INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 03 STOCKHOLM, AUGUST 19-21, 2003

UNIVERSAL ABDUCTION STUDIO —PROPOSAL OF A DESIGN SUPPORT ENVIRONMENT FOR CREATIVE THINKING IN DESIGN—

Hideaki Takeda, Hiromitsu Sasaki, Yutaka Nomaguchi, Masaharu Yoshioka,
Yoshiki Shimomura and Tetsuo Tomiyama

Abstract

This paper describes a new design support system that supports conceptual or creative design by dynamically integrating knowledge in different design domains. We argue that abduction for integrating theories can be a basic principle to formalize such design processes. Based on this principle, we propose Universal Abduction Studio, a design environment in which designers combine different theories to arrive at better design. In this new approach to computational support of conceptual design, the system should offer various types of abductive reasoning from which designers can select an interesting design method. We also discuss technologies to implement UAS and in this paper we propose to use analogical reasoning as abductive reasoning to discover relationships between knowledge from different sources. We demonstrate that the system can discover a new idea in a design example taken from a real design activity.

Keywords: design process, abduction, computer-aided-design, analogy

1. Introduction

This paper presents "Universal Abduction Studio" (UAS) that is a new approach to computational support for designers and discusses technologies to implement it. Although design support systems have been well developed for geometric and detail design stages, they are still unsuccessful in the conceptual design stage. In our opinion, the main difficulty comes from incomplete and insufficient understanding about design knowledge and its operations that play a crucial role in conceptual design. In contrast, recently thanks to development of the Internet technologies, more and more knowledge is accumulated and available electronically. It then becomes an interesting research question how to apply such an enormous amount of diverse knowledge to conceptual design.

UAS is not intended to be an automated design system. UAS is an environment in which designers can manage and apply knowledge to arrive at new, creative design collaboratively with the system. UAS provides the designer with knowledge and reasoning functions, while the designer selects operations and evaluates design results iteratively.

In the following sections, we first discuss theoretical foundations for creative design support, i.e., how to represent creative design processes, and methods to achieve this. We then present the basic architecture of a prototype system of UAS.

2. Abduction for Design

The key issue to build a CAD system capable of supporting the early stages of design, in particular, its creativity aspects, is how to represent design processes. It is no doubt that a creative design process is one of the most intellectual thought processes and is difficult to model. This is not only because generating a creative product or idea itself is hardly inimitable by computers, but also because the knowledge used for creative design is generated, modified, and updated during the process. Once a designer achieves a new creative design after struggles, s/he is able to perform similar designs easily, which implies that her/his knowledge was expanded. We believe that the expansion of knowledge is a mandatory nature of creative design. Creative design therefore has two aspects, i.e., creating a new product and expanding knowledge, and the co-relation between these two is common to various creative activities.

How can we, then, model creative design processes with these two aspects? We have discussed formalization of design processes from the logical viewpoint [1]. In short, our theory models design as iteration of deduction and abduction (see also Coyne[2], Roozenburg and Eekels[3], while they did not offer any computing mechanisms). In our theory, the core part of design, i.e., creating a new idea or thing can be attributed to "abduction" while ensuring design to deduction. Abduction is thus the crucial part in design.

What is abduction and what can abduction offer as reasoning? Abduction proposed by C.S. Peirce is a logical process to find axiom from theorem [4]. The naïve interpretation is that abduction is an opposite process of deduction. Although this naïve interpretation is somewhat popular within computer science [5], abduction should be interpreted from wider viewpoints and therefore include more various types of reasoning. Schurz [6] collected various types of abductive reasoning and categorized them. In his work, abduction is firstly classified into three, i.e., factual abduction (first-order existential abduction), law abduction, and second-order existential abduction. However, since law abduction seems a sub-category of second-order existential abduction, we regard that distinction of factual and law abduction is the primary classification of abduction. The former concerns discovery of facts and the latter concerns discovery of new laws. "Abduction as inversed deduction" is merely one category of factual abduction.

As we mentioned above, the point of this paper is the dynamics of knowledge when formalizing design processes. Factual abduction infers new facts from given facts (observable facts) with fixed theory (rules). However, as long as we use reasoning with a fixed theory, the ability to create new facts is limited. In addition, although factual abduction can satisfy the primary requirement of abduction ("finding axioms from a theory"), this interpretation does not qualify another important feature of abduction mentioned by Peirce. He explained that abduction can find a "surprising" fact. "Inversed deduction" is insufficient to realize such a process and abduction is necessarily accompanied by expansion or revision of knowledge [7]. In the domain such as design in which rich knowledge is available, a feasible expansion of knowledge is obtained by integrating existing knowledge [8]. Integration of knowledge here does not mean a simple addition of knowledge, but rather such operations as translation and modification. There seem to exist a number of possible ways to integrate knowledge. Abduction as a method to integrate knowledge can satisfy the two aspects of creative design, i.e., creating a new product and expanding knowledge. Therefore, we believe that abduction can be one model of creative design processes.

3. Our Approach

In the previous section, we proposed "abduction to integrate theories" as a model of creative design processes, but how can we build this model? Abduction to integrate theories is not of course factual abduction, but it is not just a category of law abduction, either. Abduction to integrate theories in fact concurrently performs both abduction at the factual level and one at the law level. The result of abduction integration of theories performs factual abduction that triggers and directs law abduction.

In our context, factual abduction is to create a new design while law abduction is to create a new theory. Factual abduction is relatively easy to model because "inversed deduction" can play am important role in its reasoning model. We have already discussed and modeled in the previous work [1]. However, Schurz's list of law abduction does not seem complete and law abduction still needs investigation. Moreover, further investigation is needed about the relationship between factual abduction and law abduction.

In this research, our goal is to build a software environment or workbench called "Universal Abduction Studio" (UAS) in which we can test various combinations of abductive reasoning methods and apply them to design knowledge. This will allow us to explore theoretical and practical possibility of design by abduction[1]. In the following sections, we will illustrate UAS in greater details and describe how abduction for integrating theories is implemented.

4. Universal Abduction Studio (UAS)

In this section, we introduce the concept of UAS that is a new style for supporting designers in the conceptual design stage.

4.1 The basic concept of UAS

UAS is a computer environment to support integration of theories (that contain knowledge) from various knowledge domains for creative design. UAS is not a design automation system but a cooperation system that can solve design problems by helping dynamic interaction between a designer and the system. UAS provides a toolbox consisting of a variety of domain knowledge as well as a variety of abductive reasoning mechanisms for knowledge integration. When the designer cannot solve a design problem with knowledge of one domain, the designer chooses a knowledge operation to make correspondences between that domain knowledge and another domain knowledge which UAS proposes. Then, the designer estimates and judges whether or not the proposed knowledge should be used. Finally, the designer generates design solutions based on the tentative design knowledge chosen by her/him.

Figure 1 shows the fundamental concept of UAS. In Figure 1, the designer operates design information and knowledge on the work space. The knowledge integration module consists of multiple abductive reasoning mechanisms, and the designer chooses one or some of them depending on each design problem. The knowledge base consists of multiple domain knowledge bases and the designer first chooses one to solve a design problem. When the designer cannot solve the design problem, the system reasons about another domain knowledge base that can possibly be integrated with the first domain knowledge. The abductive reasoning system then performs knowledge integration. This fundamental concept requires unified knowledge description among various domain knowledge bases.

---

[1] We discuss how knowledge should be structured for abduction from the theoretical aspect in a companion paper [9].

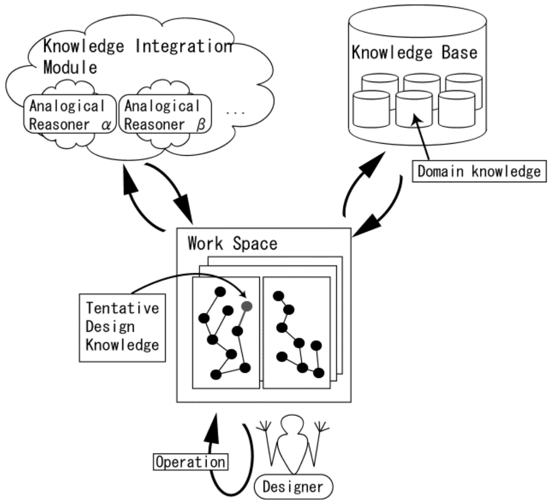

Figure 1. Fundamental concept of the Universal Abduction Studio

4.2 The UAS architecture as an abductive inference system

UAS as an inference system has two specific features. One is the two-layered architecture and the other is combination of multiple inference engines. The former means that there are object- and meta- levels in reasoning. In the object level, the main inference for design is factual abduction as well as some other inferences to support factual abduction. UAS inherits this feature from our previous system [1]. The meta-level is the level in which integration of knowledge takes place. The main inference in this level is law abduction that should be realized by various inference methods. As we mentioned, law abduction has several methods. Schurz lists (pure) law abduction, micro-part-abduction, analogical abduction, missing-link common cause abduction, and fundamental common cause abduction as categories of law abduction, and the list does not seem complete. Considering the variety of law abduction and the current state of investigation for abduction, we do not provide a single inference mechanism for law abduction, but rather multiple inference mechanisms that can be used for abduction.

4.3 UAS Prototype 1

In this paper, among the variety of law abduction, we focus on analogical abduction. The current implementation called UAS Prototype 1 is built to explore how analogical inference works as abduction or a part of abduction. Analogy is used here to discover relationships between theories. Identifying relationships between theories is the first step to integrate theories but usually concepts included in theories look irrelevant superficially. Analogy is expected to find such relationships by examining structural similarity. The system supports the designer to integrate multiple domain theories by following procedures shown in Figure 2 when s/he cannot generate a design solution with a single domain theory.

(1) Selection of a similar theory

Before analogical reasoning for knowledge integration is performed, a similar theory to the current object knowledge must be chosen. One strategy here is that the system first selects a candidate theory that has high similarity and applicability to the current one. However, this candidate will not be interesting enough, because the integration result might be too similar to the original knowledge. Another strategy is to choose a theory that has low similarity with the current one and the integrated knowledge might have low applicability for the design problem. However, it might be possible that unexpectedly such knowledge yields a surprising design.

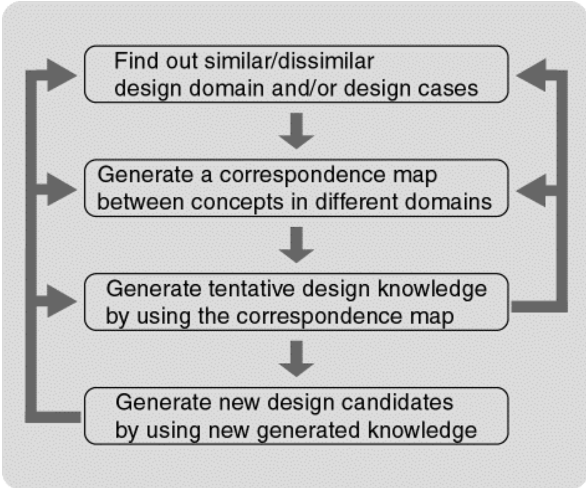

Figure 2. Design procedures in UAS

(2) Formation of correspondences among concepts in different theories

To drive analogical reasoning, it is required to make correspondences among concepts in the theories chosen in Step (1). We can automatically form such correspondences among those concepts with some methods described in the next section. Alternatively, the designer can make these correspondences manually.

(3) Generation of candidate design knowledge by correspondence

Through Steps (1) and (2), the system generates candidate design knowledge as a combination of theories as well as the correspondences, with which the designer can solve the current design problem. At this time, the candidate knowledge is still just a hypothesis, so it is necessary to distinguish this knowledge clearly from the original theories. If the candidate knowledge turned out to be inappropriate to solve the design problem, the designer should go back to Step (1) and choose a new domain theory again.

(4) Creation of candidate design solutions

The designer generates candidate design solutions with the candidate knowledge obtained in Step (3). The candidate design solutions generated in this step verify the candidate design knowledge and determine if it can be adopted as new knowledge. Namely, we estimate a candidate design solution from the candidate knowledge, which is a process of hypothesis verification, to accept the candidate design knowledge as new knowledge. If the candidate design solution can be accepted as a design solution, the designer finishes the current problem solving and starts the next design step. Otherwise, the designer repeats Steps (1) to (3) again.

4.4 Algorithms of UAS

UAS supports knowledge integration by proposing an appropriate mechanism for analogical reasoning to design problem. Before knowledge integration, correspondences among concepts in various theories should be discovered. Therefore, we need a unified knowledge description format across various domains with richness in expression. For these reasons, we adopt first-ordered predicate logic as unified information description format in UAS. Predicate logic has terms and predicates as its fundamental components. The term corresponds to a subject or an object that appears in a sentence and the predicate corresponds to a verb or an adjective.

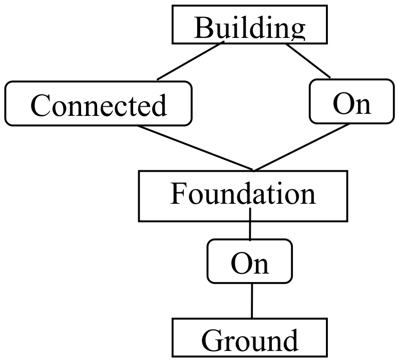

Figure 3. Graphic representation of knowledge

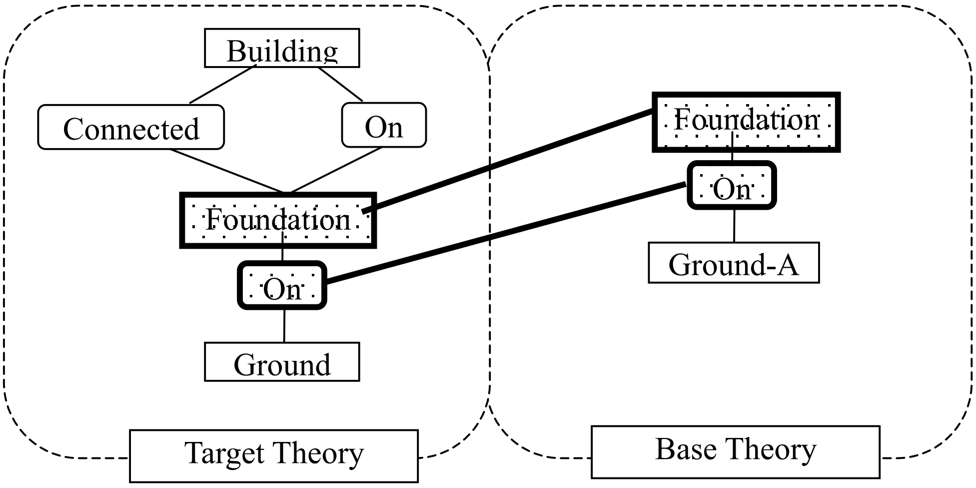

Figure 4. Graph matching

Knowledge integration in UAS consists of the following two processes. First, UAS identifies correspondences in different theories with analogical reasoning. It is thought that analogy is one of influential mechanisms that support flexible thinking of a designer. Based on similarity of domains, analogy links an unknown idea in the target domain into the base domain. In creative design, it is assumed that this analogy process is equal to linking a theory to another based on similarity among knowledge. Here, a criterion of similarity among different concepts plays an important role. Since there can be different criteria of similarity for analogy in general, we can effectively make correspondences based on combinations of various styles of analogy.

In this research, we use the graph matching technique for predicate logic to find correspondences. A theory that is described in predicate logic format can be represented as a graph structure shown in Figure 3 which depicts factual knowledge about foundation structure of a building. Figure 4 depicts an example of graph matching between two theories, in which generate correspondence relationships with sub-graph matching.

As a result of this process, UAS makes possible maps of concept correspondences among different theories. However, usually the result will likely be more than one map generated. So, the designer has to choose a suitable map to her/his design problem with help of a score for each map. The designer chooses one element, for example a bottleneck element of the current problem, as a starting point. Then, the system orders the obtained correspondence maps as follows:

1. The predicate of the chosen element is scored 1.
2. A proposition that shares the same predicate, of which score is 1, is scored 2.

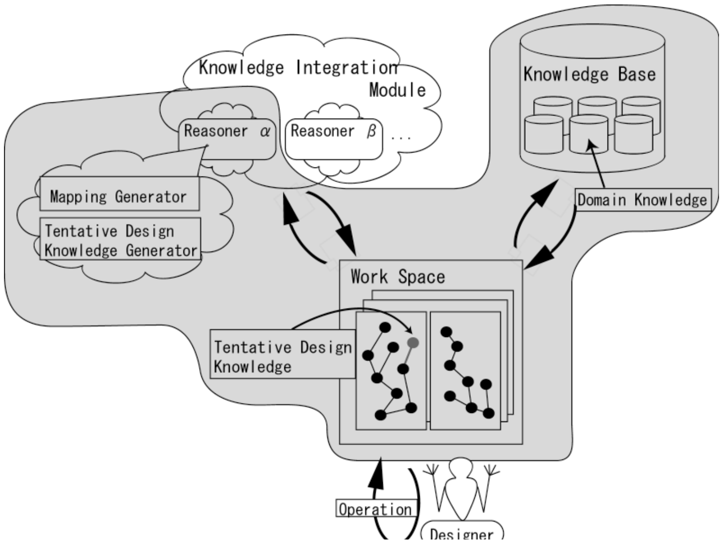

Figure 5. Fundamental architecture of UAS Prototype 1

3. Any proposition that shares the same predicate, of which score is n, obtains n+1.
4. Calculate the score for each predicate by adding the scores of the propositions that includes the predicate.
5. Obtain the total score of a sub-graph by adding all the points of the included predicates.
6. Find scores of all sub-graphs and order the sub-graphs accordingly.

However, notice that the designer should be free to choose from not only highly scored maps but also any candidate maps.

The second process of knowledge integration is to merge a base theory with a target theory to create candidate design knowledge (Figure 4). The target theory is chosen by the designer based on the scores. Figure 4 shows a matching between two sub-graphs and as a whole these two represent candidate design knowledge which states such knowledge as "Not only 'Ground' but also 'Ground-A' might be suitable for a foundation structure of a building" and "Ground-A might be similar concept to Ground."

5. Implementation

Based on the architecture and algorithms explained in the previous chapters, we developed a system of UAS Prototype 1. The system is coded with JAVA programming language on the Windows 2000. Figure 5 depicts the system architecture of the prototype UAS (shadowed part). In Figure 5, the work space provides the designer with an interface to the system, and the knowledge integration module consists of the correspondence generator and the candidate design knowledge generator. The knowledge base is a collection of domain theories.

The prototype system has the following functions.

1. Maintenance of domain theories in the knowledge base.
2. Generating correspondences among different theories.
3. Proposal of candidate design knowledge by the analogy mechanism.
4. Modification and addition of domain theories based on the result of knowledge verification.

6. An Example

As an example, let us consider the design of a wheel cap. We use design procedure documents that were actually created during the modeling process of a product on a 3D CAD by a car-component supplier. The documents are composed of design procedure documents about subcomponents of the wheel cap, such as “guiding rib” and “wheel cap nail.” Because the design of these subcomponents is independent from each other, these design procedure documents about each subcomponent form a domain theory.

We first compiled a vocabulary as a collection of terms used in the design of a wheel cap, and then converted knowledge of design procedure documents into first-order predicate logic. The vocabulary includes the following concepts.

1. Entity, such as “guiding rib” and “wheel cap nail”
2. Relation, such as “on” and “connected”
3. Attribute, such as “radius” and “rigidity”
4. Process, such as “making” and “considered”

An entity is represented as a term and a relation, attribute, and process as a predicate in predicate logic. The prototype system represents domain theories as graphs (Figure 6). In the graph, a rectangular node with black background represents an entity (term), and ones with other background colors represent predicates, i.e., relation, attribute, and process in this example.

We carried out an experiment to demonstrate the system’s performance of knowledge integration for creative design as follows.

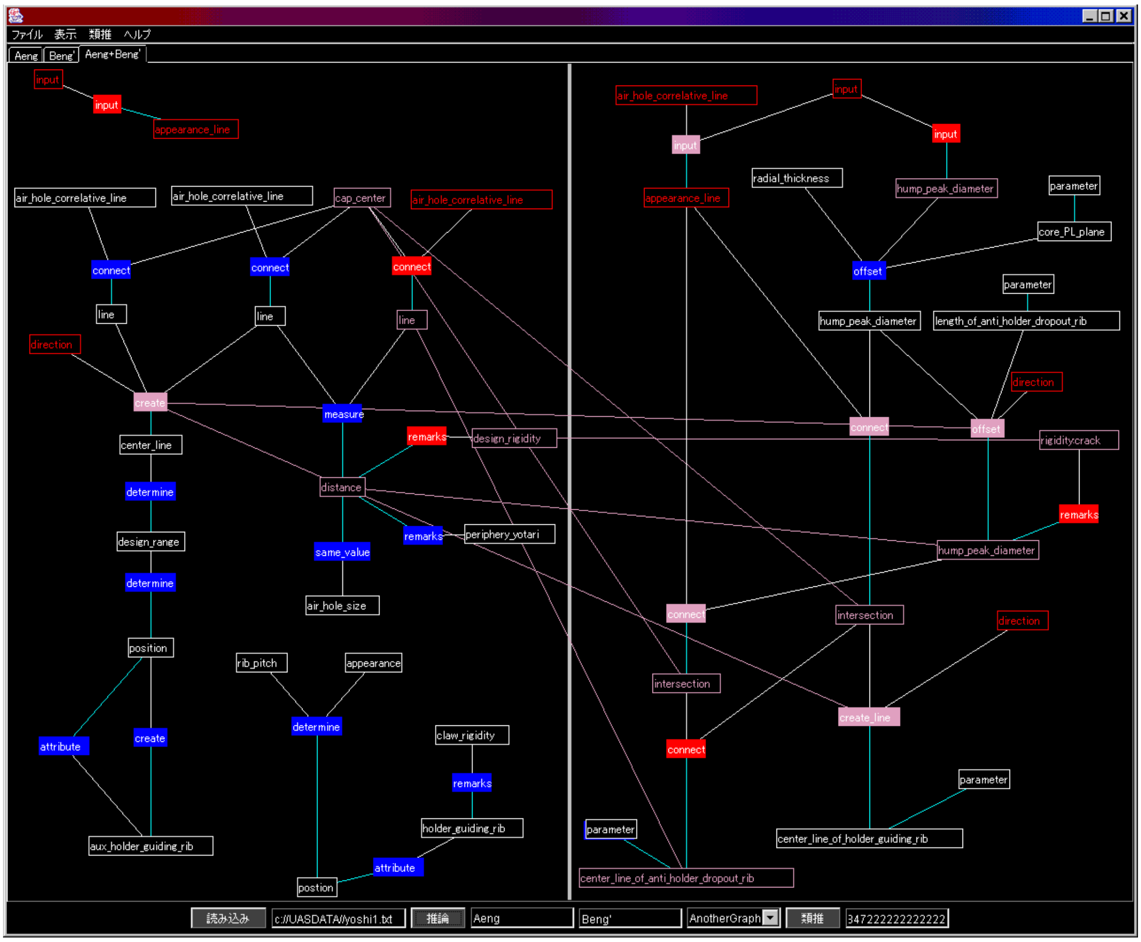

Figure 6. A screen image of the prototype system

1. The designer put in a design process document about “supplementary guiding rib” (the left side of Figure 6). The domain theory in the form of design document is called *base knowledge*.

2. The designer selected another design process document about “guiding rib” (the right side of Figure 6) in order to integrate with the base knowledge. The domain theory selected at this step is called *target knowledge*.

3. The system carries out sub-graph matching between them and discovers correspondences between the base knowledge and the target knowledge. These matched concepts are highlighted by the nodes with red background in Figure 6.

4. Based on the correspondences, the designer generates candidate design knowledge. The system highlights neighbor nodes of the matched sub-graphs (nodes with pink background) and links them with each other, whereby the system suggests that they might have analogical relationships. In this example, the system suggests that there is a relationship between “rigidity of appearance” in the base knowledge and “rigidity for split” in the target knowledge and the latter should be “considered” for “hump radius.” There are some possible ways to interpret the results. For example, the designer can interpret that not just “rigidity for split” but also “rigidity of appearance” should be considered for “hump radius.” The designer can also interpret that “rigidity for split” is a similar concepts with “rigidity of appearance.”

5. The designer integrates the base knowledge and the target knowledge based on the evaluation of this tentatively integrated knowledge (candidate design knowledge). The designer should evaluate the interpretation of the candidate design knowledge. If the designer validates the interpretation for solving design problem, the candidate design knowledge is accepted thereby the knowledge integration is done. In this example, the designer accepted that not just “rigidity for split” but also “rigidity of appearance” should be considered for “hump radius” to carry out the design of a hump.

The result of this procedure is a revised design procedure as well as the tentatively integrated knowledge, i.e., candidate design knowledge. The former is an addition (*rigidity of appearance” should be considered for “hump radius”*) to the current design procedure and this revision is supported by the latter, i.e., knowledge integrated by the discovered mappings like one from “rigidity of appearance” to “rigidity for split”. This mapping is one of the hypotheses generated by analogical abduction, and the knowledge integrated by such mappings is expected to lead to a better wheel cap that considers the aesthetic point of view. It should be noted that both results are still tentative until the total design solution according to this procedure is justified as a design product.

7. Conclusion

This paper described a new design support system that aims to support conceptual or creative design by dynamically integrating various theories in different design domains. We argued that abduction for integrating theories can be a basic principle to formalize such design processes. According to this principle, we proposed Universal Abduction Studio (UAS) and illustrated its architecture, algorithms, and prototype. This system should provide various types of abduction that designers can select and combine to archive their design. We also discussed technologies to realize UAS. We proposed to use analogical reasoning as one of key techniques to realize abduction for design. We showed in the example taken from a real design activity that the system can discover a new idea that can be used in new design.

In this paper, we showed the basic concept of UAS and its prototype, UAS Prototype 1, which is based on a single type of inference. We plan to extend it from the theoretical viewpoint by clarifying relationships among various types of abduction as well as to develop more advanced prototype systems by introducing other types of inferences.

References

[1] Takeda, H., Yoshioka, M., and Tomiyama, T., "A general framework for modeling of synthesis – integration of theories of synthesis –", In 13th International Conference on Engineering Design - ICED 01, Design Research - Theories, Methodologies, and Product Modeling, pages 307-314, Glasgow, August 2001.

[2] Coyne, R., Logic Models of Design. Pitman: London, 1998.

[3] Roozenburg, N.F.M., and Eekels, J., Product Design: Fundamentals and Methods. Chichester, MA.: John Wiley & Sons, 1995

[4] Hartshorne C. and Weiss P. (eds.), The Collected Papers of Charles Sanders Peirce, Vol. I-VI, Harvard University Press, Cambridge, MA., 1931-1935.

[5] Flach, P. And Kakas, A. (eds). Abductive and Inductive Reasoning: Essays on their Relation and Integration. Kluwer Academic Press, "Applied Logic Series". 2000.

[6] Schurz G., "Models of Abductive Reasoning", TPD Preprints Annual 2002 No. 1, Schurz G. and Werning M. (eds.), Philosophical Prepublication Series of the Chair of Theoretical Philosophy at the University of Düsseldorf, 2002. (http://service.phil-fak.uni-duesseldorf.de/ezpublish/index.php/article/articleview/70/1/14/). To appear in Synthese (Kluwer Academic Publishers, Dordrecht) in 2003.

[7] Aliseda, A., "Abduction as Epistemic Change: A Peircean Model in Artificial Intelligence", in [6].

[8] Takeda H, "Abduction for design", In J.S. Gero and E. Tyugu, editors, Formal Design Method for CAD, IFIP Transactions B-18, pages 221-244. Elsevier Science Publishers B.V., Amsterdam, 1994.

[9] Tomiyama T., "Structuring Design Knowledge for Better Design Synthesis," Proceedings of ICED '03, this volume, 2003

For more information please contact:

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