

ENVIRONMENT-BASED FORMULATION OF DESIGN PROBLEM

Yong Zeng

Concordia Institute for Information Systems Engineering
Concordia University
1455 de Maisonneuve Blvd. W., Montreal, QC H3G 1M8, Canada
E-mail: zeng@ciise.concordia.ca

This paper proposes a formal definition of design problem using the axiomatic theory of design modeling. Based on the concept of product system and an informal description of design problem, a design problem is formulated into three parts: environment, structural requirements, as well as performance requirements. On the basis of this formulation, both structural requirements and performance requirements are related to product environment, which is divided into natural, built, and human environments. As a result, product environment is identified as the source of product requirements. An environment-based formulation of design problem is thus developed. This formulation implies that both design problem and design solutions are included in a product system. The design governing equation is formulated to capture the interdependence relation between product problem and product descriptions along the design process. An example of rivet setting tool design is used to illustrate the mathematical formulations throughout the paper.

Keywords: *mathematical formulation, axiomatic approach, structure of design problem, source of product requirements, product environment, design governing equation.*

1. Introduction

The understanding and modeling of design activities has been an active research topic since 1960's (Cross, 1984), which aims to enhance the quality of designed product and to improve the efficiency of the design process. As a result of this effort, various design methodologies have been proposed, such as the systematic design methodology (Hubka and Eder, 1988; Pahl and Beitz, 1988), the theory of inventive problem-solving (Altshuller, 1984), axiomatic design (Suh, 1990), decision-based design theory (Allan and Mistree, 1997), general design theory (Yoshikawa, 1981), and formal design theory (Braha and Maimon, 1998). Attempts have been made to establish an across-disciplinary design science so that basic principles can be identified to guide the design of complex systems (Ertas *et al.*, 2000; Maxwell *et al.*, 2002). However, the progress in this field has been far lagged behind its underlying ambition and expectations. No clear answers can be found in the literature for even the most fundamental question of this exploration: what is a design problem? An interesting phenomenon is that the field of design research has been putting forward various design process models without a clear description of the problem to be solved. This paper presents our research results from formulating design problems using the axiomatic theory of design modeling (Zeng, 2002).

Design is defined in many ways. As a verb, it refers to the action in design activities and the design process. As a noun, it refers to the output of the design process. The following are some typical definitions of design summarized by Hubka and Eder (Hubka and Eder, 1996):

- Engineering design is the process of applying various techniques and scientific principles for the purpose of defining a device, a process, or a system in sufficient detail to permit its physical realization (Taylor, 1959).
- Engineering design is a purposeful activity directed towards the goal of fulfilling human needs, particularly those which can be met by the technology factors of our culture (Asimow, 1962).
- Design is the process of inventing physical things, which display new physical order, organization, form, in response to function (Alexander, 1979).
- ... the creation of a synthesized solution in the form of products, processes or systems that satisfy perceived needs through mapping between the functional requirements (FRs) in the functional domain and the design parameters (DPs) of the physical domain, through proper selection of the DPs that satisfy the FRs (Suh, 1990).

It can be observed from the definitions above that a design problem consists of human needs, constraints, and functional requirements. A close look into the literature will disclose an even more complex composition of a design problem. We can find design requirements, constraints, design tasks, design intent, design goals and objectives, etc., in the description of a design problem (Yoshikawa, 1981; Altshuller, 1984; Hubka and Eder, 1988; Pahl and Beitz, 1988; Suh, 1990; Braha and Maimon, 1998; Lossack *et al.*, 1998; Gershenson and Stauffer, 1999; Deng *et al.*, 2000; Agouridas *et al.*, 2001; McKay *et al.*, 2001; Hirtz *et al.*, 2002; Lin and Chen, 2002). This complexity is reinforced by the ever-increasing needs for product design due to the advance of technologies. How to classify, formulate, and represent the various kinds of information in a design problem is essential for modeling and improving design practices.

As more and more insights are gained into the understanding of design, there is an increasingly stronger conscience about design: design problem and design solutions evolve simultaneously in the design process. This fact has been recognized in different contexts. Simon characterized design as an ill-defined problem in the context of search paradigm of artificial intelligence (Simon, 1996). He indicated that design is such a problem that its initial and goal states are not clearly defined. Zeng and his collaborators captured the same phenomenon in their philosophical and computational studies of design (Zeng and Cheng, 1991; Zeng *et al.*, 1996; Zeng and Gu, 1999). Zeng and Cheng proposed the recursive logic of design (Zeng and Cheng, 1991). This logic indicates that design is a process recursively generating design solutions and the knowledge to evaluate the solutions. This result is confirmed by Roozenburg's research (Roozenburg, 1992). Based on this logic, a formal design process model is developed where a design problem is not completely defined until the design solutions are found (Zeng and Gu, 1999). This is described in Fig. 1, where R_d , P , and S represent design requirements, product performances, and product descriptions, respectively. Maher and Tang (Maher and Tang, 2003) recently developed a co-evolution model using genetic algorithm to capture this nature. This model was further investigated by Dorst and Cross (Dorst and Cross, 2001). This dynamic and recursive nature of design has added more complexity to the classification, formulation, and representation of design problems.

To support the representation of various information that may appear in a design problem throughout the dynamic and recursive design process, an invariant structure for the formulation of design problem should be developed. This invariant structure provides a foundation to represent not only the identified and known components in a design problem but also the components that might be coming in the future for various reasons.

In achieving the goal of developing an invariant structure for representing design problem, there are two basic approaches. One is bottom-up and the other is top-down. The research with the bottom-up

approach attempts to generalize the invariant structure by observing engineer's design activities. Protocol analysis has been one of the popular methods used for this purpose. Case studies are another method. Results from these explorations include the work accomplished by Pahl and Beitz (Pahl and Beitz, 1988), Hubka and Eder (Hubka and Eder, 1988; Hubka and Eder, 1996), and Hirtz et al (Hirtz *et al.*, 2002). These results are usually described by natural languages, flow charts, graphic illustration, and/or mathematical symbols. On the other hand, the research with the top-down approach attempts to derive the models from the first principles. The axiomatic approach is one of the major tools for this purpose. It usually targets the general design problem and the general model of design so that models for concrete design problems can be logically deduced. At the base of this strategy are axioms, which are self-evident truths that can be taken as the premises for inference. A mathematical tool is usually required for the sake of clarity and consistency. Over the last three decades, different efforts have been made along this research line, such as the general design theory (Yoshikawa, 1981; Tomiyama and Yoshikawa, 1987), the axiomatic theory of design information (Salustri and Venter, 1992), the formal theory of design (Braha and Maimon, 1998), and the axiomatic theory of design modeling (Zeng, 2002). This paper will apply the mathematical operations developed in the axiomatic theory of design modeling to derive the invariant structure for design problem representation.

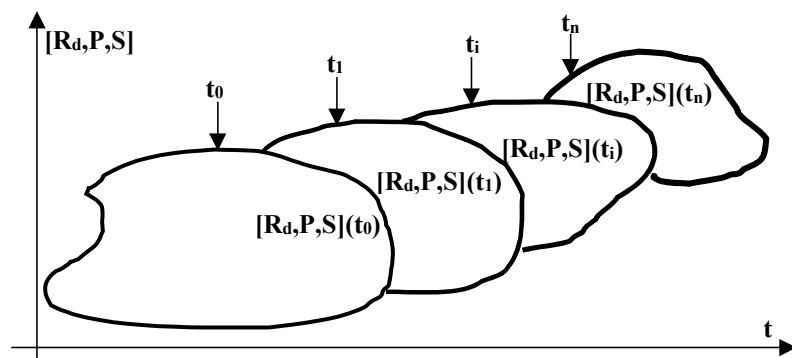


Fig. 1 A Framework of design representation (Zeng and Gu, 1999).

To illustrate the notions and ideas presented in this paper, the following gives a design example adapted from a book by Hubka *et al* (Hubka *et al.*, 1988). The task of this problem is to design a tool for riveting brake linings onto brake shoes for internal drum brakes as shown in Fig. 2 (Hubka *et al.*, 1988). Additional information regarding this design problem includes: the user of the tool is a car mechanic. The hand force, foot force, and the working height should follow ergonomic standards. The safety of the tool should conform to related industry standards. The service life of the tool is around 5 years. The tool should be easy for transportation and maintenance. The tool will be manufactured in a specific workshop. The cost of the tool cannot exceed CAN\$190.00.

From the descriptions above, it can be seen that this design problem includes the following information: functional requirement (riveting brake linings onto brake shoes for internal drum brakes), information about the user (car mechanic, hand and foot forces, working height, etc.), description of the operand of this tool (geometric definition of internal drum brake as in Fig. 2), industrial standards, transportation requirements, financial constraints, manufacturing capabilities, etc.

Next section derives a mathematical structure of a design problem, which includes the structure of a design problem, the sources of product requirements, and the dynamics underlying the design process. Section 3 concludes the work and lists some future research issues. For understanding of the mathematical operations used in this paper, an appendix is provided to introduce the basic concepts of the axiomatic theory of design modeling.

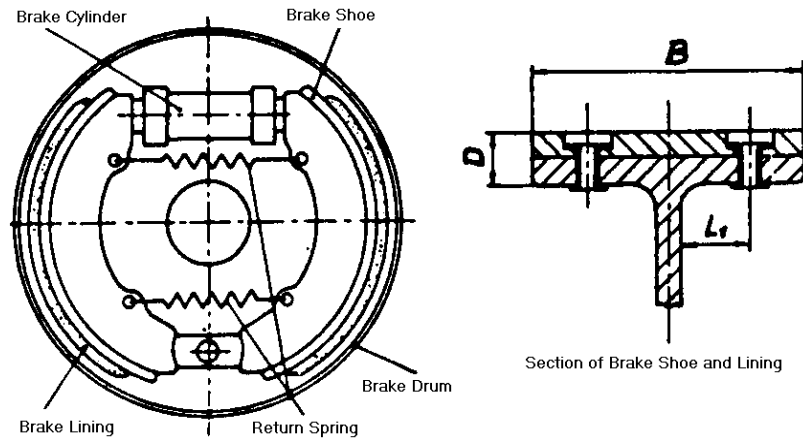


Fig. 2 Internal drum brake (Hubka *et al.*, 1988).

2. Mathematical Structure of Design Problem

In order to capture the invariant structure underlying the complex and changing descriptions of a design problem, this section derives a mathematical structure of design problem using the axiomatic theory of design modeling (Zeng, 2002). This invariant structure is further related to product environment, which is the source of product requirements. This is the foundation for representing various components in a design problem along the dynamic and recursive design process.

In Section 2.1, the notion "product system" is defined using the axiomatic theory of design modeling. Followed by the formal definition of product system, an informal definition of design problem is used to derive the mathematical structure of design problems in Section 2.2. Section 2.3 identifies the source of product requirements by associating design problem to product environment. Section 2.4 derives the dynamics governing the design process.

2.1. Product System

[Definition 1] A *product system* is the structure of an object (Ω) including both a product (S) and its environment (E).

The structure of an object is defined in (37) given in the Appendix. It should be noted that the notion "structure" here is different from that used in the "structure-behavior-function" model of design (Akin, 1992). In the axiomatic theory of design modeling, "structure (\oplus)" is a mathematical operation that includes both an object and relations on this object. The concept "structure" in the "structure-behavior-function" model is the physical configuration of a product. It is only a part of structure defined in this paper.

The product can be a machine, a software package, a process, an idea, etc. Everything except the product itself can be seen as its environment. According to Hubka and Eder, there are direct, close, and remote environments (Hubka and Eder, 1988). For the example of rivet setting tool design, the environment includes the natural environment such as gravity field, the ergonomic environment such as the forces that the user may impose on the tool, the spatial environment such as the physical properties of the brake assembly, the financial environment such as the price of each component, the manufacturing environment such as the available manufacturing tools, and so on.

Let

$$\Omega = E \cup S, \forall E, S[E \cap S = \Phi], \quad (1)$$

where Φ is the object that is included in any object.

Based on the definition of structure operation, the product system ($\oplus\Omega$) can then be expanded as follows:

$$\oplus\Omega = \oplus(E \cup S) = (\oplus E) \cup (\oplus S) \cup (E \otimes S) \cup (S \otimes E), \quad (2)$$

where $\oplus E$ and $\oplus S$ are structures of the environment and product, respectively; $E \otimes S$ and $S \otimes E$ are the interactions between environment and product. A product system can be illustrated in Fig. 3.

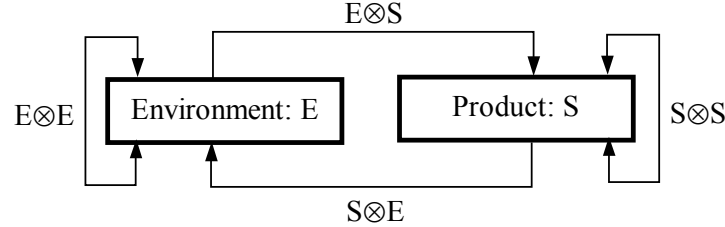


Fig. 3 Product system.

As can be seen from (2), a product system is different from a technical system used by Hubka and Eder (Hubka and Eder, 1996). Technical system refers to the structure of a product, which excludes product environment. Product system is, however, the structure of both a product and its environment.

[Definition 2] Product boundary, denoted by B , is the collection of interactions between a product and its environment.

$$B = (E \otimes S) \cup (S \otimes E). \quad (3)$$

There are two types of product boundary: structural boundary and physical interactions. The structural boundary (B^s) is the shared physical structure between a product and its environment. Examples include the geometric surface of brake lining and shoe in the rivet setting tool design example. The physical interactions include actions (B^a) of the environment on the product and responses (B^r) of the product to its environment. Examples include the hand or foot forces on the rivet setting tool. Therefore, product boundary can be further represented as

$$B = B^s \cup B^a \cup B^r, \forall B^s, B^a, B^r [(B^s \cap B^a = \Phi) \wedge (B^s \cap B^r = \Phi) \wedge (B^a \cap B^r = \Phi)]. \quad (4)$$

Since environment as well as product may have components, structures $\oplus E$ and $\oplus S$ can be further decomposed into the structures of these components as well as their mutual interactions according to the definition of structure operation. For the example of rivet setting tool design, $\oplus S$ includes all its components that may come as a result of design, such as spring, rack and pinion, stopper, counterweight, closure and preform heads, as well as their interactions. Eq. (2) indeed presents a recursive structure of a product system. A more comprehensive study of this recursive structure is done by Zeng *et al.* (Zeng *et al.*, 2003; Zeng *et al.*, 2004).

2.2. Structure of Design Problem

A design problem can be literally defined as a request to design something that meets a set of descriptions of the request. Based on the axiomatic theory of design modeling, both "something" and "descriptions of the request" can be seen as objects and can be further seen as product systems in the context of formulating design problem. Thus a design problem, denoted by P^d , can be formally represented as,

$$P^d = \lambda (\oplus\Omega_0, \oplus\Omega_s), \quad (5)$$

where $\oplus\Omega_0$ ($\Omega_0 = E_0 \cup S_0$, $E_0 \cap S_0 = \Phi$) can be seen as the descriptions of a request for the design, $\oplus\Omega_s$ ($\Omega_s = E_s \cup S_s$, $E_s \cap S_s = \Phi$) is something to be designed, and λ is the "inclusion" relation (\supseteq) implying that $\oplus\Omega_s$ will be a part of $\oplus\Omega_0$ so that the designed product will meet the descriptions of the design. Obviously, if $\oplus\Omega_s$ is a part of $\oplus\Omega_0$, then Eq. (5) is satisfied. At the beginning of the design process, $\oplus\Omega_s$ is an unknown and $\oplus\Omega_0$ is the only thing defined. The truth value of P^d is undetermined, which means the request is yet to be met.

According to (2) and (3), we have

$$\begin{aligned}\oplus\Omega_0 &= (\oplus E_0) \cup (\oplus S_0) \cup B_0, \\ \oplus\Omega_s &= (\oplus E_s) \cup (\oplus S_s) \cup B_s.\end{aligned}\quad (6)$$

Since $E_i \cap S_j = \Phi$, $\forall i, j = 0, s$, according to Lemma 1 given in the appendix, we have

$$P^d = \lambda(\oplus E_0, \oplus E_s) \wedge \lambda(\oplus S_0, \oplus S_s) \wedge \lambda(B_0, B_s). \quad (7)$$

where the symbol \wedge denotes logical "and".

Substitute (4) into (7), and according to Lemma 1 again, we have

$$P^d = \lambda(\oplus E_0, \oplus E_s) \wedge \lambda(\oplus S_0, \oplus S_s) \wedge \lambda(B_0^s, B_s^s) \wedge \lambda(B_0^a, B_s^a) \wedge \lambda(B_0^r, B_s^r). \quad (8)$$

Equation (8) indeed can be organized into three parts: $\lambda(\oplus E_0, \oplus E_s)$, $\lambda(\oplus S_0, \oplus S_s) \wedge \lambda(B_0^s, B_s^s)$, and $\lambda(B_0^a, B_s^a) \wedge \lambda(B_0^r, B_s^r)$. The implications of these three parts are as follows:

(1) $\lambda(\oplus E_0, \oplus E_s)$

$\lambda(\oplus E_0, \oplus E_s)$ corresponds to requirements on product environment. Since it can be generally assumed that the product environment E_0 is not going to be designed, if we intend to change our environment, the part that needs to be changed will be a part of product, which leaves the rest of it as the product environment. As a result, this predicate is always true. The product environment E_0 itself can be taken as an invariant of a design problem. For simplicity, both E_0 and E_s can be denoted by E . As a result, the design problem in (8) can be further represented as

$$P^d = \lambda(\oplus S_0, \oplus S_s) \wedge \lambda(B_0^s, B_s^s) \wedge \lambda(B_0^a, B_s^a) \wedge \lambda(B_0^r, B_s^r), \forall E. \quad (9)$$

(2) $\lambda(\oplus S_0, \oplus S_s) \wedge \lambda(B_0^s, B_s^s)$

$\lambda(\oplus S_0, \oplus S_s)$ defines direct constraints on product. Examples include the working height of the rivet setting tool. They can be seen as a partial definition of a product structure.

$\lambda(B_0^s, B_s^s)$ defines the structural boundary of a product imposed by its environment. Since it is the common part between the product and its environment, the structural boundary is implied in the environment at the beginning of the design process when product is not yet completely defined. This part of environment can be seen as a constraint on the product to be designed.

For example, one of the structural boundaries for the rivet setting tool design is the surface formed by the palm when a mechanic applies his hand force.

Therefore, both $\lambda(\oplus S_0, \oplus S_s)$ and $\lambda(B_0^s, B_s^s)$ are called structural requirements.

(3) $\lambda(B_0^a, B_s^a) \wedge \lambda(B_0^r, B_s^r)$

$\lambda(B_0^a, B_s^a) \wedge \lambda(B_0^r, B_s^r)$ defines direct constraints on actions and/or responses. They are together called performance requirements. Some authors call it functional requirements (Hubka and Eder, 1988; Pahl and Beitz, 1988; Suh, 1990).

Therefore, we have the following theorem:

[Theorem 1] Structure of Design Problem. A *design problem* is implied in a product system and composed of three parts: the environment in which the designed product is expected to work, the requirements on product structure, and the requirements on performances of the designed product.

This theorem can be shown in Table 1.

Table 1 Structure of design problem

Design Problem: P^d	
Product Environment	E
Performance Requirements	$\lambda(B_0^a, B_s^a) \wedge \lambda(B_0^r, B_s^r)$
Structural Requirements	$\lambda(\oplus S_0, \oplus S_s) \wedge \lambda(B_0^s, B_s^s)$

For the rivet setting tool design, the design problem is given in Table 2.

Table 2 Design problem: rivet setting tool design

Rivet Setting Tool Design	
Product Environment	<ul style="list-style-type: none"> • Nature. • Mechanics. • Manufacturing Shop. • Transportation facilities. • Market • Brake: linings and shoe.
Performance Requirements	R-R1. To rivet brake linings onto brake shoes. R-R2. The hand and foot forces should follow ergonomic standards. R-R3. The use of the tool should conform to related industry safety standards. R-R4. The tool can be manufactured in the specific workshop. R-R5. The service life of the tool will be around 5 years. R-R6. The tool should be easy for transportation and maintenance. R-R7. The cost of the tool cannot be over \$190.00
Structural Requirements	R-R8. The working height should follow ergonomic standards

The requirements listed in Table 2 is only a starting point for the design process. New requirements will emerge as the design is refined. These new requirements stem from the interactions between the refined product and its environment.

2.3. Environment-Based Formulation of Design Problem

In describing a design problem, Hubka and Eder (Hubka and Eder, 1996) stated: "Products must be able to perform the desired task (drive the transformation, achieve the results of the process, they must function). They must do this with the demanded performance, operability, sufficiently long life, safety, reliability, adjustability, maintainability, and so on. The above groups of properties are sometimes collected under the term of functionality. In addition, products must be manufacturable to the appropriate quality, packaged, distributed, commissioned, etc. The time scale of planning, designing, manufacturing, and delivering must be suitable. Recycling and disposal have recently become more important. Products must be suitable for humans to operate them (ergonomics) and live with them (esthetics). They must conform to the laws, regulations, standards, codes, moral and ethical

considerations applicable for the place of origin and of use. Cash flow and financing must be considered. They must also be economical, offered at the right price, with appropriate services and support, and acquired at a suitable cost (e.g., to the manufacturer)."

In the statement above, it is not clear which are structural requirements and which are performance requirements because those requirements are not explicitly specified in terms of product structure or product performances. Hence, the application of Theorem 1 to formulate product requirements might generate unnecessary ambiguities. However, it is relatively easy to identify the environment in which the product is expected to work. It can be observed from the statement that any product will work in three environments: natural, built, and human. To work in natural environment, a product should obey all natural laws, otherwise the product will not be able to exist. This involves requirements such as safety and reliability. The built environment includes all artifacts built or created by human beings. To work in the built environment, a product must satisfy the requirements such as manufacturability and transportability. The human environment includes all human users and operators in the life cycle of a product. To survive in the human environment, a product must satisfy the requirements such as salability, operability, and maintainability. Denote natural, built, and human environments by E^n , E^b , E^h , respectively, we have

$$E = E^n \cup E^b \cup E^h. \quad (10)$$

It can be seen from (3) that B_s^s , B_s^a , and B_s^r in (9) are originated from environment. However, $\oplus S_s$ in (9) seems to be independent of environment. But in real life design problem, any structural requirement is given either by regulations and standards or restrictions from three environments defined in (10). Hence, it is safe to say that all the product requirements are imposed by the product environment in which the product is expected to work. Formally, $\lambda(\oplus S_0, \oplus S_s)$ can be viewed as an implication of $\lambda(B_0^s, B_s^s)$. Therefore,

$$P^d = \lambda(B_0^s, B_s^s) \wedge \lambda(B_0^a, B_s^a) \wedge \lambda(B_0^r, B_s^r), \forall E. \quad (11)$$

The following derives a formulation for organizing and classifying product requirements in terms of product environment. In general, the natural, built, and human environment can usually be further divided into different types and/or parts. Suppose

$$E = \bigcup_{i=1}^n E_i, \exists E_1, E_2, \dots, E_n, \forall i, j, i \neq j, E_i \cap E_j = \Phi, \quad (12)$$

where n is a finite positive number. Each E_i can be an individual environment. Substituting (12) into (3), we have

$$\begin{aligned} B &= (E \otimes S) \cup (S \otimes E) \\ &= [(\bigcup_{i=1}^n E_i) \otimes S] \cup [S \otimes (\bigcup_{i=1}^n E_i)]. \end{aligned} \quad (13)$$

Applying (36) in the Appendix,

$$B = \bigcup_{i=1}^n [(E_i \otimes S) \cup (S \otimes E_i)]. \quad (14)$$

Let

$$B^i = (E_i \otimes S) \cup (S \otimes E_i). \quad (15)$$

We have

$$B = \bigcup_{i=1}^n B^i. \quad (16)$$

By substituting (12) and (16) into (11), we have

$$P^d = [\lambda (\bigcup_{i=1}^n B_0^i, \bigcup_{i=1}^n B_s^i)], \forall E = \bigcup_{i=0}^n E^i. \quad (17)$$

where both B_0^i and B_s^i are defined as follows:

$$\begin{aligned} B_0^i &= (E_i \otimes S_0) \cup (S_0 \otimes E_i), \\ B_s^i &= (E_i \otimes S_s) \cup (S_s \otimes E_i). \end{aligned} \quad (18)$$

Therefore, Equations (17) and (18) define a design problem based on the environment components included in the descriptions of a design problem. As such, we have the following theorem:

[Theorem 2] Source of Product Requirements. All the product requirements in a design problem are imposed by the product environment in which the product is expected to work.

Based on this theorem, a design problem can be formulated based on the product environment $\bigcup_{i=0}^n E^i$. Obviously, different ways to organize the components in product environment will lead to different formulations of product requirements. Chen and Zeng formulate design problems in terms of different classification schemes of environment (Chen and Zeng, 2004).

Table 3 Example of product requirement classification

	E_n	E_h	E_m	E_p	E_t	E_b
Structural		R-R8				
Performance	R-R3, R-R5	R-R1, R-R2, R-R3, R-R4, R-R5, R-R6	R-R7	R-R4	R-R6	R-R1

For the example of rivet setting tool design, corresponding to different types of environment, we have ergonomic requirements, manufacturing requirements, financial requirements, transformation requirements, operand requirements, etc. Table 3 organizes the requirements on the rivet setting tool design in terms of the natural, human, market, production, transportation, and operand environment, which are denoted by E_n , E_h , E_m , E_p , E_t , and E_b , respectively. The operand of the riveting tool is the brake. Human environment may include the mechanic, maintenance technicians, transportation movers, etc. The index "R-Rx" comes from Table 2.

Formally, the product environment for this design problem can be represented as

$$E = E_n \cup E_h \cup E_m \cup E_p \cup E_t \cup E_b. \quad (19)$$

Therefore, the product system for this design problem at the beginning of the design process can be illustrated in Fig. 4. Operand (brake and shoe), production, transportation are built environment, which is made up of artifacts. Natural environment imposes safety requirements. Mechanic and market constitute the human environment, together with other operators in the life cycle of the tool.

We have developed a software system to capture the product, environment, and their interactions implied in a natural language based description of a design problem. Fig. 5 is the result for the rivet setting tool design. The details of this system can be found in another paper (Chen *et al.*, 2004).

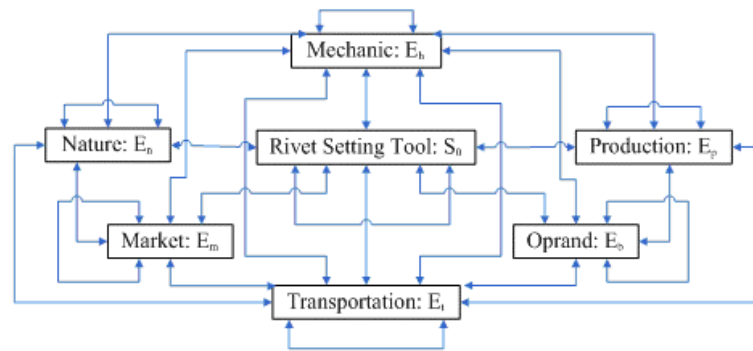


Fig. 4 Product system for the riveting tool design problem.

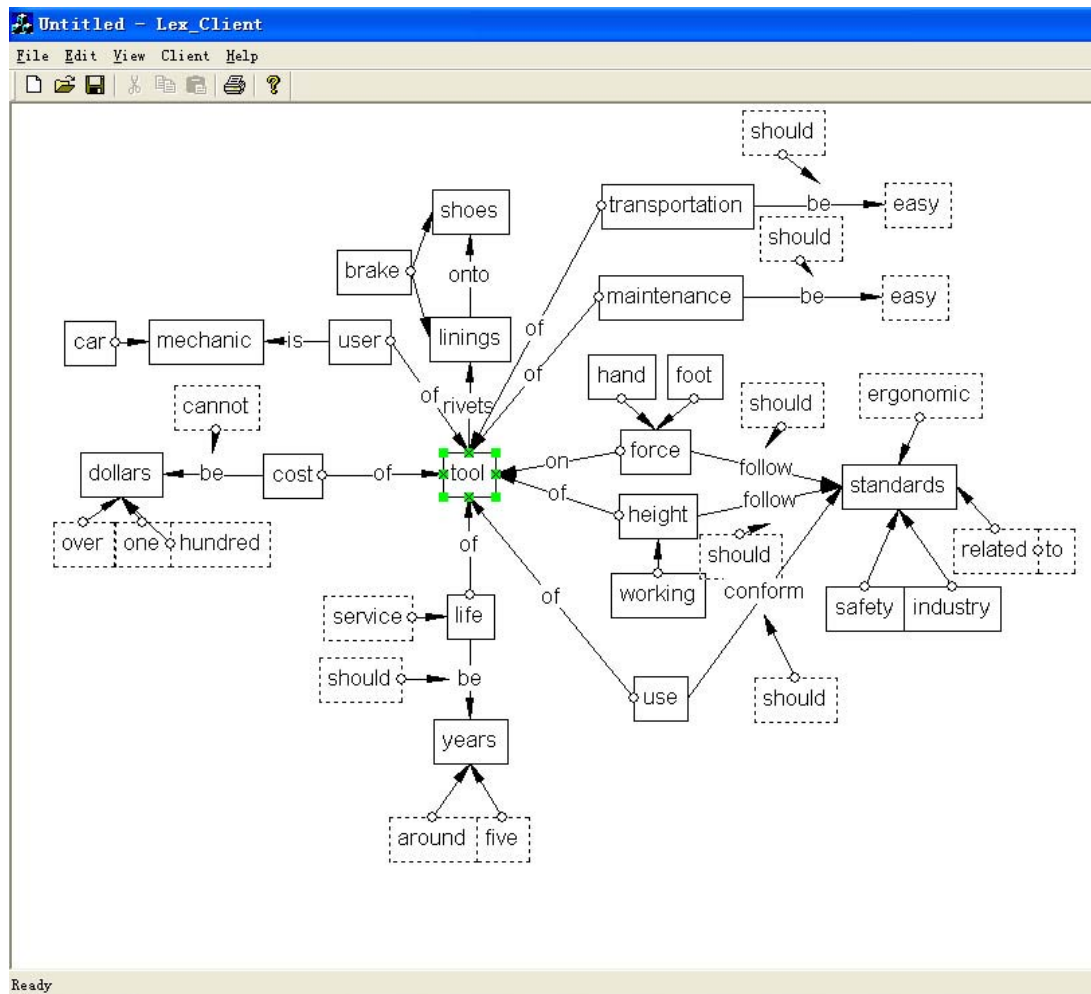


Fig. 5 Identification of product system from natural language based description of a design problem (Chen *et al.*, 2004).

2.4. Dynamics Underlying Design Problem

As was given in Fig. 1, design problem and product descriptions evolve along the design process. Theorems 1 and 2 present a static structure of design problem. This section will discuss the mechanism driving the evolution of design by looking at the relations between product and design problem.

For the rivet setting tool design, Fig. 6 lists a design concept generated in reference (Hubka *et al.*, 1988) and the components included in this concept. Without loss of generality, it can be assumed that this concept was generated in the order of handle, punch heads, transmission rod, spring, and frame. The generation process is discussed in a separate paper (Zeng, 2004). Formally, these components are denoted by s_1 , s_2 , s_3 , s_4 , and s_5 , respectively. The concept evolution process can then be illustrated in Fig. 7.

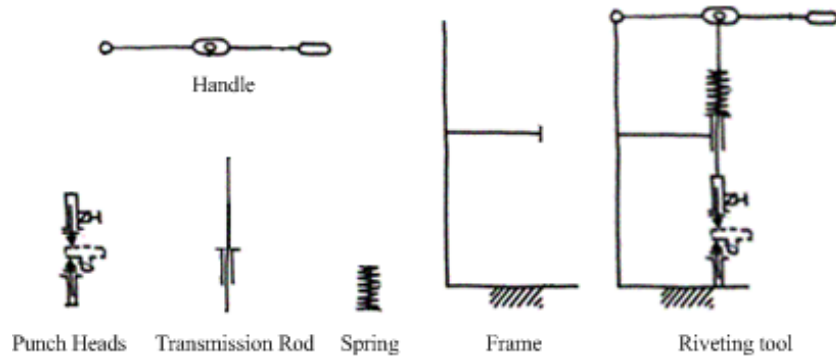


Fig. 6 Rivet setting tool: one design concept.

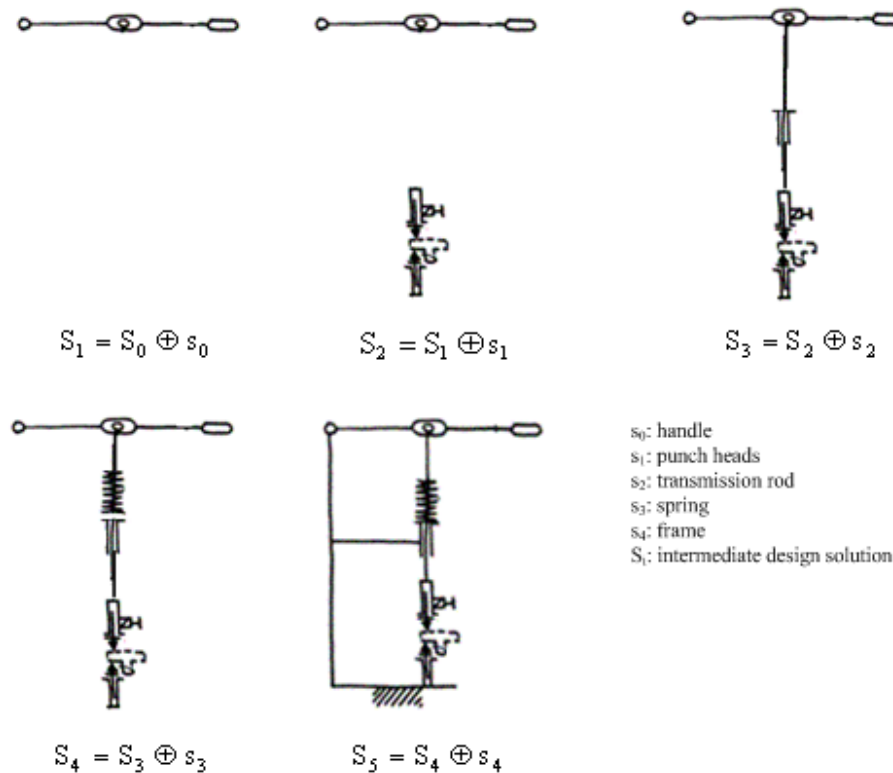


Fig. 7 Rivet setting tool design: an example of concept development.

This evolution process can be generalized as is shown in Fig. 8. The product description is refined as the design progresses.

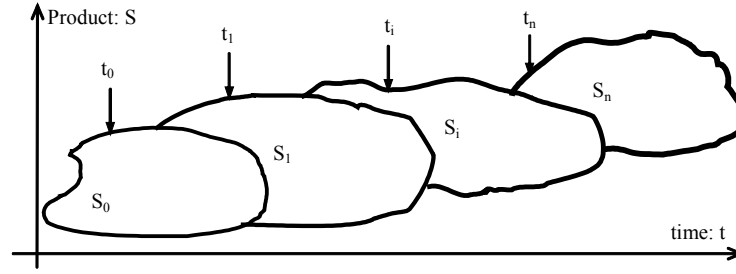


Fig. 8 Evolution of product in the design process.

At each stage of the product evolution, both B_i^j and B_{i+1}^j are defined as follows:

$$\begin{aligned} B_i^j &= (E_j \otimes S_i) \cup (S_i \otimes E_j); \\ B_{i+1}^j &= (E_j \otimes S_{i+1}) \cup (S_{i+1} \otimes E_j). \end{aligned} \quad (20)$$

According to (17) and (20), the design problem evolves along the evolution of the product. At each stage of this evolution process, the design problem is defined by its current product system $\oplus\Omega_i$, which is called the state of the design. Suppose that P_i^d is the design problem at the i^{th} stage of the design process, it can be represented as

$$P_i^d = K_i^e(\oplus\Omega_i). \quad (21)$$

where K_i^e is evaluation operator responsible for identifying the conflicts between the current and desired states of design. Thus P_i^d is a proposition or predicate. The components of product system $\oplus\Omega_i$ keep on changing while P_i^d is not true. This evolution process of design can be illustrated as in Fig. 9.

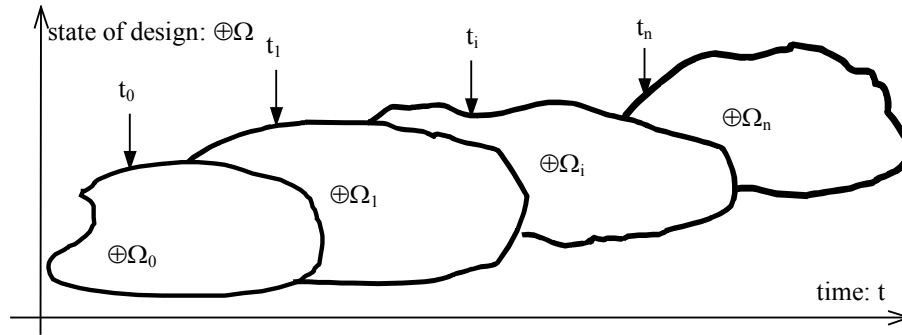


Fig. 9 Evolution of the design process: refined.

It can be seen from (20) that though the product environment does not change in most cases throughout the design process, the product-environment boundary ($B_i = (E_i \otimes S_i) \cup (S_i \otimes E_i)$) may still be updated every time when the design solutions S_i are refined to S_{i+1} . As a result, the design problem (P_i^d) will be updated as the design process progresses. This results in the zig-zag design process as is

shown in Fig. 10. This nature of design is revealed in the recursive logic of design (Zeng and Cheng, 1991). It was also discussed by Suh (Suh, 1990), Dorst and Cross (Dorst and Cross, 2001), Maher and Tang (Maher and Tang, 2003), etc.

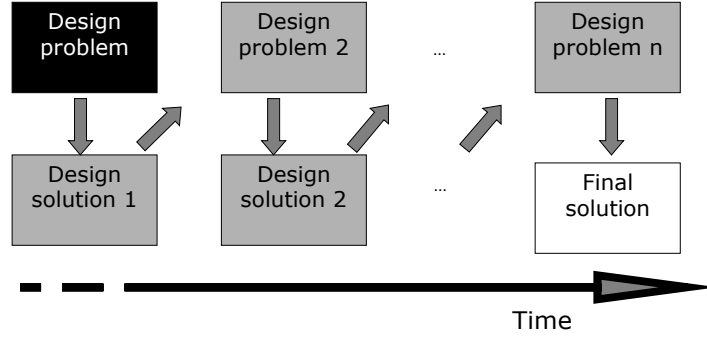


Fig. 10 Zig-zag design process.

This phenomenon can be stated as the following theorem:

[Theorem 3] Dynamic Structure of Design Problem. In the design process, design solutions to a design problem may change the original design problem, if the design solutions are different from their precedents, either by refinement or by alteration.

As can be seen in Fig. 10, for each design problem P_i^d , there may exist design solutions S_i so that a new state of design, $\oplus\Omega_{i+1}$, can be derived as follows:

$$\oplus\Omega_{i+1} = K_i^s(P_i^d), \quad (22)$$

where K_i^s is a synthesis operator responsible to generate design concepts from a design problem.

By substituting (21) into (22), we have

$$\oplus\Omega_{i+1} = K_i^s(K_i^e(\oplus\Omega_i)). \quad (23)$$

We can also substitute (22) into (21),

$$P_i^d = K_i^e(K_i^s(P_{i-1}^d)). \quad (24)$$

However, since P_i^d can always be derived from $\oplus\Omega_i$ according to (21), Eq. (24) is implied in (23). Hence, (23) is sufficient to represent Theorem 3. This is also the mathematical form of Theorem 3. The form of this equation has evolved along with the improvement of the formal tool for representing design activities (Zeng and Gu, 1999; Zeng and Gu, 1999; Zeng, 2001).

Eq. (23) is called *design governing equation*. It underlies the design process and governs design activities. It defines the dynamics of design. The basic concept behind this equation is the recursive logic of design (Zeng and Cheng, 1991), which states that design is a recursive process in which the design solution and design problem interdependently evolve (Zeng and Cheng, 1991; Zeng and Cheng, 1993; Dorst and Cross, 2001; Maher and Tang, 2003).

Eq. (23) can be further formulated as

$$\oplus\Omega_{i+1} = D^i(\oplus\Omega_i) \text{ where } D^i = K_i^s \bullet K_i^e, \quad (25)$$

where " \bullet " combines two relations.

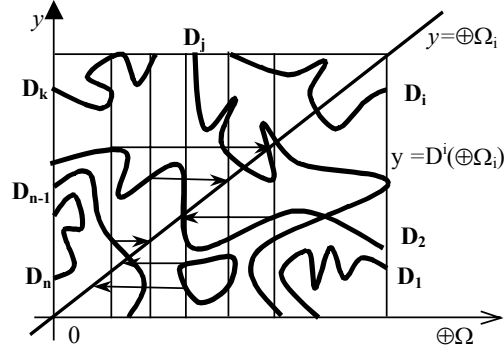


Fig. 11 The process looking for design solutions.

Eq. (25) means that design problem solving is a process looking for fixed points of the function D^i . The fixed points for (25) are the interaction points between $y = \oplus\Omega_i$ and $y = D^i(\oplus\Omega_i)$. A fixed point is usually found through an iterative method. Starting from an initial design state $\oplus\Omega_0$, $\oplus\Omega_1$ can be found through (25), which updates the design state as well as the function D^i . If a sequence of $\oplus\Omega_i$ has a limit $\oplus\Omega_s$, then $\oplus\Omega_s$ is a root of (25). This process is illustrated in Fig. 11. In the Figure, D^0, D^1, \dots , and D^n are the function in (25) for each stage of the design process. They define different domains for the exploration of design solutions.

Based on (25), the relation between the final design solution and the initial problem definition can be represented as

$$\oplus\Omega_s = D^n D^{n-1} \dots D^0(\oplus\Omega_0) \quad (26)$$

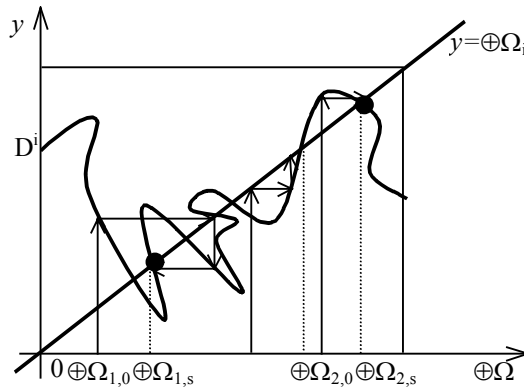


Fig. 12 Multiple solutions of a design problem.

The design process above is indeed consistent with the recursive logic of design (Zeng and Cheng, 1991). The evaluation operator K_i^e is defined in terms of the design solutions generated by the synthesis operator K_i^s in each step of the design process. This fact means that the two operators K_i^e and K_i^s are interdependent. Hence, function D^i is nonlinear. As a result, there may exist multiple fixed points for the same design problem. The convergence depends on the initial design state. This process can be seen in Fig. 12.

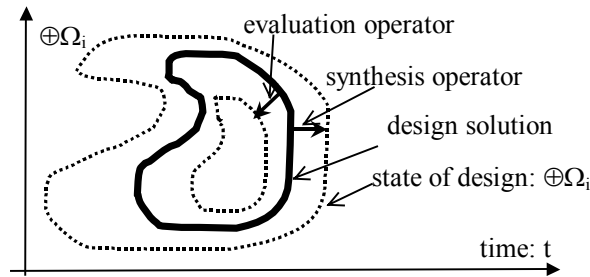


Fig. 13 State space of design under synthesis and evaluation operators.

The two operators K_i^e and K_i^s in (23) correspond to two major phases in the design process: synthesis and evaluation. The synthesis process is responsible for proposing a set of candidate design solutions based on the design problem. It stretches the state space of design. The evaluation process is used to screen candidate solutions against the requirements in the design problem. It folds the state space of design. The interaction of both synthesis and evaluation processes gives rise to the final balanced design solutions as shown in Fig. 13. Equation (26) provides a dynamic mechanism for creative design (Zeng *et al.*, 2004).

3. Conclusions

A mathematical formulation of design problem is derived following logical steps using the axiomatic theory of design modeling. This formulation defines the structure of design problems: environment, structural requirements, and performance requirements. These components are further related to product environment, which is divided into natural, built, and human environments. These environments are the sources of product requirements

On the basis of the formal representation of design problem, the dynamic nature of design is captured in the design governing equation, which considers the interaction between synthesis and evaluation operations in the design process in a formal and integrated manner. The interdependence between design problem and product descriptions is implied in this equation.

A unique nature of this research is its research methodology. Traditionally, a design theory is established using speculation, observation, and experimentation and is validated through computer implementation, protocol analysis, and case studies. This research, however, derives the theory from the first principle. Hence, it is validated by the premises and the mathematical logic behind it.

This work is the foundation for a formal design process model, which deals with the decomposition of design problem in a more effective manner. Future research includes the refined studies of product environment and their roles in triggering product requirements, based on which product requirements can be managed in a more rational manner throughout the design process.

4. Acknowledgements

This work is partially supported by NSERC (Grant number RGPIN 298255). The author also thanks his graduate students Z. Y. Chen, B. Q. Yan, and S. J. Yao for their comments and suggestions.

5. References

- Agouridas, V., Baxter, J., de Pennington, A. and McKay, A., 2001, "On defining product requirements: a case study in the UK health care sector," Proceedings of ASME 2001 Design Engineering Technical Conferences and Computer and Information in Engineering Conference, Pittsburgh, Pennsylvania.
- Akin, O., 1992, "A structure and function based theory for design reasoning," Research in Design Thinking, N. F. M. Roozenburg, Delft University Press, pp. 37-60.

- Alexander, E. R., 1979, "The design of alternatives in organizational contexts: a pilot study," *Administration Science Quarterly*, Vol. 24, No., pp.382-404.
- Allan, J. K. and Mistree, F., 1997, "Decision-based design: Where should we go from here?," Notes of 1997 Decision-Based Design Workshop.
- Altshuller, G. S., 1984, *Creativity as an Exact Science: the Theory of the Solutions of Inventive Problems*, Gordon and Breach Science Publishers.
- Asimow, M., 1962, *Introduction to Design*, Englewood Cliffs, NJ, Prentice-Hall.
- Braha, D. and Maimon, O., 1998, *A Mathematical Theory of Design*, Kluwer Academic Publishers.
- Chen, Z. and Zeng, Y., 2004, "Classification of product requirements using axiomatic theory of design modeling," *Concurrent Engineering: Research and Applications*, An International Journal, Submitted.
- Chen, Z. Y., Yao, S. J. and Zeng, Y., 2004, "A systematic approach for the specification of user requirements," The Inaugural CDEN Design Conference, Montreal, Canada.
- Cross, N., Ed., 1984, *Developments in Design Methodology*, Wiley, Chichester.
- Deng, Y. M., Tor, S. B. and Britton, G. A., 2000, "Abstracting and exploring functional design information for conceptual mechanical product design," *Engineering with Computers*, Vol. 16, pp.36-52.
- Dorst, K. and Cross, N., 2001, "Creativity in the design process: co-evolution of problem-solution," *Design Studies*, Vol. 22, No. 5, pp.425-437.
- Ertas, A., Tanik, M. M. and Maxwell, T. T., 2000, "Transdisciplinary engineering education and research model," *Transaction of the SDPS: Journal of Integrated Design & Process Science*, Vol. 5, No. 4, pp.1-21.
- Gershenson, J. K. and Stauffer, L. A., 1999, "A taxonomy for design requirements from corporate customers," *Research in Engineering Design*, Vol. 11, pp.103-115.
- Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S. and Wood, K. L., 2002, "A functional basis for engineering design: reconciling and evolving previous efforts," *Research in Engineering Design*, Vol. 13, pp.65-82.
- Hubka, V., Andreasen, M. and Eder, W., 1988, *Practical Studies in Systematic Design*, Butterworths.
- Hubka, V. and Eder, W., 1988, *Theory of Technical Systems*, Springer-Verlag.
- Hubka, V. and Eder, W., 1996, *Design Science*, Springer.
- Lin, L. and Chen, L. C., 2002, "Constraints modeling in product design," *Journal of Engineering Design*, Vol. 13, No. 3, pp.205-214.
- Lossack, R. S., Umeda, Y. and Tomiyama, T., 1998, "Requirement, function and physical principle modeling as the basis for a model of synthesis," *Computer Aided Conceptual Design'98, Proceedings of the 1998 Lancaster International Workshop on Engineering Design*.
- Maher, M. and Tang, H., 2003, "Co-evolution as a computational and cognitive model of design," *Research in Engineering Design*, Vol. 14, pp.47-63.
- Maxwell, T. T., Ertas, A. and Tanik, M. M., 2002, "Harnessing complexity in design," *Transaction of SDPS: Journal of Integrated Design and Process Science*, Vol. 6, No. 3, pp.63-74.
- McKay, A., de Pennington, A. and Baxter, J., 2001, "Requirements management: a representation scheme for product specifications," *Computer-Aided Design*, Vol. 33, pp.511-520.
- Pahl, G. and Beitz, W., 1988, *Engineering Design: A Systematic Approach*, Springer-Verlag.
- Rozenburg, N. F. M., 1992, "On the logic of innovative design," *Research in Design Thinking*, N. F. M. Rozenburg, Delft University Press, pp. 127-138.
- Salustri, F. A. and Venter, R. D., 1992, "An axiomatic theory of engineering design information," *Engineering with Computers*, Vol. 8, No. 4, pp.197-211.
- Simon, H., 1996, *The Sciences of the Artificial*, The MIT Press.
- Suh, N., 1990, *The Principles of Design*, Oxford University Press.
- Taylor, E. S., 1959, M.I.T. Report, Cambridge, MA, MIT Press.

Tomiya, T. and Yoshikawa, H., 1987, "Extended general design theory," Design Theory for CAD, Proceedings IFIP WG5.2 Working Conference on Design Theory for CAD, E. A. Warman, Amsterdam, Elsevier Science Publishers B. V., pp. 95-130.

Yoshikawa, H., 1981, "General design theory and CAD system," Man-Machine Communication in CAD/CAM, Proceedings IFIP WG 5.2 Working Conference, E. A. Warman, Amsterdam, North Holland, pp. 35-38.

Zeng, Y., 2001, An Axiomatic Approach to the Modeling of Conceptual Product Design Using Set Theory. Department of Mechanical and Manufacturing Engineering, Ph.D. Thesis, University of Calgary.

Zeng, Y., 2002, "Axiomatic theory of design modeling," Transaction of SDPS: Journal of Integrated Design and Process Science, Vol. 6, No. 3, pp.1-28.

Zeng, Y., 2004, "Environment-based design: process model," Technical Report, Concordia Institute for Information Systems Engineering, Concordia University, Montreal.

Zeng, Y. and Cheng, G., 1991, "On the logic of design," Design Studies, Vol. 12, No. 3, pp.137-141.

Zeng, Y. and Cheng, G., 1993, "Knowledge based free mesh generation of quadrilateral elements in two dimensional domains," Microcomputers in Civil Engineering, Vol. 8, No. 4, pp.259-270.

Zeng, Y. and Gu, P., 1999, "General Design Governing Equation," NSFC Grantee's Conference on Design and Manufacturing Engineering, Vancouver, BC, Canada.

Zeng, Y. and Gu, P., 1999, "A science-based approach to product design theory Part I: Formulation and formalization of design process," Robotics and Computer-Integrated Manufacturing, Vol. 15, No. 4, pp.331-339.

Zeng, Y., Jin, J., Liu, H. and Xie, X., 1996, Formal Framework for Architectural Design. Chongqing, Institute of Logistics Engineering.

Zeng, Y., Pardasani, A., Antunes, H., Li, Z., Dickinson, J., Gupta, V. and Baulier, D., 2003, "Representation and interpretation of sketches in mechanical design: experimental and theoretical approaches," Proceedings of ASME 2003 Design Engineering Technical Conferences and Computer and Information in Engineering Conference, Chicago, USA.

Zeng, Y., Pardasani, A., Antunes, H., Li, Z., Dickinson, J., Gupta, V. and Baulier, D., 2004, "Mathematical foundation for modeling conceptual design sketches," Transactions of the ASME: Journal of Computing and Information Science in Engineering, Vol. 4, No. 2, pp.150-159.

Zeng, Y., Yan, B. Q., Chen, B. and Yao, S. J., 2004, "A Theoretical and Experimental Study on Design Creativity," The Inaugural CDEN Design Conference, Montreal, Canada.

APPENDIX: AXIOMATIC THEORY OF DESIGN MODELING

Axiomatic theory of design modeling provides a logical tool for representing and reasoning about object structures (Zeng, 2002). It allows for the development of design theories following logical steps based on mathematical concepts and axioms. It is different from the existing design theories that use mathematical symbols to represent the notions and ideas in the theory. Here, mathematics is used not only as a formal representation instrument, but also used as a logic tool.

Three primitive concepts are used in the axiomatic theory of design modeling: universe, object, and relation. Their informal definitions are adapted from Random House Webster's Unabridged Dictionary and are given in the following:

[Definition 1] The universe is the whole body of things and phenomena observed or postulated. It is denoted by U .

[Definition 2] An object is anything that can be observed or postulated in the universe. It is denoted by any capital letter in English, such as O, A, B , etc.

It can be seen from the two definitions above that universe is the whole body of objects.

[Definition 3] A relation is an aspect or quality that connects two or more objects as being or belonging or working together or as being of the same kind. Relation can also be a property that holds between an ordered pair of objects. It is denoted by \sim .

If two objects A and B are related, then this relation can be represented symbolically as

$$A \sim B, \exists A, B, \quad (27)$$

where A and B are objects. $A \sim B$ is read as "A relates to B".

It should be noted that symbol \otimes was used to denote the general relation (Zeng, 2002). In the process of the evolution of this theory, the author had to make a decision to keep \otimes for a more specific relation (structure operation).

Based on definitions 1-3, the following two axioms can be introduced:

[Axiom1] Everything in the universe is an object.

[Axiom 2] There are relations between objects.

According to Axiom 1, relations between objects are also objects. Therefore,

$$R = A \sim B, \forall A, B, \exists R, \quad (28)$$

where R is the relation from object A to object B . A relation of one object to itself is called the relation on the object itself.

Formally, relations can be classified as:

1) Idempotent relation

$$R = A \sim A, \forall A. \quad (29)$$

2) commutative relation

$$R = A \sim B = B \sim A, \exists A, B, R. \quad (30)$$

3) Transitive relation

$$R = A \sim C \text{ if } R = A \sim B = B \sim C, \exists A, B, C, R. \quad (31)$$

4) Associative relation

$$(A \sim B) \sim C = A \sim (B \sim C), \exists A, B, C. \quad (32)$$

5) Distributive relation

$$(A \sim_1 B) \sim_2 C = (A \sim_2 C) \sim_1 (B \sim_2 C), \exists A, B, C, \sim_1, \sim_2. \quad (33)$$

It can be seen from Axioms 1 and 2 that the characteristics of relations would play a critical role in the axiomatic theory of design modeling. We need to define a group of basic relations to capture the nature of object representation. There are many types of relations, such as structural relations, interactions, temporal relations, and logical relations. Based on a defined group of basic relations, new relations can be established in the theory. These basic relations are the corollaries of this theory. This paper will need the following two corollaries:

[Corollary 1] Every object in the universe includes other objects. Symbolically,

$$A \supseteq B, \forall A \exists B. \quad (34)$$

B is called a subobject of A. The symbol \supseteq is inclusion relation. The inclusion relation is transitive and idempotent but not commutative. Other operations such as \subseteq , $=$, \cup , and \cap are also defined based on this corollary (Zeng, 2002).

[Corollary 2] Every object in the universe interacts with other objects. Symbolically,

$$C = A \otimes B, \forall A, B \exists C. \quad (35)$$

C is called the interaction of A on B. The symbol \otimes represents interaction relation. Interaction relation is idempotent but not transitive or associative. The following rules hold for interaction relations:

$$\begin{aligned} A \otimes (B \cup C) &= (A \otimes B) \cup (A \otimes C), \\ (A \cup B) \otimes C &= (A \otimes C) \cup (B \otimes C), \\ A \otimes (B \cap C) &= (A \otimes B) \cap (A \otimes C), \\ (A \cap B) \otimes C &= (A \otimes C) \cap (B \otimes C). \end{aligned} \quad (36)$$

Based on the above two corollaries, the structure operation is established. It provides the aggregation mechanism for representing the object evolution in the design process. The definition of this operation is given as follows:

[Definition 4] *Structure operation*, denoted by \oplus , is defined by the union of an object and the relation of the object to itself.

$$\oplus O = O \cup (O \otimes O), \quad (37)$$

where $\oplus O$ is the structure of object O.

The following lemma holds for these relations:

[Lemma 1] For four objects A_1, A_2, B_1 , and B_2 , $A_i \cap B_j = \Phi$ ($i, j=1, 2$). If $(A_1 \cup B_1) \supseteq (A_2 \cup B_2)$, then $A_1 \supseteq A_2$ and $B_1 \supseteq B_2$.

Φ is the object that is included in any object.

It should be noted that though the mathematical symbols in this theory look like the ones from elementary set theory, this theory is not the same as set theory. A basic operation " \in " in set theory is removed from this theory. The distinction between "class" and "element" has hence disappeared.