INTEGRATION OF ASPECTS IN DESIGN PROCESSES

HIDEAKI TAKEDA AND TOYOAKI NISHIDA
Graduate School of Information Science
Nara Institute of Science and Technology
address: 8916-5, Takayama, Ikoma, Nara 630-01, Japan
phone: +81-7437-2-5260
fax: +81-7437-2-5219
E-mail: {takeda, nishida}@is.aist-nara.ac.jp

Abstract. In this paper, we discuss dynamic integration of multiple aspects, i.e., integration accomplished according to progress of design. It is not prepared in advance, but created in design processes. Firstly, we introduce our model of design processes that is based on a logical framework. Secondly, we define aspects in the logical framework. An aspect is represented as a tuple of theory and vocabulary in the logical framework. In particular knowledge in analytical aspects is represented as virtual logical theory. Thirdly, we propose integration of aspects by abduction that is another approach than integration of models. Abduction defined with multiple aspects integrates aspects by superposition of hypothesis which is identification of instantiated entities in hypothesis. It also examines connectivity of hypotheses by explanatory coherence. Since superposition of hypotheses and theories used in abduction tell us how aspects are integrated in design, they can contribute to re-organize aspect knowledge-bases.

1. Introduction

Designers use different kinds of aspects when they recognize artifacts. Some aspects have been developed in traditional engineering fields and have firm theories like kinematics and electric circuits. Other aspects are more vague and have not established firm theories like cost estimation, and manufacturability. Some aspects are numerical, others are symbolic or linguistic.

It is nature of design to take different aspects into consideration. Even if purpose of design can be described in a single aspect, artifacts in the real world would receive various kind of effects which come from not only the original aspect but many different aspects. Designers, thus, should consider various aspects in order to accomplish design successfully.

Traditional design studies emphasize uniqueness of representation of artifacts and therefore dismiss importance of variety of aspects.

On the other hand, various kind of analysis methods have been developed in the engineering field. They emphasize completeness of their methods and representation of artifacts. They ignore aspects behind themselves, which are important to use these analysis methods in design.

In this paper, we discuss dynamic integration of multiple aspects. Dynamic integration means that integration is accomplished according to progress of design. It is not prepared in advance, but created in design processes. Firstly, we introduce our model of design processes that is based on a logical framework. Secondly, we define aspects in the logical framework. In this definition, knowledge in analytical aspects is represented as virtual logical theory. Thirdly, we propose integration of aspects by abduction that is another approach than integration of models. Then we show examples with our prototype system. Finally we conclude the paper.

2. Logical Design Process Modeling

We need a theory about design which is *formal*, *general* and *descriptive* in order to understand and represent design. And considering to apply it to CAD (Computer-Aided Design) system, it should be also computable so that computation systems would be drawn from it. There are many design models proposed such as Pahl *et al.*(1984), Hubka(1988), and Suh(1990), but they are not sufficient for above requirements. For example, Pahl and Beitz's approach is specific to domains, and Suh's approach is too prescriptive.

We have proposed a logical model of design processes (Takeda *et al.*, 1990c) (Takeda *et al.*, 1990d) as a descriptive and also computable model. A design process is interpreted as combination of inferences defined in the logical framework in this model. In this section, we depict our design process model shortly. Details are shown in Ref. (Takeda *et al.*, 1990c) and (Takeda *et al.*, 1990d).

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, i.e., 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 *et al.*(1987) took this approach, and we also took it in Ref. (Takeda *et al.*, 1990a). 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. For example, solutions and knowledge are always incomplete in design, but 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 *et al.*, 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 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 K_O, 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 evolves design objects and their properties and behaviors,

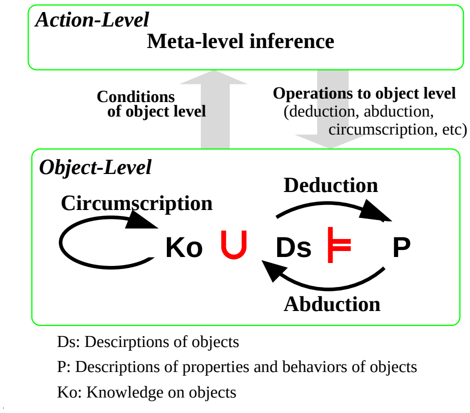


Figure 1. The logical design process model

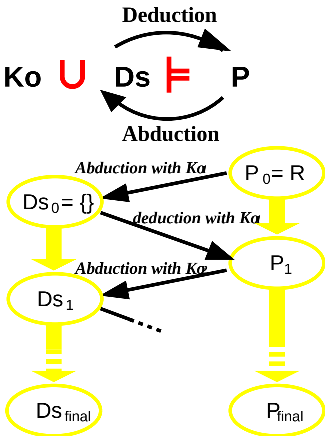


Figure 2. Iteration of abduction and deduction

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, K_O, 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.

2.3. ITERATION OF ABDUCTION AND DEDUCTION AS THE BASIC PROCESS

We interpret a design process as an evolutionary process, that is, the design objects are refined in step-wise manner (see Figure 2). We call each state of step-wise refinement as a *design state*. In each state, the following three types of descriptions are 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 P, descriptions of properties and behaviors of the current design solution. 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 K_O, knowledge that is available at the current state. These descriptions are kept consistent to satisfy the following formula;

$$Ds \cup K_O \;\vdash\; P.$$

Given design knowledge K_O and the required properties P as the specifications, designers try 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 designers either try an alternative solution or modify 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).

3. Integration of Multiple Aspects

3.1. TELEOLOGICAL INTEGRATION VS. ONTOLOGICAL INTEGRATION

It is important for future CAD systems to keep integration of knowledge. In order to support design, we have been putting models and knowledge from various backgrounds in computer. But there are no unified methods to integrate them. Since design is not archived with a single model and 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, both *ontological* and *teleological* integrations of models are required.

The current approach to integrate multiple aspects is to integrate aspect models, for example, product model(Suzuki *et al.*, 1990), STEP, metamodel(Tomiyama *et al.*, 1990). We can summarize this approach as *ontological integration* of aspects theories, because it aims to establish relations among representations of objects, i.e., relations among different ontologies(Gruber, 1993).

As an approach to model integration, Tomiyama *et al.*(1990) 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 *et al.*, 1991). Here qualitative physics plays as an inter-aspect theory among models.

Model integration approach provides basic and common connections among aspect models, but it is not *all* information to integrate aspects.

To design new objects, in particular in creative design, yields new relations among aspects. Creative design does not happen within a single existing aspect, but with a new aspect which is new combination of existing aspects. New combination means that designers find new way to combine aspects, i.e., new relations among aspects. To design objects creatively is, thus, to find new relations among aspects that have not been recognized yet. It is another kind of integration of aspects which is guided by designers' intension. We thus call them *teleological integration* of aspects.

It is to notice that such relations are not firm ones until design is completed, because they just depend on designers' intension and are never examined in the real world. They should be examined by experiment and manufacturing.

For example, suppose that a screw is introduced in a design from structural aspect and a stopper of linear movement from kinematics aspect. Then a designer decides to use the screw as the stopper. In this case an inter-aspect relation between the kinematics aspect and structural aspect is arisen. The designer is not sure that this relation, i.e., "screw as stopper" is really true before precise estimation of geometry, but s/he tries to keep it unless it turn out false. If screw as stopper is a general idea not but a special case in a special situation, it can possibly be added to ontological relations between kinematics and structural views. In this paper we formalize this process by abduction with multiple aspects.

Another problem in integration of aspects is that even relations between aspects which are conceptually clear are often difficult to describe in a formal way, because ontological descriptions would be exhaustive. For example, interpretation of results of stress analysis is conceptually clear, but it is not easy task to describe relation between stress analysis aspect and some designing aspect, i.e., how it would affect design processes. For this problem, we propose a virtual theory to describe relations between analytical aspects and synthetic aspects. A virtual theory is a logical description of an analytical (non-logical) aspect from a point of view of using the aspect.

3.2. ASPECTS IN THE LOGICAL FRAMEWORK

In logical design process modeling, we assumed a single theory K_o as designers' knowledge. As we mentioned, it is not a good assumption to deal with multiple aspects. So we re-define theory in logical process modeling.

Instead of assuming a single theory, we here assume a set of theories, i.e., the theory is divided into separate *aspect theories* each of which has its own perspective of description.

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. Every aspect has its vocabulary so that any propositions can be determined whether they are in the aspect or not. Because purpose of aspects is to show a consistent and independent view of the real world, aspect theories should be consistent and closed for its vocabulary.

Definition 1 *(Aspect)*
An ***aspect*** *A_i is a tuple of an aspect theory K_i and vocabulary V_i. The aspect theory is a set of formulae. An aspect $\langle K_i, V_i \rangle$ should satisfy the following conditions;*

1. *K_i is consistent, and*
2. *Any atomic formulae in K_i is within V_i.*

Furthermore we assume clusters in an aspect theory. Although an aspect theory can be huge, what is needed in design is not always the whole of an aspect theory but some part of the aspect theory. We assume that an aspect theory consists of a set of cluster theories.

Definition 2 *(Aspect theory)*
An ***aspect theory*** *K_i is union of cluster theories $KC_i^j (j \in \Lambda_i)$, i.e.,*

$$K_i = \bigcup_{j \in \Lambda_i} KC_i^j$$

where Λ_i is a set of identifiers for cluster theories in aspect theory K_i.

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.

Definition 3 *(Inter-aspect theory)*
An ***inter-aspect*** *for a set of aspects $\mathcal{A}$ is a tuple of inter-aspect theory K_I and vocabulary V_I that is union of vocabularies in $\mathcal{A}$, i.e.,*

$$V_I = \bigcup_{i \in \mathcal{A}} V_i$$

Then we can define a theory for design.

Definition 4 *(Background design theory)*
The ***background design theory*** *K_B is union of aspect theories K_i and the inter-aspect theory K_I, i.e.,*

$$K_B = \bigcup_{i \in \mathcal{A}} K_i \cup K_I,$$

where $\mathcal{A}$ is a set of aspect identifiers.

We can now define *an explanatory design theory*, i.e., a theory that is needed to design objects from given requirements.

Definition 5 *(Explanatory design theory)*

An ***explanatory design theory*** K_E *for the background design theory* K_B *is union of cluster theories taken from aspect theories in the background design theory, i.e.,*

$$K_E = \bigcup_{i \in \mathcal{A}} \bigcup_{j \in \mathcal{E}_i} KC_i^j \cup K_I,$$

where $\mathcal{E}_i$ *is selector of aspect theory* K_i*, which is subset of its cluster theory indicators, i.e.,* $\mathcal{E}_i \subset \Lambda_i$*.*

An explanatory design theory is defined as collection of clusters of knowledge selected from knowledge of aspects (see Figure 3).

3.3. VIRTUAL LOGICAL THEORY

Aspects in the engineering field are so various in representation scheme and in reasoning style that it is impossible to provide a single representation scheme with a single reasoning style that covers all the aspects.

Instead of a representation scheme covering for all the aspects, we assume a representation scheme that can be accessed from all the aspects. In our approach logic is the shared scheme. Every vocabulary in an aspect is defined in the logical framework. But it is impossible in general to represent whole of an aspect theory as a logical theory, because reasoning in some aspects is beyond logical reasoning. We describe every execution of inference as a formula, i.e, condition as premise and results as conclusion. Since such logical formulae would cover all the situations ultimately, we can say we could represent an aspect theory as a logical theory.

For example, suppose an analysis system that can calculate the maximum displacement of beam with given force, we can write a formula as follows;

cantilever(X) ∧ *form(X,A)* ∧ *vertical_force(F)* ∧ *contact_with_the_end(X,F)* ∧
beam_bending_calculation(X,F,D) → *maximum_displacement(X,D)*

In the antecedent there are conditions to determine whether this system is applicable and a predicate which is interface to the analysis system. In this example, *beam(X)* ∧ *form(X,A)* ∧ *vertical_force(F)* ∧ *contact_with_the_end(X,F)* are conditions for applying the aspect system, and *beam_bending_calculation(X,F,D)* is the interface term. Here A is a constant associated to a specific geometric form. It passes values of X and F to the analysis system, and returns a value of D as a result.

A virtual logical theory is a set of formulae each of which is combination of conditions to use analysis systems, and interfaces to them. From point of view of logical inference, it behaves like ordinal logical theories, and from point of view of application systems it acts as interface to other systems.

There can be many formulae each of which represents relations between two aspects in a specific situation. These formulae can be generated dynamically in design processes when such situations occur, and can be accumulated in a logical theory.

4. Integration of Aspects by Abduction

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 maintains changing of object descriptions.

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

4.1. RESEARCH ON ABDUCTION

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, for example Levesque(1989), Poole(1988), Cox *et al.*(1986), and Finger *et al.*(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 that $A \cup T \vdash O$ is hold and that $A \cup T$ 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. Abduction should include not only problem solving but also problem formation. 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. For the former issue, we propose abduction as a process which includes finding theory used in deduction-style inference. For the latter, we use structuralized theory and hypotheses according to aspects.

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 used to find explanation is called a *background theory.* A *hypothesis* is an idea conjectured by abduction.

4.2. DEFINITION OF ABDUCTION WITH MULTIPLE ASPECTS

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

Definition 6 *(Explanation)*
An ***explanation*** *of an observation O with a background theory K_B is $\langle A, K\rangle$, a tuple of an explanatory hypothesis A and an explanatory theory K which satisfy the following conditions;*

- $K \subseteq K_B$,
- $K \cup A$ *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).

The definition may seem identical to the definition in Section 4.1, 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. As we have mentioned, theory used in abduction is a theory which consists of part of aspect theories (see Definition 5).

Then we should discuss how integrated and creative abduction is realized in this framework. The key idea is that minimalization of hypotheses and explanation with given constraints. The first approach is to minimalize hypotheses and the second is to minimalize explanations.

4.3. SUPERPOSITION IN HYPOTHESES

According to the structure in explanatory theory, we can divide an explanatory hypothesis as follows;

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

Here, A_{TH} is derivative hypothesis that can be derived from the background theory and the observation. A_I is connective hypothesis that integrates members of the derivative hypothesis (see Figure 3). A derivative hypothesis A_{TH} alone can satisfy derivativeness of the observation O, i.e.,

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

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 hypothesis.

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 entities needed to solve the given problem. We 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. Here we assume that there are no constants in the explanatory theory.

Suppose $\boldsymbol{a} = \langle a_1, ..., a_k\rangle$ a tuple of constants appeared in the observation, $\boldsymbol{i} = \langle i_1, ..., i_m\rangle$ a tuple of instantiated constants appeared in the hypothesis

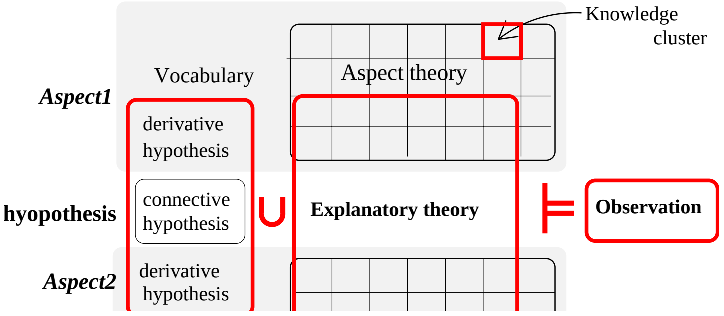


Figure 3. Hypothesis and explanatory theory

A, and $\boldsymbol{x} = \langle x_1, ..., x_n \rangle$ a tuple of variables. 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 elaborate to 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 $\gamma_a = \{a_1/y_1, ..., a_k/y_k\}$ is a substitution. Then,

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

The fact that the observation is given as $O(\boldsymbol{y})\gamma_a$ not as $O(\boldsymbol{y})$ indicates that terms which satisfy every predicate in O should be restricted to constants used in the substitution γ_a. It means that $A(\boldsymbol{x})\theta \cup K \cup O(\boldsymbol{y})$ 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 $A(\boldsymbol{x})\theta \cup K \cup O(\boldsymbol{y})$ should be the same to the extension for $O(\boldsymbol{y})\gamma_a$. This restriction is realized as a substitution θ_a for $A(\boldsymbol{x})$, which abductive procedures with the resolution principle can find. But $A(\boldsymbol{x})\theta_a$ can have free variables still. 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 $\{bird(a),\ fly(a)\}$ as O, and

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

as K. If there are no ideas to identify entities, both

$$\begin{array}{rcl} A_1 & = & \{is\_alive(a),\ has(a,b),\ wing(b),\ is\_feather(b),\ has(a,c), \\ & & \ wing(c),\ is\_big(c)\} \ \ \text{and} \end{array}$$

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

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.

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. 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, 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

$$\begin{aligned} K_1 &= \{is\_alive(x) \wedge has(x,y) \wedge wing(y) \wedge is\_feather(y) \rightarrow bird(x)\} \\ K_2 &= \{part(x,y) \wedge lift\_force\_device(y) \rightarrow fly(x)\} \\ K &= K_1 \cup K_2 \\ O &= bird(a) \wedge fly(a) \end{aligned}$$

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

$$\{has(x,y),\ part(x,y)\} \text{ and } \{wing(x),\ lift\_force\_device(x)\},$$

then the hypothesis is

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

It is to notice that 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 $\{wing(b),\ is\_feather(b),\ lift\_force\_device(b)\}$.

4.4. EXPLANATORY COHERENCE

Ng *et al.* (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;

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

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 tightly connected hypotheses, but we need 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 7 *(Partial explanation) 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 $K(O')[i]$.

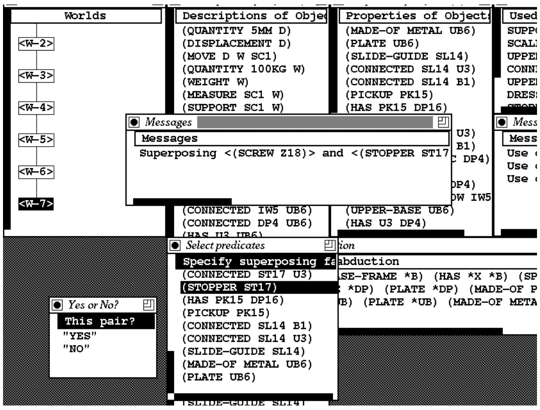
Definition 8 *(Direct connection)*
Given an explanation $\langle A, K \rangle$ *for* O, $O_1 \subset O$ *and* $O_2 \subset O$ *are* ***directly connected*** *to each other if and only if* $(A(O_1) \cap A(O_2)) \cup (K(O_1) \cap K(O_2)$ *is non empty. This set is called "direct connection of explanation between* O_1 *and* O_2*".*

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

Definition 9 *(Indirect connection)*
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 10 *(Coherent segment)*
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



```
(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)
```

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

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

```
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"
```

Figure 4. A snapshot of the design simulator

some explanations that are not related to each other. Those coherent segments are calculated by tracing dependency of members of hypotheses.

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 is to show computability of the proposed model, and the other is to show adaptability to actual design processes. 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 *et al.*, 1990b). 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 have been described in Ref. (Takeda *et al.*, 1990d).

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

Designers suggest a structure of typical scales(see Figure 5(b)). Highlighted lines indicate 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))
```

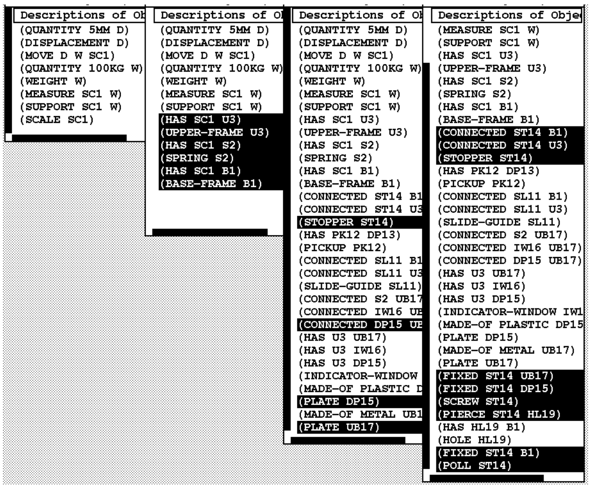
Application of this rule introduces “`(stopper st14)`” and “`(slide-guide sl11)`’, and also makes these propositions connect to the segment which includes “`(upper-frame u3)`” and “`(base-frame b1)`”. Thus the number of coherent segments is decreased. Using the exterior-design aspect, they get a new 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 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, highlighted lines are descriptions of the object which should act as both screw and stopper. One problem they should solve next is to develop and examine descriptions of this object.

6. Conclusion

We discussed representation and integration of aspects in design in this paper. We represent aspect knowledge as virtual logical theory. It enables to represent multiple aspects in a single framework. Then we discussed how different aspects are related in design. We categorize relationship between aspects in two types. One is ontological integration, and the other is teleological integration. The former is already established relations between aspects, and it is realized as model integration. The latter is relations which we wish to establish in design. We described it by abduction.

Abduction is reasoning to find feasible hypotheses from given theory. In this paper, we characterized abduction as integration of aspect theories. Our abduction with combination of multiple aspect theories proposes descriptions of objects supported by multiple aspects. Integration of isolated aspect theories is realized by superposition in hypothesis and evaluated by explanatory coherence. It is important to generate object descriptions from



(a)

```
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)
```

(b)

```
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)
```

(c)

```
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)
```

(d)

```
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)
```

Figure 5. Changing of descriptions of design objects

multiple aspects in design, because object descriptions not from a single aspect but from various aspects are necessary to create new objects in the real world. In other words, designing itself is integrating of aspects.

When we consider our abduction as knowledge-based system, we could conclude as follows. Abduction can support designers to achieve dynamic integration of knowledge. Both explanatory theories and superposed propositions represent dynamic integration of knowledge. A generated explanatory theory is an example of combination of knowledge used in design. Superposed propositions is also an example how descriptions in different aspects can be related to each other. Accumulation of these information in successful design processes shows how knowledge is used in design processes. It is useful to re-organize aspect knowledge-bases as well as to create new relationship among different aspect knowledge-bases. Thus cooperation of two types of integration of knowledge could make CAD systems more flexible and more designer-oriented.

Acknowledgement

Part of work in this paper was done when the first author was in Norwegian Institute of Technology (NTH). The author would like to thank Prof. Øyvind Bjørke of NTH for his useful advice and support. The author are also grateful to Prof. Hiroyuki Yoshikawa and Prof. Tetsuo Tomiyama of the University of Tokyo.

References

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

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

M. Davis. Notes on the mathematics of non-monotonic reasoning. *Aritifical Intelligence*, 13, 1980.

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

K.T. Fann. *Peirce's Theory of Abduction*. Martinus Nijhoff, The Hague, The Netherlands, 1970.

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

Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. Technical Report KSl 93-04, Knowledge Systems Laboratory, Stanford University, August 1993.

V. Hubka and W.E. Eder. *Theory of Technical Systems*. Springer-Verlag, Berlin, 1988.

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

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

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

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

G. Pahl and W. Beitz. *Engineering Design*. The Design Council, London, 1984.

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

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

N.H. Suh. *The Principles of Design*. Oxford University Press, New York, Oxford, 1990.

H. Suzuki, H. Ando, and F. Kimura. Synthesizing product shapes with geometric design constraints. In H. Yoshikawa and T. Holden, editors, *Intelligent CAD, II*, pages 309–324. North-Holland, Amsterdam, 1990.

H. Takeda, T. Tomiyama, and H. Yoshikawa. A logical formalization of design processes for intelligent CAD systems. In H. Yoshikawa and T. Holden, editors, *Intelligent CAD, II*, pages 325–336. North-Holland, Amsterdam, 1990.

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

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

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

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

J. Treur. A logical framework for design processes. In P.J.W. ten Hagen and P.J. Veerkamp, editors, *Intelligent CAD Systems III — Practical Experience and Evaluation*. Springer-Verlag, Berlin, 1991.

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