ICED13
INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

# PROMOTING VOID-BASED DESIGN CONCEPT GENERATION THROUGH COMPUTER-SUPPORTED INTERACTIVE STRUCTURIZATION OF VERBAL AND DRAWING EXPRESSION

**Yutaka NOMAGUCHI, Tatsuya OGAWA, Kikuo FUJITA**
Osaka University, Japan

## ABSTRACT

An iterative process of externalizing ideas in verbal and drawing expression is a crucial activity for design concept generation. This paper proposes its systematic methodology focusing on ‘void’ of concepts. The proposed methodology supports classification of expressed ideas according with their related concepts, structurization and finding unnoticed combination of concepts, which this research calls ‘void’. Those steps promote a designer to reflect their own ideas within designers’ mind so as to notice potential of new design solutions. This paper also proposes its implementation plot. It incorporates a concept network model, which integrates verbal and drawing expression, and manages a draw layer that is a unit of combination and associates it with a verbally-expressed concept. The tool provides user-friendly interface to cut off a part of drawn sketch, and associate it to concepts. Behind those back-and-forth sketching process, the tool captures alternatives of sub functions’ solutions and automatically generate a concept matrix so as to clarify voids. A design example of a wheelchair is demonstrated in order to show the effectiveness of the proposed method.

*Keywords: design concept generation, sketch, creativity*

Contact:
Dr. Yutaka Nomaguchi
Osaka University
Department of Mechanical Engineering
Suita
565-0871
Japan
noma@mech.eng.osaka-u.ac.jp

# 1 INTRODUCTION

Sketching is a quick way of expressing and navigating ideas, which are initially ill-defined and gradually concretized, in conceptual design stages. Many researchers have focused on the role of sketching activity in design concept generation. Researches of design protocol analysis have revealed that concept generation process is carried out by reflective interaction between verbal and drawing expression (e.g., Yang and Cham, 2007). Designers notice concepts implied by drawing expression, clarify those by verbal expression, and then develop ideas through the interaction. Prior works of systematic design methodology, e.g., Pahl and Beitz (P&B) approach (Pahl and Beitz, 1988), have stated that classification of externalized design concepts facilitates designers to find unnoticed combination of concepts, called *void* (Tomiyama *et al*., 2010), and promote to generate new design ideas. Although these methodologies mainly focus on verbal expression and rarely on the role drawing expression, they give an important notion toward any systematic way of design concept generation through the interaction of verbal and drawing expression.

The aim of this research is to develop a computational support method for promoting void-based design concept generation through the interaction of verbal and drawing expression. In this paper, after the concept of void is clarified under the viewpoint of General Design Theory (GDT) (Yoshikawa 1981), a framework is proposed for supporting a designer to classify expressed ideas according with their related concepts, find voids and notice potentiality of new design solutions. This paper also proposes its implementation methodology of a support system. It incorporates a concept network model, which integrates verbally-expressed classification system and idea drawings, and manages association among them. A system provides user-friendly interface to externalize and modify the expressions. Behind those back-and-forth sketching process, a system captures alternatives solution and automatically generate a classification table so as to clarify voids. A design example of a manual wheelchair is demonstrated in order to show the effectiveness of the proposed method.

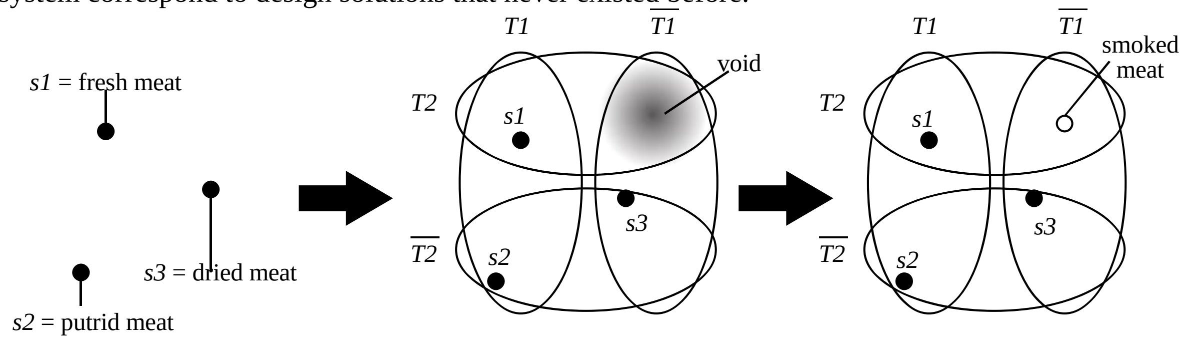
# 2 THEORETICAL BACKGROUND OF DESIGN CONCEPT GENERATION

## 2.1 Void

Design engineering community has addressed the formalization of the design concept generation process, and proposed methodologies supporting it in a computable, structured, and rigorous, not adhoc manner (Antonsson and Cagan, 2010). In its systematic approach, a classification system of generated concepts takes an important role to help designers think logically and create new concepts. A structurized table format is often used for classification. The generated solutions are classified depending on any similarities by those table formats. This will result in classification system which contains only generated solutions, but potentially there can be many *voids*, that is, empty cells in the table. Making an incomplete table complete is a process that helps designers to arrive at creative design solutions.

Tomiyama focused on the void, and give theoretical foundation to the void-based concept generation process (Tomiyama, *et al.*, 2010) using GDT, which models design process as design knowledge operations, i.e., set operational processes regarding the entity set and its subsets. The following anecdote clarifies the role of void. Let us consider a primitive world, where only three kinds of meat are found, fresh meet, putrid meet, and dried meet; all natural entities, without any artificial processing. As shown in Figure 1-(a), people of the world recognized and memorized them as natural entities; $s_1$ = fresh meat, $s_2$ = putrid meat, $s_3$ = dried meat. People in the world construct concepts about peculiar characteristics abstracted from these entities, giving classifications from the viewpoint of function or value of the entity. For example, let $T = \{T_1, T_2\}$ be a classification: $T_1$ = {deteriorates with lapse of time}, $T_2$ = {eatable} as shown in Figure 1-(b). It is assumed that people had the ability of logical operations such as disjunction, conjunction, or complement besides memory, abstraction and classification. Someday, a person in the primitive world thought accidentally about; $\overline{T_1} \cap T_2 = \emptyset$. This combination leads to void. This had no correspondence to a real entity but the value of it was higher than any existing entity. “Eatable and not deteriorate with lapse of time” had the highest value, and this conceptual combination was the necessary condition to invent the smoked meat which had been the first artificial entity for the human beings (Figure 1-(c)).

The above simple anecdote implies that it is critical to have an image about specifications that have no immediate solutions. This is a necessary condition for creative design. Voids in the classification system correspond to design solutions that never existed before.




(a) Perception of entities (b) Classification, Finding of void (c) Creating new concept

*Figure 1 Void-based Concept Generation Process*

## 2.2 Interaction between Verbal and Drawing Expression

When reflecting on the nature of thinking and the classification system, most design theories including the systematic approaches and GDT associate it primary with words or with language (Schön and Wiggins, 1992). Design concept can be theoretically represented by verbal expression. In actual design process, however, many of design concepts cannot be perceived with words but with just images at the beginning. Drawing expression is used instead of verbal expression in order to represent such image of concepts. A designer finds unnoticed design concepts by seeing drawing expression, and revises the classification system within the designer's mind. Figure 2 illustrates concept generation process of a water tank of a coffee maker. At the beginning, a design solution of a water tank is represented by a few concepts such as "storing water" and "easy-to pour water." A designer draws a sketch, sees its details, and finds unnoticed concepts; such as a tank should "prevent spilling water" or have "rounded shape" for its good appearance. As design process continues, designers acquire knowledge, and the number of concepts characterizing design solution increases.

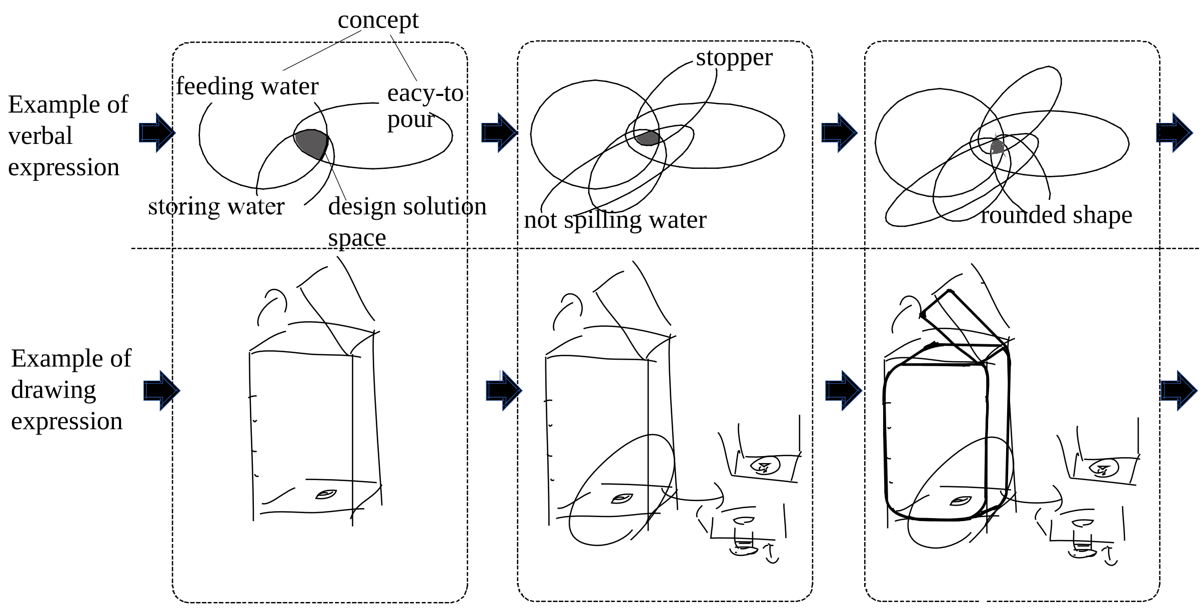



*Figure 2 Progress of Design Concept Generation with Sketching*

Visual thinking is a familiar viewpoint of sketching activity focusing on concept generation through the interaction of designer's thought and expression (Shar, et al., 2011). It is asserted that reflection-in-action (Schön, 1983) is a key paradigm to understand cognitive behaviors of designers in visual thinking (Gero and Kannengiesser, 2008, Tang, *et al.* 2011). Woolsey formalized process of visual thinking as iterative cycle of three phases, i.e. act, reflect and change (Woolsey, 1996). Act is a phase in which designers externalize design concepts by verbal and drawing expression. Reflect is a phase in which designers see, analyze and evaluate the expressed concepts such that designers not only can communicate each other, but also can notice from the expression by backtalk. Reflection is often carried out by comparing multiple alternatives of design solution, or by comparing with the past design cases. Void would be useful for the reflection step. Change is a phase in which designers refine the design solution based on the results of refection in his/her mind by generating design concepts. As a consequent of change step, designers externalize concepts again. It would backtrack to a former design solution that was formerly rejected.

## 2.3 Requirements for Computer Support of Design Concept Generation

According with the viewpoint of visual thinking process and the notion of void, a computer-support tool for design concept generation is required to support the three phases of visual thinking;

1. For act phase, a support tool should have a mechanism by which designers can easily represent both verbally expressed concepts, which used for representing classification system, and drawing expression of design solution ideas.
2. For reflect phase, a support tool should have a mechanism of structurization of multiple design solution ideas based on the classification system, and support to find voids.
3. For change phase, a support tool should have a mechanism to flexibly change design concepts.

Many sketching tools have even been developed in various domains for supporting design concept generation, and they have tackled those research issues. For instance, there are tools that facilitates designers to draw 3D concept sketches (e.g., Kara and Shimada, 2008), tools that have user-friendly interface of drawing and annotating text (e.g., Zurita *et al.*, 2008), tools that can manage alternatives and argumentation of sketches (e.g., Demian and Fruchter, 2009), tools that are integrated with a classification table (e.g., Lo *et al.*, 2010), tools that record history of drawing and facilitating backtrack (e.g., Yamamoto et al., 2006) and so on. However, few tools have comprehensively solved those four requirements.

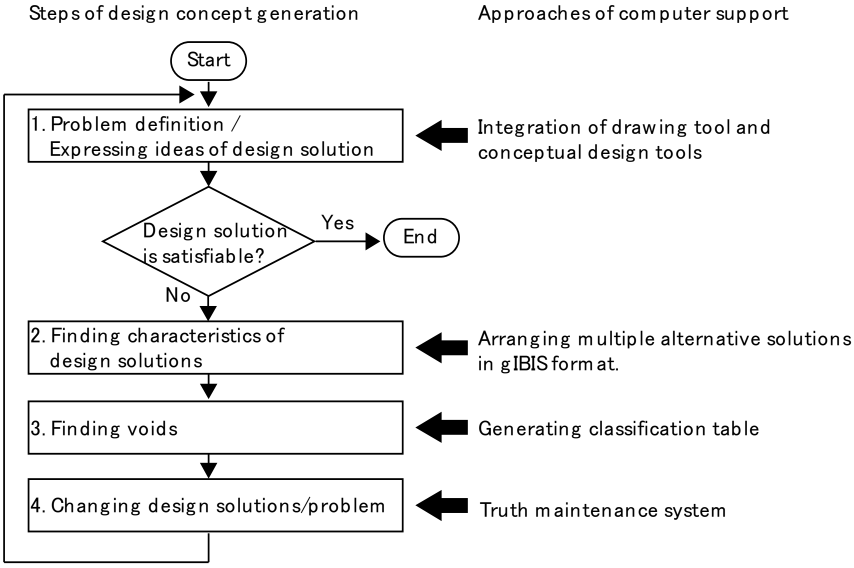



*Figure 3. Steps of Design Concept Generation and Approaches of Computer Support*

# 3 VOID-BASED DESIGN CONCEPT GENERATION PROCESS

This research models a design concept generation process that consists of the four steps considering issues stated in Subsection 2.4. The left side of Figure 3 shows the proposed process.

The first step corresponds to act phase. A designer describes a problem that he/she is going to tackle and expresses rough ideas of design solutions. Unless a designer can find a satisfiable design solution, the process proceeds to the following steps. The second and the third step correspond to reflect phase. A designer gives a characteristic of expressed design solution using classification system, and finds voids. The forth step corresponds to change phase. A designer modifies design solutions or problem definition if necessary, and generates new design solution ideas.

The right side of Figure 3 shows our approaches for computational support for each step. For the first step, an integrated environment of a drawing tool for drawing expression and conceptual design tools for verbal express is provided. For the second step, a mechanism to arrange multiple alternative solutions is provided. This research uses gIBIS (graphical Issue Based Information System) (Conklin and Begeman, 1988), a well-known argumentation model, to visually represent multiple alternatives and their logical relationships. For the third step, a mechanism to generate a classification table that structures drawn ideas is provided. For the fourth step, a truth maintenance system that manages multiple alternatives in a single context is used.

# 4 METHODOLOGY OF COMPUTER SUPPORT

This research adopts a framework of DRIFT (Nomaguchi and Fujita, 2007) to realize the approached noted in Section 3. The following subsections explain its details.

## 4.1 Integration of Drawing and Verbal Expression

A support system has a front end of a drawing tool and a concept tree tool, such as a function-structure map, which represents design concepts in a verbal expression (Figure 4-(1)). The framework employs a concept network model (Figure 4-(2)) as a generic form of design representation. The concept network model associates a drawing expression with each design concept represented in verbal expression.

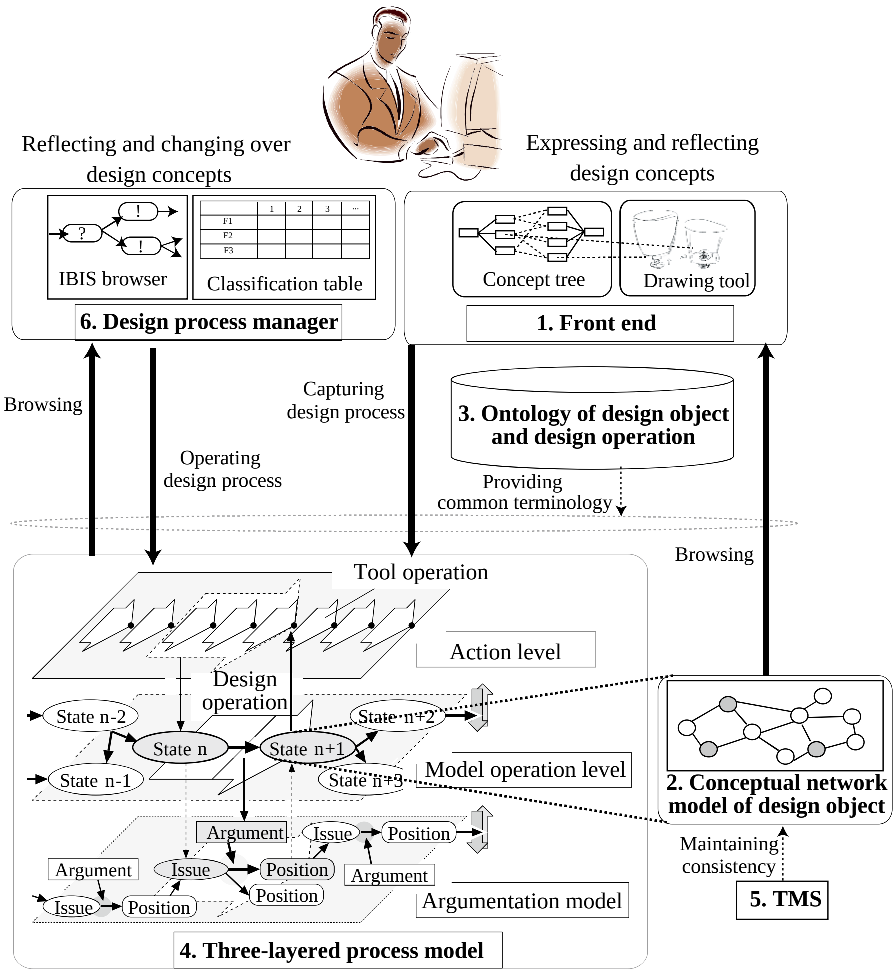



*Figure 4. Overview of Design Concept Generation Tool based-on DRIFT*

Common terminology for integrating tools is defined as an ontology (Figure 4-(3)). *Figure 5* shows classes of node of the concept network that is related to a sketch tool in UML format. Our prior work defined an ontology of node classes representing verbal expression of design concepts, i.e., function, entity and attribute. Relationships between the concepts and hierarchy were also already defined. This subsection briefly explains node classes of drawing expression, i.e. sketch, draw layer, draw line and sketch derivation relation. A draw line is a primitive element of drawing expression. Its node has information of coordinates of control points composing the line, and information of line appearance such as line color, line type and line width. A draw line belongs to only one draw layer (*Figure 5*-(1)). A sketch is composed of multiple draw layers (*Figure 5*-(2)). A draw layer can be included in different sketches. A sketch is associated with entities, functions or attributes that characterize the design solution drawn in the sketch (*Figure 5*-(3)). The layer-based mechanism facilitates to flexibly superpose draw layers and modify drawing (Nomaguchi, *et al.*, 2011).

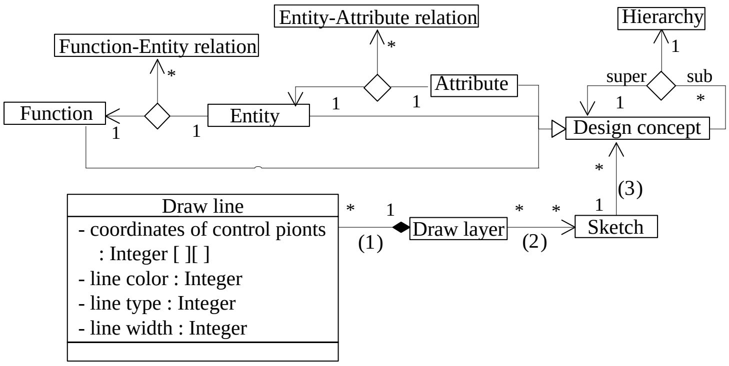



*Figure 5. Node Classes and their Relation*

### 4.2 Management of Multiple Alternative Solutions

The core of this framework is the three-layered model of action, model operation, and argumentation, which is shown in Figure 4-(4). The action level represents a sequence of tool operations, such as generating node and drawing a line. The model operation level represents a sequence of design operations, such as developing a function, relating attributes to an entity, drawing a sketch of an entity, and so on. It records the design state transition by capturing a snapshot of the design state before and after performing the design operation. The argumentation level represents a logical structure of issue and its alternative positions with a form of gIBIS.

In the three-layered process model, a design operation is defined as a unit design activity. Because the design process is a sequence of tool operations, an ontology of a design operation defines sequence of tool operations corresponding to a unit design activity. A definition of a design operation also gives a format of issue-position pair. Those features facilitate to convert a sequence of tool operations to a sequence of design operations, and further connect to argumentation structure. The framework realizes a recording of the design process as a byproduct of the natural design activities that is performed on conceptual design tools and a drawing interface.

Our past research (Nomaguchi and Fujita, 2007) has already defined some design operations performed on verbal expression of design concept, such as developing function, developing entity and setting relation between a function to entities. This research adopts those, and defines new design operations performed on drawing expression i.e., (1) drawing a sketch about a function, (2) drawing a sketch about an entity, (3) associating draw layers to a sketch, and (4) associating draw lines to a draw layer. Figure 6 shows a definition of "associating draw layers to a sketch." This definition illustrates that the design operation consists of tool operations of creating multiple draw layers and linking them to a sketch.

```
Operation name: Associating draw layers to a sketch
Referred node: Sketch s;
Added node: Draw_layer l[];
Sequence of tool operations:
  for (int i=0; i<n; i++){
      create(l[i]);
      link_to(l[i], s);
  }
Issue text: print("What layers compose "+ s+"?");
Position text: for(int i=0; i<l[].size; i++) print(l[i].name);
```

*Figure 6. Example of Design Operation Definition*

### 4.3 Structurization of Design Solutions with Classification Table

A classification table is implemented on a mechanism of design process manager (Figure 4-(6)). A classification table refers an alternative sketches related to design concepts of entity, function or attribute, which are captured by the three-layered model, and organizes that information into a table format. Note that this research is based on an assumption that design concepts are verbally represented in hierarchy, and drawing expression is associated to concepts. The concept hierarchy and the association would be incomplete in design process as stated in Subsection 2.3 because of the lack of knowledge. A designer updates it through the design process as well as design solutions. A classification table is generated by the following sequence.

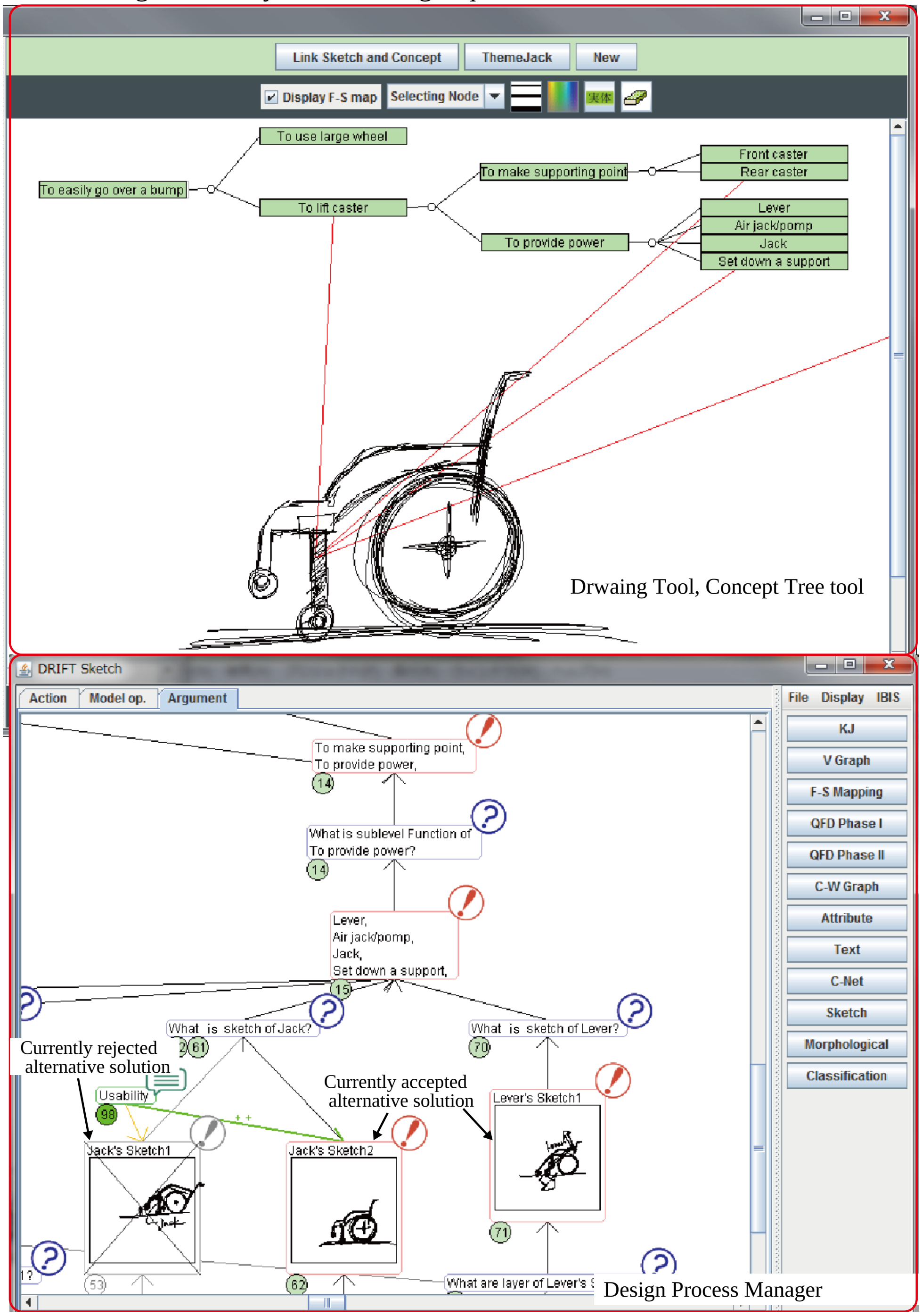



*Figure 7. Snapshot of Wheelchair Design Performed on Prototype System*

1. A user designates design concepts that are used as axes of classification. In the anecdote of Subsection 2.3, “eatable” and “deteriorative” can be axes.
2. A support system gets sub concepts of the axis concepts designated in the Step 1 using the concept hierarchy. For example, “eatable/not eatable”, “deteriorative/not deteriorative” can be considered as sub concepts in the primitive-world anecdote.
3. A support system gets sketches of design solutions that are associated with the sub concepts, and arrange them in a table format that are consisted with the axis concepts.
4. A user find empty spaces in the table, that is, voids. A system suggests a user to try to create a design solution corresponding to the void.

## 4.4 Flexible Change of Alternative Solution

The framework manages multiple design states over the single concept network model using a truth maintenance system (TMS) (Doyle, 1979) (Figure 4-(5)). Its algorithm maintains what concept is valid or invalid, and updates the valid/invalid status at every moment when a new concept is added. Thanks to this mechanism, a designer can change alternative design concepts and backtrack to earlier design states by one-click operation on an IBIS browser (Figure 4-(6)).

# 5 SUPPORT TOOL IMPLEMENTATION

## 5.1 Prototype System

A prototype system is implemented by object-oriented programing language Java (SDK 6.0) on Windows 7. WACOM LCD tablet DTU-710 is used in order to input drawing expression by pen gestures. Figure 7 shows a snapshot of a wheelchair design process performed on the prototype system. An upper window is a drawing tool incorporated with a function-structure mapping tool. A designer can represent design solution ideas by drawing and verbal expression and associate them. A lower window is a design process manager. It automatically captures and organizes expression of alternative ideas in a gIBIS format. A designer can change a current alternative solution by one-click operation. Figure 7 shows that two ideas using a jack are drawn. “Jack’s sketch 2” displayed by a red node is currently accepted, and “Jack’s sketch 1” displayed by a gray hatched node is currently rejected. A designer can compare those alternatives, and can write the contents of the reflection using an argument node displayed by a green node. An argument node in Figure 7 shows that the current alternative solution of “Jack’s sketch 2” is accepted because of it better usability.

## 5.2 Design Example of Wheel Chair

This subsection demonstrates concept generation process of manual wheelchair design performed on the prototype system.

### *5.2.1 Problem Description*

One of issues of manual wheelchairs is to go over a small bump in a pavement or in a house. As for a usual wheelchair which caster’s diameter is about 130 mm, it is difficult for a wheelchair user to go over a 20 mm-height bump without a helper’s assistance. The bigger caster, the easier to go over a bump. However, a bigger caster has less usability. This design example aims to propose a new manual wheelchair that satisfies both of functions; going over a bump, and good usability.

### *5.2.2 Initial Ideas*

A designer draws initial rough ideas on a drawing tool of the prototype system. *Figure 8* shows its examples, such as an idea of lifting a caster by declining a wheelchair backward, an idea using tri-wheel caster that gives larger turning radius with a smaller caster, and so on. Among those initial ideas, he/she focuses on ideas of lifting a caster by any mechanisms. Some variations of this idea are proposed according with mechanisms of lifting up a caster, such as using a jack, using an air-jack, using a lever, and so on.

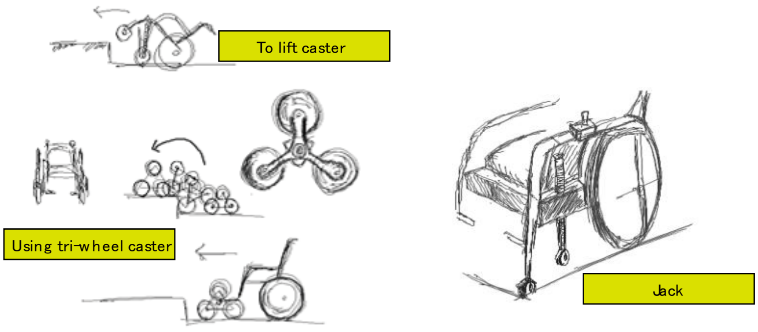



*Figure 8. Initial Sketches*

### *5.2.3 Classification of Ideas*

Those proposed idea have two pairs of caster; one is used for driving, and the other is used for providing supporting point when a wheelchair is lift up. A designer reviews sketches of those multiple design solutions, and noticed that those can be categorized by the position of supporting point; some idea using front-side caster as a supporting point and other uses rear-side caster. This classification was without a thought when those ideas were drawn. Using front-side caster gives a stability of driving, and using rear-side caster give an ability of easily turn in a small radius.

A designer changed a concept tree by adding this classification. According the classification, the system automatically generated a classification table focusing on the two axis concepts; to support a wheel chair, and to provide power to lift up. *Figure 9* shows a generated classification table.

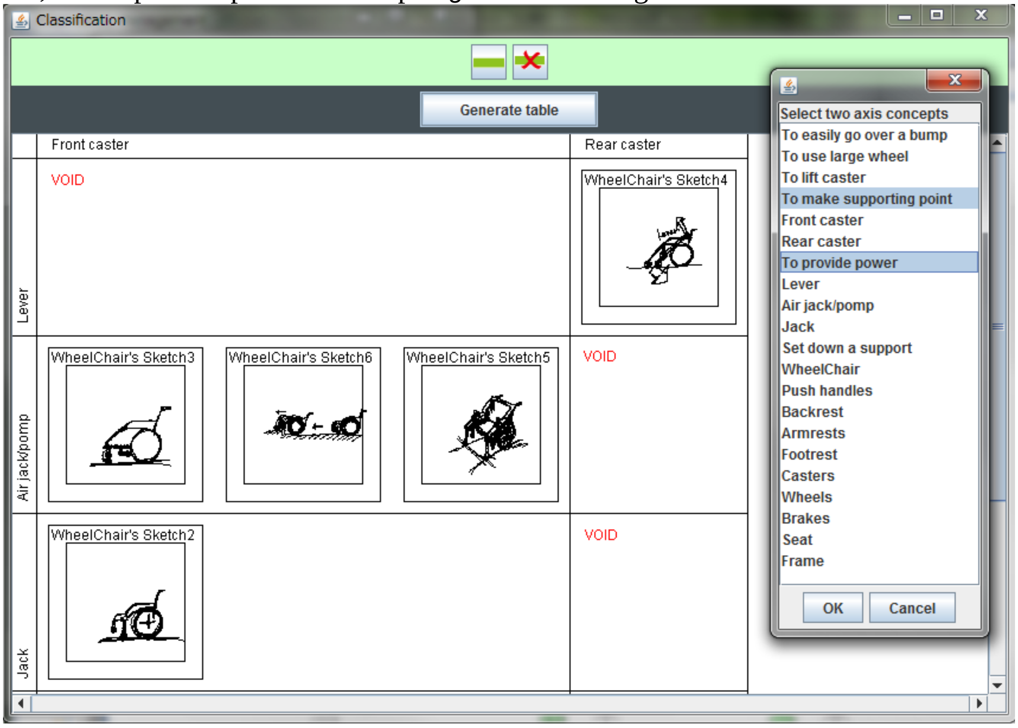



*Figure 9. Classification Table*

### *5.2.4 Finding Voids and Generate New Idea*

A designer found that there are some voids in the classification table, and generates new ideas corresponding to them. Figure 10 shows an example of one of voids, using rear-side caster for supporting point and using air-jack. Those were not drawn at an initial sketch phase. A designer can notice those new ideas thanks to the void-based methodology.

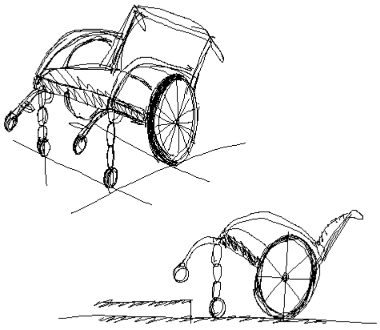


*Figure 10. New Design Solution using Air-jack and Supporting with Rear Caster*

## 6 CONCLUSION

This paper focused on the void-based design concept generation through the interaction of verbal and drawing expression, and its computational support methodology. The proposed methodology supports a designer to classify expressed ideas according with their related concepts, and find voids. Those steps promote a designer to reflect their own ideas within designers' mind so as to notice potential of new design solutions. This paper also proposed its implementation plot. A design example of a wheelchair shows that the implemented system can support void-based design concept generation process, and that a designer can propose new ideas.

Our future works include analysis of the cognitive aspects of void-based concept generation process. This research does not focus on how a designer finds characteristics of drawn design solutions and axes concepts for making a classification table. A theory of ontology or concept dictionary, e.g., WordNet (Fellbaum, 2013), will give a useful notion for supporting this step. After refining a classification system and finding voids, some systematic concept generation approaches such as TRIZ (Altshuller, 1984) will be useful to generate new design solutions ideas. Integrating those technologies into the proposed methodology will help to refine human-computer interaction and encourage designers to make use of the tool.

## REFERENCES


Altshuller, G., (1984) *Creativity as an Exact Science*, NY, Gordon and Breach.

Antonsson, E., and Cagan, J., (2010) *Formal Engineering Design Synthesis*, New York, Cambridge University Press.

Conklin, J., and Begeman, M. L., (1988) gIBIS: A hypertext tool for exploratory policy discussion, *ACM Transactions on Office Information Systems*, Vol. 6, No. 4, pp. 303–331.

Demian, P., and Fruchter, R., (2009) Effective visualisation of design versions: Visual storytelling for design reuse, *Research in Engineering Design*, Vol. 19, No. 4, pp. 193–204.

Doyle, J., (1979) A truth maintenance system, *Artificial Intelligence*, Vol. 12, No. 3, pp. 231–272.

Fellbaum, C. (2013) *WordNet – A Lexical Database for English* [online], http://wordnet.princeton.edu/ (10 January 2013).

Gero, J. S. and Kannengiesser, U. (2008) An ontological account of Donald Schön's reflection in designing, *International Journal of Design Sciences and Technologies*, Vol. 15, No. 2, pp. 77-90.

Kara, L. B., and Shimada, K., (2008) Supporting early styling design of automobiles using sketch-based 3D shape construction, *Computer-Aided Design and Applications*, Vol. 5, No. 6, pp. 867–876.

Lo, C.-H., Tseng, K. C., and Chu, C.-H., (2010) Onestep QFD based 3D morphological charts for concept generation of product variant design, *Expert Systems with Applications*, Vol. 37, No. 11, pp. 7351–7363.

Nomaguchi, Y. and Fujita, K. (2007) Ontology building for design knowledge management systems based on patterns embedded in design-for-x methodologies, *Proceedings of 16th International Conference on Engineering Design (ICED 07)*, Paper No. 442

Nomaguchi, Y., Kotera, Y. and Fujita, K., (2011) A knowledge-based superposing sketch tool for design concept generation through reflection of verbal and drawing expression, *Proceedings of 18th International Conference on Engineering Design (ICED 11)* , Paper No. 405.

Pahl, G. and Beitz, W. (1988) *Engineering Design*, Berlin, Springer.

Schön, D, A. (1983) *The Reflective Practitioner*, New York, Basic Books.

Schön, D. A. and Wiggins, G. (1992) Kinds of seeing and their functions in designing, *Design Studies*, Vol. 13, No. 2, pp. 135-156.

Shah, J. J., Woodward, J. and Smith, S. M., (2011) Applied tests of engineering design skills: Visual thinking characterization, test development and validation, *Proceedings of International Conference on Engineering Design - Impacting Society Through Engineering Design* 7, pp. 127-139.

Tang, H. H., Lee, Y. Y. and Gero, J. S. (2011) Comparing collaborative co-located and distributed design processes in digital and traditional sketching environments: A protocol study using the function-behaviour-structure coding scheme, *Design Studies*, Vol. 32, No. 1, pp. 1-29.

Tomiyama, T., Breedveld, P., and Birkhofer, H., (2010) Teaching creative design by integrating general design theory and the Pahl & Beitz methodology, *Proceedings of the ASME Design Engineering Technical Conference & Computers and Information in Engineering Conference*, Montreal, DETC2010-28444.

Woolsey, K. H. (1996) *VizAbility Handbook*, Boston, PWS Publishing.

Yamamoto, Y., Nakakoji, K., Nishinaka, Y., and Asada, M., (2006) Art019: A time-based sketchbook interface, *Technical Report, KID Laboratory*, RCAST, University of Tokyo, March.

Yang, M. C. and Cham, J. G., (2007) An analysis of sketching skill and its role in early stage engineering design, *Journal of Mechanical Design*, Vol. 129, No. 5, pp. 476-482.

Yoshikawa, H. (1981) General design theory and a CAD system, in Sata, T. and Warman, E. (eds.), *Man-Machine Communication in CAD/CAM*, Amsterdam, North-Holland, pp. 35-58.

Zurita, G., Baloian, N., and Baytelman, F., (2008) A collaborative face-to-face design support system based on sketching and gesturing, *Advanced Engineering Informatics*, Vol. 22, No. 3, pp. 340–349.