pp. 153–160.

Takeda, H., Tomiyama, T. and Yoshikawa, H.: 1990, A logical formalization of design processes for intelligent CAD systems, *in* H. Yoshikawa and T. Holden (eds), *Intelligent CAD, II*, North-Holland, Amsterdam, pp. 325–336.

Takeda, H., Tomiyama, T. and Yoshikawa, H.: 1992, A logical and computerable framework for reasoning in design, *in* D. Taylor and L. Stauffer (eds), *Design Theory and Methodology – DTM '92 –*, The American Society of Mechanical Engineers (ASME), pp. 167–174.

Takeda, H., Tomiyama, T., Yoshikawa, H. and Veerkamp, P.: 1990, Modeling design processes, *Technical Report CS-R9059*, Centre for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands.

Takeda, H., Veerkamp, P., Tomiyama, T. and Yoshikawa, H.: 1990, Modeling design processes, *AI Magazine* **11**(4), 37–48.

Tomiyama, T., Kiriyama, T., Takeda, H. and Xue, D.: 1989, Metamodel: A key to intelligent CAD systems, *Research in Engineering Design* **1**, 19–34.

Treur, J.: 1991, A logical framework for design processes, *in* P. ten Hagen and P. Veerkamp (eds), *Intelligent CAD Systems III — Practical Experience and Evaluation*, Springer-Verlag, Berlin.

Veltman, F.: 1981, Data semantics, *in* J.A.G.Groenendijk, T. Janssen and M.B.J.Stokhof (eds), *Formal methods in the study of language*, Mathematisch Centrum, Amsterdam, pp. 541–565.

Weyhrauch, R.: 1980, Prolegomena to a theory of mechanized formal reasoning, *Artificial Intelligence* **13**, 133–170.

Xue, D., Takeda, H., Kiriyama, T., Tomiyama, T. and Yoshikawa, H.: 1991, An intelligent integrated interactive CAD — a preliminary report, *Proceedings of the IFIP 5.2 Working Conference on Intelligent Computer Aided Design*, Ohio, USA.