THE UNIVERSITY OF CALGARY

Axiomatic Approach to the Modeling of Product Conceptual Design Processes Using Set Theory

By

Yong Zeng

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL AND MANUFACTURING ENGINEERING

CALGARY, ALBERTA

MAY, 2001

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

0-612-64894-X

Abstract

This thesis presents an axiomatic approach to the modeling of product conceptual design processes using set theory. It can be divided into three parts: the axiomatic system, the nature of the design problem and design processes, and applications. The first part aims to establish the theory whereas the latter two test and justify the theory.

The axiomatic system consists of two axioms: axiom of bounded rationality and axiom of object structuring. The axiom of bounded rationality states that human recognition is not perfect while the axiom of object structuring indicates what should be a full picture of an object. These two axioms deal with human and natural parts in the design process respectively. Set theory is used as the language to represent axioms, theorems, and facts appearing in the theory. Based on this theory, formal models of the product-environment system, design requirements, and the design process are derived following logical steps. These formal models are characterized by: the dynamic and evolving nature of product descriptions and product performances, the uniform representation of design requirements, and an environment decomposition-based conceptual design process. These three parts constitute an integral formal model of product design. It supports the overall design process from the abstract and general state to the concrete and specific.

A design governing equation, which captures the ill-structured nature of design problem, is obtained from the axiomatic system. This equation implies that design problem solving is a process looking for fixed points under the design function, which is nonlinear in nature. This associates designing to nonlinear dynamics and leads to an explanation of design creativity. In this way, the randomness and uncertainty of design creativity could have a position in a scientific framework with determined laws. These are the three routes to creative designs presented in the thesis. To illustrate and test the ideas in this theory, a rivet setting tool design case study is used throughout the thesis.

This thesis also presents a mechanism design software developed based on the principles implied in the established theory. The software automatically generates multiple design concepts for changing straight-line motions merely with the performance knowledge. This software prototype demonstrates the usefulness of the theory in the development of conceptual design tools in aiding design engineers.

Acknowledgements

I would like to thank my supervisor Dr. P. Gu. He provided me the chance to do the kind of research I have accomplished in this thesis three years ago. He has created a very flexible and reliable environment for free scientific exploration while his strong vision always leads the exploration to fruitful results. His financial support made it possible for me to focus on my research. His strong drive to achieve excellence has also motivated me in every aspect of my life. I would also like to thank Dr. Fauvel and Dr. Xue, my supervisory committee members, for their very constructive suggestions and comments on my research. They have been a great help, support and encouragement. Part of the research presented here is accomplished under the support of the Dean's Special Doctoral Scholarship from the Faculty of Graduate Studies.

My interest in design research started about ten years ago when I was doing course work in artificial intelligence for my Master of Science degree. My former supervisor, Dr. G. Cheng at Dalian University of Technology, encouraged the interest and helped me publish a paper in the logic of design. His creative thinking and sincerity toward academic research have nourished me throughout my academic career.

I owe Dr. Haghighat at Concordia University in Montreal for my being able to come to Canada and to restart my research in design which was interrupted for years. During my short stay with him, he had given me a lot of support in my life as well as in my research.

During my thesis work, I have benefited a lot from a discussion group on design research. I would like to extend my thanks to Mr. Paul Wenkelman, Mr. Mehdi Hashemian, and Mr. Neil Schemenauer for the joyful and pleasant discussions.

Two important ideas in my thesis were from the International Fellowship Group in the Brentview Baptist Church. One was the adoption of bounded rationality as an axiom, another was the separation of environment as a crucial part of design theory. These two ideas came to me when we had intensive discussions on creation and eternity.

Miss Nareeza, our department graduate secretary, has always been a pleasant and a resourceful source for seeking administrative help.

Last but not least, I want to thank my beloved family for their support over years. I can not imagine being able to reach this point without their love.

I dedicate this thesis to Prof. Hongbao Liu, my respectful teacher and friend

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1 Introduction

1.1 Motivation

Engineering design is a crucial component of the product realization process. It is estimated that over 70 percent of the total life cycle cost of a product is determined during the design stage. Effective product design can improve quality, reduce cost, and reduce time to market, thereby produce products that match customer needs more precisely. Improving the practice of product design is essential to industrial excellence and competitiveness. An effective way to achieve this goal is to develop product design tools. Over the years, a great deal of effort has been made to develop different kinds of design tools. By far the most work to date has been done on synthesis at the parametric level. Optimization techniques, Taguchi and other statistical methods, as well as knowledge-based methods have been developed with varying degrees of generality and usefulness. However, compared to the relatively mature study in parametric design, conceptual design is still an area lacking robust theories and models, despite the fact that it is one of the critical design stages where some of the most important design decisions are made. To support the development of conceptual design tools with better efficiency, enhanced quality, and/or less resources, the main aim of this thesis is to understand and model conceptual design process.

1.2 Research Methodology and Scheme

Many methods are available to achieve the goal of understanding and modeling design (Cross, 1992). This thesis will adopt the deductive method, which depends on logic and reasoning. The scheme of this method is illustrated in Figure 1-1. Models of design processes identify the factors affecting design processes and describe how a design solution is found. Design tools target to improve design by managing those factors in the prescribed design process.

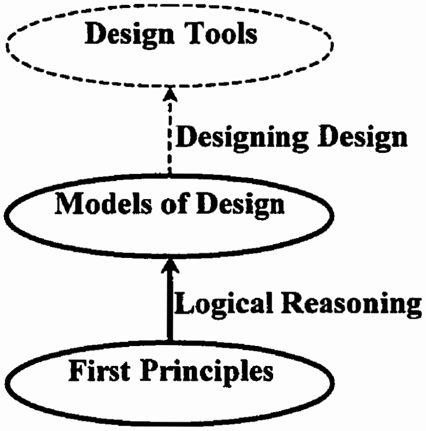


Figure 1-1 Approach of the thesis research

This is considered a theory building process that has been following the procedures proposed by Popper (1961). He suggested that in establishing a scientific theory to explain some phenomena or experimental data, a conjecture should be made in the first place. Conclusions will then be deduced from the conjecture. If there is one conclusion found to contradict the phenomena or experimental data then the conjecture should be rejected or modified. Three elements are essential in Popper's proposal: 1) propose conjecture; 2) derive conclusions from the conjecture in a deduction chain; and 3) compare the conclusions with the existing phenomena or experimental data. The schema is shown in Figure 1-2.

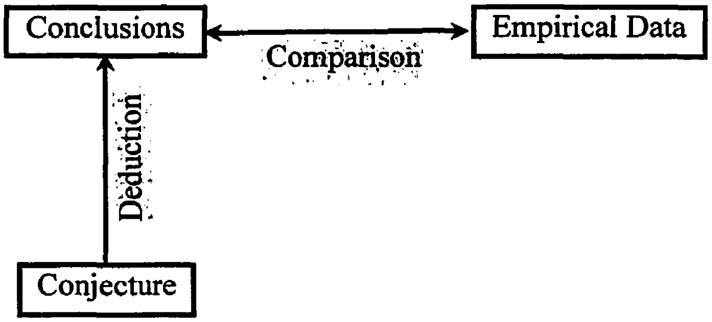


Figure 1-2 Schema of scientific research

Based on Popper's schema, a scheme is illustrated in Figure 1-3 for the current research. The core of this scheme is an axiomatic system which includes axioms and a language. Axioms are conjectures required by Popper's theory. They provide the first principles for the exploration. The language gives the basic means for representing each entity appearing in the process, which could include axioms, theorems, and facts. From this axiomatic system, theorems can be derived logically. There are at least three ways to justify the established axiomatic system and the derived theorems: 1) use the derived theorems to represent design cases through progress of the design; 2) compare the derived theorems against the properties of the design from empirical studies; and 3) apply the derived theorems to manage and control the design process. All three ways will be used to test and justify the proposed theory in this thesis.

In accordance with the above research scheme, major research tasks include:

1) establish an axiomatic system of design;

2) derive theorems about the design from the established axiomatic system;

3) compare the derived theorems against existing knowledge of design properties;

4) carry out case studies; and

5) apply the theory to improve design practice.

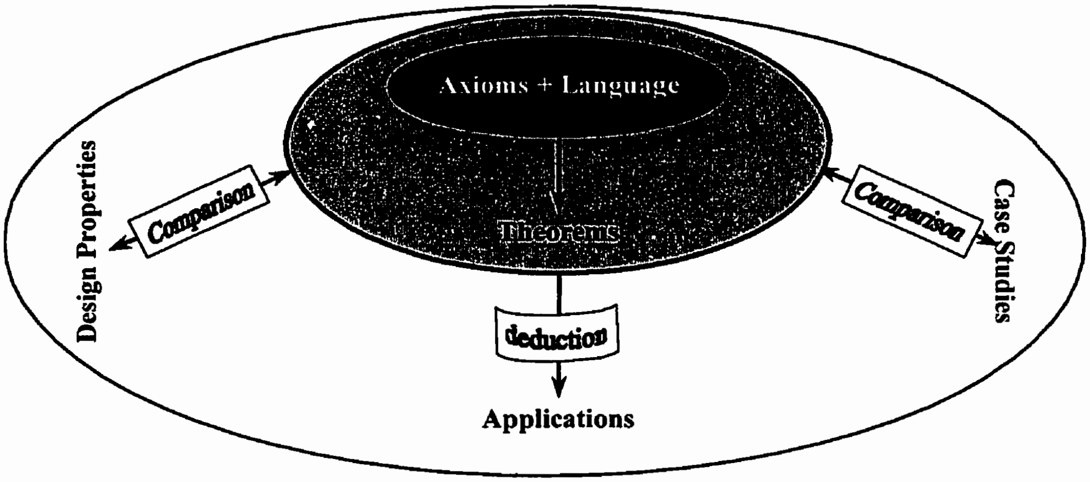


Figure 1-3 Research scheme

The first two tasks aim to establish the theory while the latter three test and verify the reliability and usefulness of the theory. The vision behind this approach is that if an axiomatic system can be established for a class of problems, then this class of problems can be studied in a logical way so that the solutions to the problems can be explored in more meaningful and robust approaches. This exploration could lead to results that are already known from other approaches. This is how an axiomatic theory should look. However, it opens the possibility that some unknown results may also be found. This axiomatic nature underlies most scientific systems.

1.3 Language of Research

Various means have been used to represent results in design studies. These include natural language, flow charts, graphic illustrations, as well as mathematical tools. Different merits can be gained for different means. Compared to other means, mathematical representation has the following merits:

- more justifiable, reliable, and accurate;
- it is the foundation of better and robust computer aided product design systems;
- once the mathematical approach succeeds in an area, the breakthrough in this area would be profound.

So far, many mathematical tools have been used to study design. Examples are shape grammar from formal language theory (Stiny, 1980; Schmidt and Cagan, 1996), general design theory and mathematical theory of design from set theory and topology (Yoshikawa, 1981; Tomiyama and Yoshikawa, 1985; Braha and Maimon, 1998), and design information model from axiomatic set theory (Salustri and Venter, 1992). The current research will use formal set theory to represent the axioms, theorems, and facts appearing in the axiomatic system. The objective of this thesis research can then be refined as:

To formulate and formalize the conceptual design process using the axiomatic method and to test the theory using case studies, design properties, and applications.

1.4 Specific Aims and Assumptions

A conceptual design process begins with design requirements and ends with product descriptions of design concepts, as is shown in Figure 1-4. Therefore, the formulation and formalization of conceptual design include three parts: 1) design requirements; 2) product descriptions; and 3) the design process.

Figure 1-4 Design activity

Correspondingly, the specific aims of this thesis research include:

Specific Aim 1: Establish an axiomatic system of product design. This includes the axioms about design and a mathematical language representing design requirements, design concepts, and design processes.

The following assumptions have been made to support this research:

Assumption 1.1: Product design is a phenomenon involving human beings as a part of the process. This phenomenon can be studied using scientific approach like other phenomena. The research objects include designers as well as products that designers create.

Assumption 1.2: Mathematics as a tool of representation and logical reasoning can be used to study phenomena existing in this world qualitatively as well as quantitatively.

Specific Aim 2: Derive theorems about product design from the established axiomatic system following logical steps. These theorems include the definition and nature of the design requirements, design concepts, and design processes.

Assumption 2: Theorems can be derived from an axiomatic system.

Specific Aim 3: Justify the derived theorems by comparing against design cases and existing knowledge about the nature and properties of product design.

Assumption 3: Any scientific theory (axiomatic theory) can be justified by comparing its theorems against facts.

Specific Aim 4: Develop a prototype of product design software system based on derived theorems.

1.5 Organization of the Thesis

To fulfill the tasks given in Section 1.4, the rest of this thesis includes nine chapters. Chapter 2 reviews the up-to-date research in the related areas and summarizes the widely accepted recognition of design properties. Chapter 3 discusses the axioms and mathematical language constituting the axiomatic system. Major theorems about product conceptual design are also derived in this chapter. Chapter 4 through Chapter 6 focus on the modeling of design objects and the design process as shown in Figure 1-4. Chapter 7 and 8 compare the theorems derived from the axiomatic system against two properties and observations about design problems and processes. Chapter 9 introduces a prototype software system of a simple mechanism design based on the derived theorems. Chapter 10 is the conclusion of this thesis. The logical relation of these chapters is illustrated in Figure 1-5.

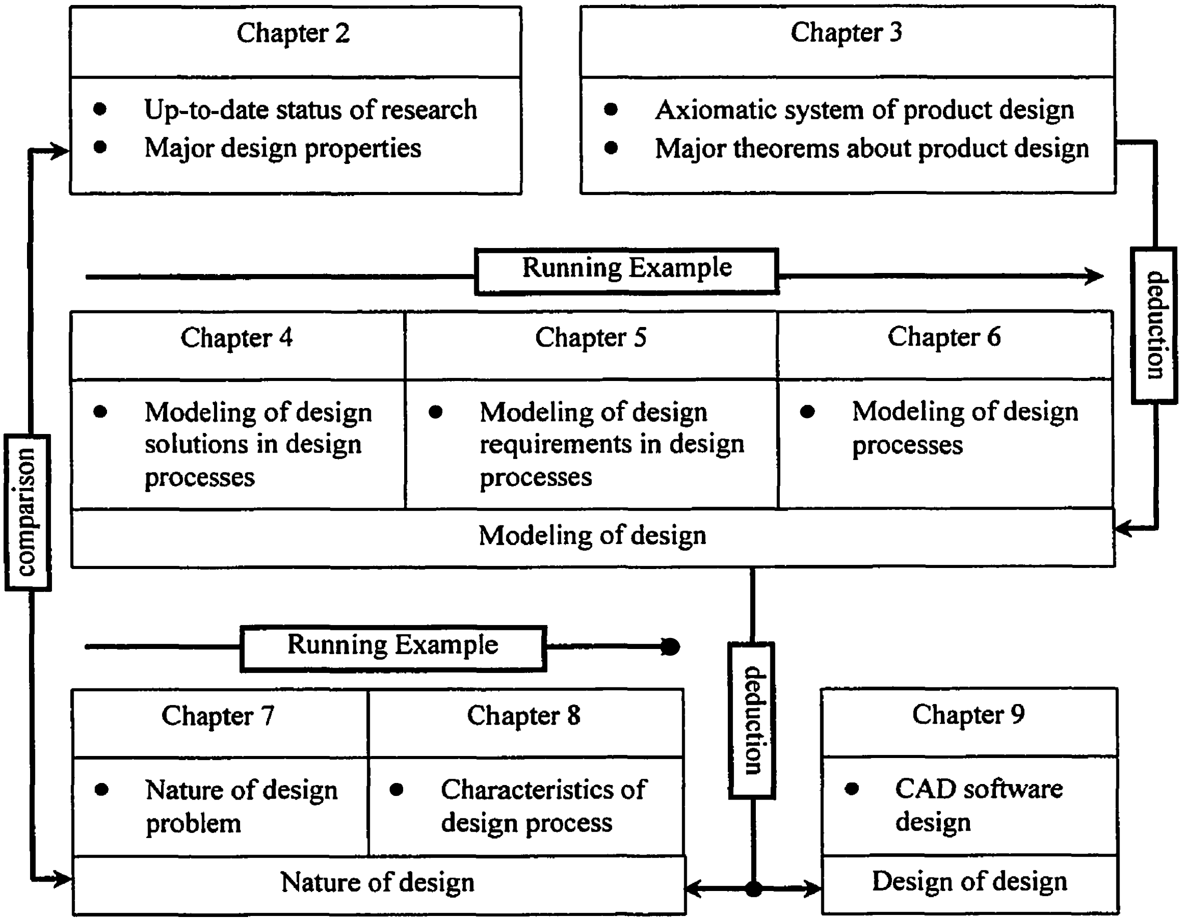


Figure 1-5 Logic flow of the thesis

The above diagram can be mapped into the structure given in Figure 1-3, which is shown in Figure 1-6.

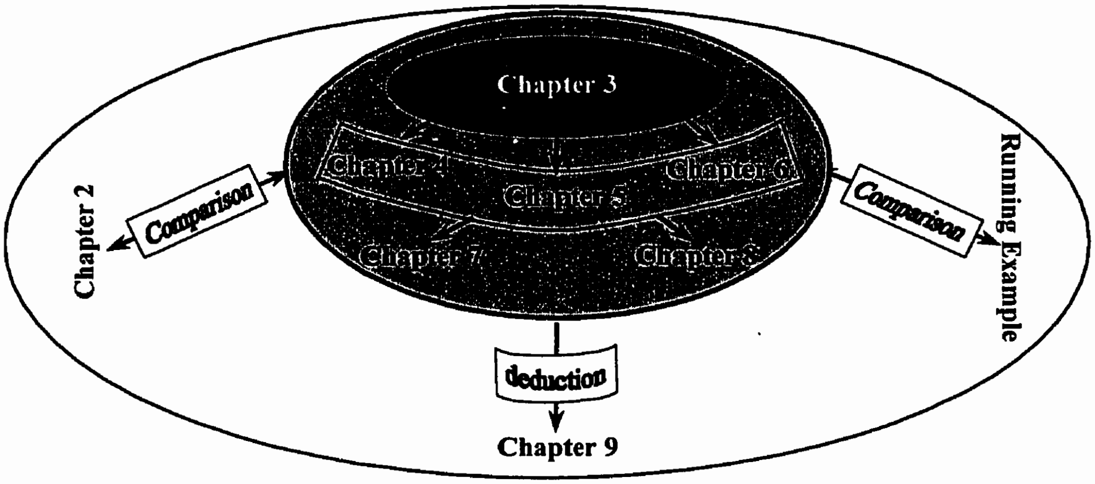


Figure 1-6 Relation of the thesis to the research scheme in Figure 1-3

Chapter 2 Literature Review

2.1 Introduction

Design is a basic human activity. During a long period, design had been taken as an 'art' which was taught through a 'master-student' model. Only after the 1960's did intensive studies into the design activity start as many researchers attempted to propose better design methodologies to improve design and design education. Systematic design methodology (Pahl and Beitz, 1988), axiomatic design (Suh, 1990), and computational design theory (Gero, 1997) are examples of achievements from this research. The research has been driving design from 'art' to 'art and science' (Kirschman *et al*, 1996). The science aspect allows people to better understand design processes while the art aspect allows designers to keep their creativity in rationalized design processes resulting from the scientific investigation. Generally speaking, they can be categorized into three schools: philosophy, theory and methodology, as well as application. It is shown in Figure 2-1. These endeavors aim to understand, simulate, and/or improve design from different perspectives.

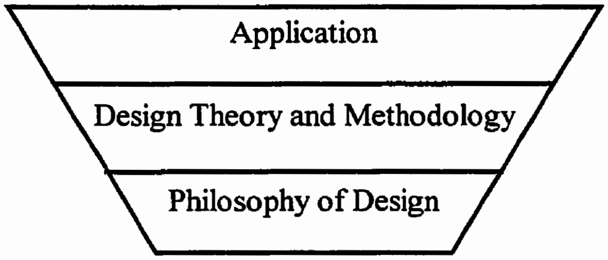


Figure 2-1 Schools of design research

In the school of philosophy of design, efforts have been made to investigate the nature of design from psychological, epistemological and cognitive perspectives. A key area of research in this aspect is design thinking (Cross *et al*, 1992), including its form (March, 1976; Zeng and Cheng, 1991; Roozenburg, 1992; Palle, 1995), its capability and limitation (Simon, 1969), and psychological foundation (Akin, 1979). These studies defined design problem, design objects, and design process at a philosophical level. More publications can be found in an international journal *Design Studies*. In the school of application, the major task is to apply the knowledge about design to improve design practice by developing computer aided design software systems, establishing robust and reliable design guidelines, proposing new curriculum for design education, and so on. There is a huge amount of literature in this area (such as Liang and O'Gray, 1998; Harmer *et al*, 1998; Xue, *et al*, 1999; Sun, *et al*, 2000; Zhang and Xue, 2001). The annual ASME conference on design engineering include many related publications. Since these two areas are not the major concern of this present thesis, the detailed review will not be included. The rest of this chapter will review the work in the school of design theory and methodology in detail, followed by a list of some common properties of product design.

2.2 Design Theory and Methodology

In the school of *design theory and methodology*, researchers have been attempting to formulate the results from design philosophy (Galle, 1996) or to induce general theories from design practice to establish a step-by-step design methodology (Wiegers, *et al*, 1996; Suma, *et al*, 1998; Knoop, 1997; Shah, 1998), as was shown in Figure 1-3. Cross (1993) classified the research into three groups: scientific design, design science, and the science of design.

Scientific design refers to modern, industrial design (as distinct from pre-industrial, craft oriented design) based on scientific knowledge but utilizing a mix of both intuitive and non-intuitive design methods. Through the application of scientific knowledge in practical tasks, design makes science visible (Willem, 1990). *Design science* refers to an explicitly organized, rational and systematic approach to design: not just the utilization of scientific knowledge of artefacts, but design also in some sense as a scientific activity itself. It includes two important areas of theory: theory of design process (general procedures, methods, tools) and theory of machine systems (classification, modeling, etc of technical systems) (Andreason, 1991). It has led to the attempts to formulate design methods based on formal languages and theories. *The science of design* refers to that body of work which attempts to improve our understanding of design through scientific methods of investigation.

Among the above three categories, scientific design falls into the category given in Figure 1-1, which has been the major goal of engineering research and practice. This is not the concern of this thesis. Publications can be found in the design within different subjects. Only the progress in design science and the science of design will be reviewed. Typical achievements in these two research areas include:

- Systematic design methodology;
- Decision-based design theory;
- AI-based design theory;
- Axiomatic design;

- Science-based design theory.

2.2.1 Systematic design methodology

The systematic design methodology prescribes step-by-step procedures including four stages: 1) product planning and clarification of task; 2) conceptual design; 3) embodiment design; and 4) detailed design (Pahl and Beitz, 1988; Hubka, 1980; Hubka and Eder, 1988). Each of these phases can be decomposed into a sequence of operations with specific objectives. The results of one operation becomes the input to the next operation. In the phase of product planning and clarification of task, the problem is formulated by identifying all customer requirements as a set of specifications. Design specifications are then abstracted into overall functions in the conceptual design phase. The overall function of the system is initially viewed as a "black box" that operates on materials, energy, and signals. After these overall functions are systematically decomposed into functional structures, solution principles are sought that can potentially fulfill the functional needs of each functional element (Liu *et al*, 1999). The functional element is then replaced by the sought solution principles. Many principles could be found for each functional element. Combinations of the most promising principles for each functional element should be enumerated and compared. The most promising design concepts are taken as the input for further embodiment design phase. During the embodiment design phase, the requirements that affect the physical embodiment of the solution are determined. These requirements include: safety, cost, environmental factors, ergonomic factors, manufacturability, operational procedures, maintainability, etc. Subjected to the requirements, combinations of embodiments are laid out to produce physical configuration models or layouts. The most promising configurations are refined and enhanced, resulting in a physical representation of the system, which is the product architecture (Stone, *et al*, 1999; Zamirowski and Otto, 1999). Because many potential architectures exist, these must be evaluated and compared (Finch and Ward, 1997). The best architecture becomes the input for the detailed design stage. The detailed design stage completes the design of each component within the product architecture, resulting in a complete description of the system. This procedure-based approach aims to improve design by providing guidelines for engineers to follow in the design processes (e.g., Coulter and Bras, 1999; Warell, 1999). The methodology has been extended to include Design for X (assembly, manufacturing, service and so on) (Dixon and Poli, 1995), quality function deployment (QFD) (which aims to ensure the quality throughout the design and manufacturing process), house of quality (HOQ) (which is a technique developed to capture the "voice of customer" and analyze design specifications and compare design solutions with benchmark products) (Clausing, 1994), and robust design (Taguchi, 1987).

Another major achievement in systematic design is the Theory of Inventive Problem-Solving (TIPS, also known as TRIZ, the Russian abbreviation), which was developed in the former Soviet Union by Altshuller (1988) and co-workers. Hashemian (2000) summarized the fundamental assumptions implied in this theory: 1) Invention is to find a good solution in the solution space. Traditionally this is done using the trial-and-error method. It is possible to increase the efficiency of this method using scientific rules that help avoid "empty" trials. 2) It is possible to reduce the complexity of an inventive problem to lower levels and use simple techniques to solve them. 3) Invention is done through analogy. Thus, the laws of development (invention) of technical systems can be obtained by observing previous inventions. 4) Any invention requires removing contradictions. These contradictions can be technological or physical. Contradictions happen when some improvement may cause a deterioration. 5) Objective laws must exist that tell when and what methods to use for the removal of contradictions. These laws can be developed through studying the methods of contradiction removal embedded in prior inventions.

Based on the above postulates, they analyzed a large number of patents (>2 million) in order to find general patterns. The analysis showed that most patents suggested means for eliminating system conflicts in a system. Such conflicts arise when a certain parameter cannot be improved without causing another to deteriorate. Furthermore, it was found that many inventions were based on the same underlying principles and physical, chemical and geometrical effects, and that only about 40 principles and a few hundred effects were used. The result was TIPS, which includes five parts: engineering systems evolution laws, algorithm for inventive problem-solving, patents collection, value-engineering analysis, and creative personal development.

In TIPS, eight laws have been identified. The most fundamental law is that of the ideal system. This law says that the development of a system proceeds towards an ideal system that provides the function without having a system (the ideal solution). In other words, in an ideal system, the given function is realized but no resources are consumed (Malmqvist, *et al*, 1996). Another law states that the S-Field, consisting of substances (S1, S2) and a field, is both necessary and sufficient for the minimal description of a technical system. This helps to decide which standard to apply so that undesired, insufficient or missing interactions are eliminated. Other laws include the law of minimum completeness which says that the system must fulfill the minimum requirements that bring the system to life. If any of these requirements fails, the whole system fails. Further, it was stated that systems tend to include a larger number of functions over time. Moreover, systems are originally conceived on a macroscopic level (machine elements) after which a transfer to a microscopic level takes place. Here, the function is realized by molecules rather than machine elements.

The Algorithm for the Solution of Inventive Problems (ASIP) is the practical formulation of TIPS. The stated aim of this algorithm is to eliminate the conflict while making minimal changes to the system. It is very straightforward. The main steps are given in Figure 2-2.

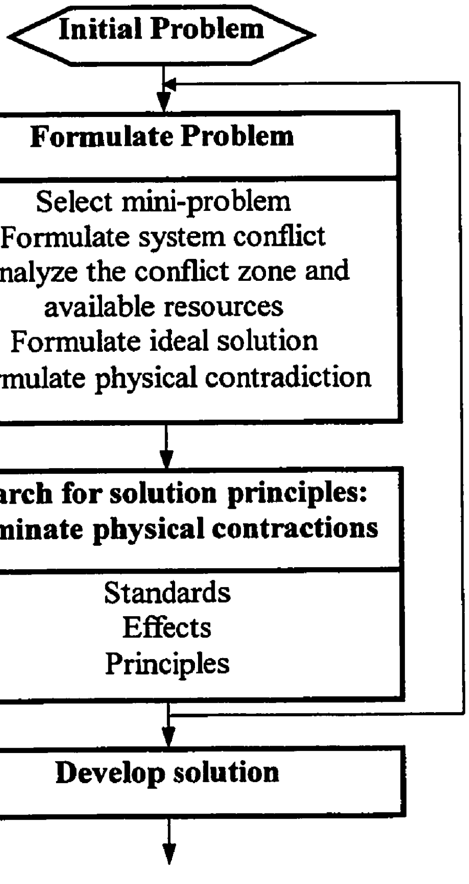


Figure 2-2 Algorithm for inventive problem-solving (adapted from Fey *et al*, 1994)

Malmqvist, *et al* (1996) compared the theory of inventive problem solving and the systematic approach of Pahl and Beitz from 14 different aspects including scope, task clarification, problem formulation, solution generation methods, function vocabulary, creative methods, solution space, product models, principles, knowledge base, evaluation, learning time, and computer system. Based on this comparison, both approaches were unified.

2.2.2 Decision based design theory

Decision-based design is a methodology that uses the rules of decision theory and its related science in design. Related headings of this area of science include Utility Theory, Multi-Attribute Decision Theory, Game Theory, Information Science, Analytical Hierarchy Process, etc. It views the designing as a decision-making process, which includes a series of decisions. Decision sciences and their application in design have been extensively discussed in literature for many years (e.g., Tribus,1969; Hazelrigg 1996). Except the generation of design space and design candidates, main steps in this design process involve either selection of decisions or compromise of decisions (trade-off) (Allen and Mistree, 1997). Design decision support systems have thus been developed to assist engineers in making better decisions in evaluating several alternatives and selecting the best one.

An important aspect of decision-based design is that it replaces the artifact-oriented design process with decision-oriented process that delivers the artifact. In the latter process, the final design proposals depend more on the decisions made by human being. The decisions could be different from person to person and from time to time. The study in decision-based design would enhance the quality of this decision making process and in turn improve design practice.

In terms of the nature of design, decision based design mainly: 1) attempts to offer tools and techniques to deal with conflict resolution and multi-objective optimization, since design decisions are usually made based on multiple criteria, which are often in conflict with each another (Haroud *et al*, 1995; Bahler *et al*,1995; Otto and Antonsson, 1991; Brazier *et al*, 1995; Reddy *et al*, 1996); 2) develops quantitative measures for the qualitative parameters so that they can be evaluated especially in early design stages (Finch and Ward, 1997); 3) deals with ambiguity, risk, and uncertainty in human decision making processes (Wan and Krishnamurty, 1999); 4) proposes approaches for distributed decision making to model the collaborative design processes (Oh and Sharpe, 1995).

2.2.3 AI-based design theory

There are three major issues in AI-based design research. The first is modeling multiple facets of mechanical products. These include the representation of products in different levels. Design knowledge is another issue (Sainter *et al*, 1998). The third problem is how to generate and select appropriate means of mapping the user's requirements to some physical structures that can realize the given set of requirements (Hsu and Woon, 1998).

In modeling mechanical products, the representation should be able to describe the products at different design stages. In the conceptual design phase, products are mostly defined by linguistic information and sketches. In the detailed design, geometric information with the detailed technical specifications is dominant. Typical representation schemes for conceptual design include language and object. Language attempts to represent design in a formal way. It provides an unambiguous means to describe the design. Mullin and Rinderle (Mullin and Rinderle, 1991; Rinderle, 1991) used a graph-based language to describe behavioral specifications of design as well as the behavior of the components. Vescovi *et al*(1993) developed a language, CFGL, for specifying the causal functionality of engineered devices. In terms of grammar, Stiny(1980) used shape grammar to represent physical design forms.

A recent trend is the application of object-oriented techniques. The emphasis of the object-oriented design is decomposition and representation. Decomposition is the breakdown of the functions of a product so that the lowest level of the function structure consists exclusively of functions that cannot be subdivided further. Decomposition differentiates object-oriented design from conventional structured design. Representation is concerned with defining an object and organizing objects. It is generally thought that a good representation leads to a successful object-oriented design (Liang and O'Grady, 1998). Work done on object-oriented design systems includes that by Kusiak *et al*(1991), Ohki(1994), Segapeli and Cavarero (1996), Gorrti *et al* (1998), and Liang and O'Grady(1998).

The approaches to representing design knowledge are closely related to the reasoning scheme in modeling the design process. The modeling and representation of knowledge are most important to knowledge-based design. Different types of knowledge are modeled differently. The fundamental research issue is to determine what are appropriate representation frameworks for different types of design knowledge. Generally, knowledge-based design can be classified into model-based design, rule-based design, and case-based design. Model-based design relies on a qualitative model of causal relationships among all the concepts used for representing the design object (Sekiya *et al*, 1998). The rule-based design paradigm has been adopted by many researchers to provide advice on design solutions (e.g, Rao and Prakasa, 1992). But the structure of rule-based system is usually too simple for design problem solving, it is generally combined with other approaches. Case-based design has been successfully applied to the problems where the structure and content of design information can be encoded symbolically. Examples of the work in this field include KRITIK(1992), Li *et al*(1996), Mostow *et al*(1992). In any case, artificial neural network may be applied to support the knowledge acquisition processes (Sun *et al*, 2000).

More publications on AI-based design can be found in the Annual ASME Design Engineering Technical Conferences.

2.2.4 Axiomatic design

Suh proposed the concept of axiomatic approach in the late 1970s, which reached maturation around 1990 (Suh, 1990). The theory perceives the design process as mapping between four domains: customer domain, functional domain, physical domain, and process domain. The vectors associated with each domain are customer attribute (CAs), functional requirements (FRs), design parameters (DPs), and process variables (PVs). Two axioms were identified for presenting a good design:

- The Independence Axiom: Maintain the independence of functional requirements.
- The Information Axiom: Minimize the information content that satisfies the functional requirements.

According to the axioms, the idea design provides a linear mapping between design domains. Although most engineering designs are actually coupled complicatedly, Suh's axiomatic approach is likely to provide the conceptual framework to many other approaches, such as Quality Function Deployment (mapping from CAs to to FRs), Feature-Based Design (mapping from FRs to DPs), and Computer Aided Process Planning (mapping from DPs to PVs) (Zhu and Kazmer, 1999; Yu *et al*, 1998).

In developing and applying the axiomatic design theory, Rudolph(1996a,b) provided a theoretical framework and proof to verify the above two axioms. Jahangir and Frey(1999) proposed differential entropy to measure the information content. El-Haik and Yang used the information axiom to measure the complexity in design (1999). Johnnesson (1996, 1997) attempted to solve the problem of functional coupling in configuration design based on Suh's axioms. Marston *et al*(1997) proposed a model for variant design by combining axiomatic design with decision-based design.

The research in this area can be seen as the endeavor in design science, which views design as similar to any other scientific subjects. The solving of design problem can follow axioms logically. There are other axiomatic approaches in design, such as Yoshikawa (1981), Salustri and Venter (1992), as well as Maimon and Braha (1996). But these are not the axiomatic approaches for solving design problems, but are axiomatic approaches to representing and studying design problems. They will be reviewed at the beginning of Chapter 3.

2.2.5 Science-based design theory

Despite the ever-increasing literature in the research of design, the investigation is by no means concluding and profound. As was shown in Figure 1-2, the breakthrough in design methods, which supports design activities, depends on more scientific exploration of design. This exploration attempts to improve the understanding of design processes across engineering disciplines by identifying the common elements and disclosing the underlying order of design processes so that fundamental principles and theories can be established. However, it is still in pre-theory stage(Shah and Hazelrigg, 1996; Gu, 1998).

As the name implies, science-based design theory aims to establish some fundamental principles and theories for engineering design. Like any of other engineering sciences, the science of design should include two fundamental parts: laws and languages. The laws can be axioms and/or postulates which establish basic assumptions and principles for design processes. The languages provide a basic means to represent the laws. The existing theories reviewed here will be evaluated from these two aspects wherever appropriate.

The investigation of the basic laws is usually accomplished philosophically, psychologically, and/or experimentally. The work and research progresses accomplished by Simon(1969), Yoshikawa(1981), Suh(1989), Takeda(1994), and Marston *et al*(1997) are examples of the former and that done by Umeda *et al*(1990), Salustri and Venter(1992), Eastman and Fereshetian(1994), and Gui and Mantyla(1994) are the examples of the latter. They have focused more on data modeling. Maimon and Braha(1996) tried to put two parts together from computational point of view. Gorrti *et al*(1998) proposed an objected-oriented representation for product and design processes. However, in most of the existing research, the proposed languages often do not sufficiently support design processes and most of existing endeavors were trying to realize computational design, and thus limited by the concept "computational" to some extent, where natural and graphic languages are the major representation tools in the research(Hsu and Woon, 1998). The mathematical formulation of the law is always essential for the purpose of scientific enquiries of design. (Yoshikawa, 1981; Tomiyama and Yoshikawa, 1987; Braha and Maimon, 1998).

2.3 Common Design Properties

Based on the design research in the last several decades, the design research community has recognized some common properties of design problems and design processes. These properties can be used as empirical criteria to test an established design theory. If a theory leads to conclusions contradicting the properties, this theory should be rejected. Since many of these properties have been so widely discussed and used that it is difficult to identify the original authors in most cases. The references will be given only when it is possible.

Property 1: Design is an activity that begins with an acknowledgement of design requirements and ends with a development of product descriptions that satisfy the design requirements. It is illustrated in Figure 1-6.

The design requirements can be motives or demands for a completely new product, the complaints on the performance of existing products, or the failure due to malfunctions of existing products. Product descriptions define intermediate and final design results. They can be graphic, linguistic, and/or mathematical. The product descriptions will be evaluated against the design requirements. The design process then provides a mapping from design requirements to design descriptions.

Corresponding to this property, Yoshikawa's general design theory was illustrated as a mapping from function space to attribute space in Figure 2-3 (Yoshikawa, 1981).

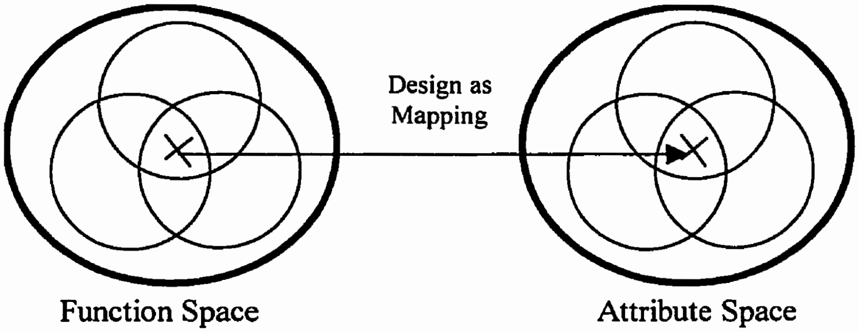


Figure 2-3 Design process in ideal knowledge (Yoshikawa, 1981)

Property 2: Design is an evolving process which can be considered as a transition from abstract concepts to concrete descriptions(Suh,1991). This hierarchical nature of design can be illustrated in Figure 2-4.

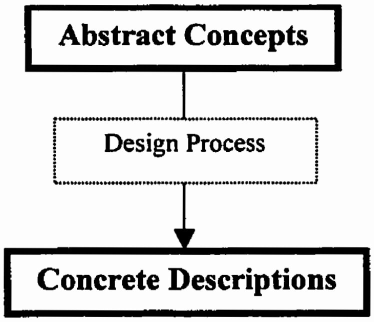


Figure 2-4 Evolution nature of design process

Property 3: A basic design process exists for all design problems. It is illustrated in Figure 2-5. It includes two major processes: synthesis and evaluation. According to given design requirements, candidate design descriptions are generated in the synthesis stage. Then the product descriptions and the corresponding product performances are evaluated against the design requirements to determine if the designed product satisfies the requirements in the evaluation stage. The process iteratively generates more and more concrete designs.

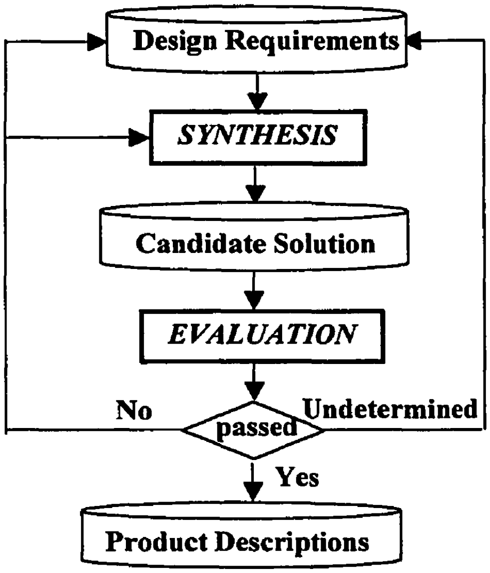


Figure 2-5 Basic design process

By considering the evaluation process, Tomiyama and Yoshikawa(1985) extended general design theory to include product performance. They proposed a design process model in real knowledge in Figure 2-6.

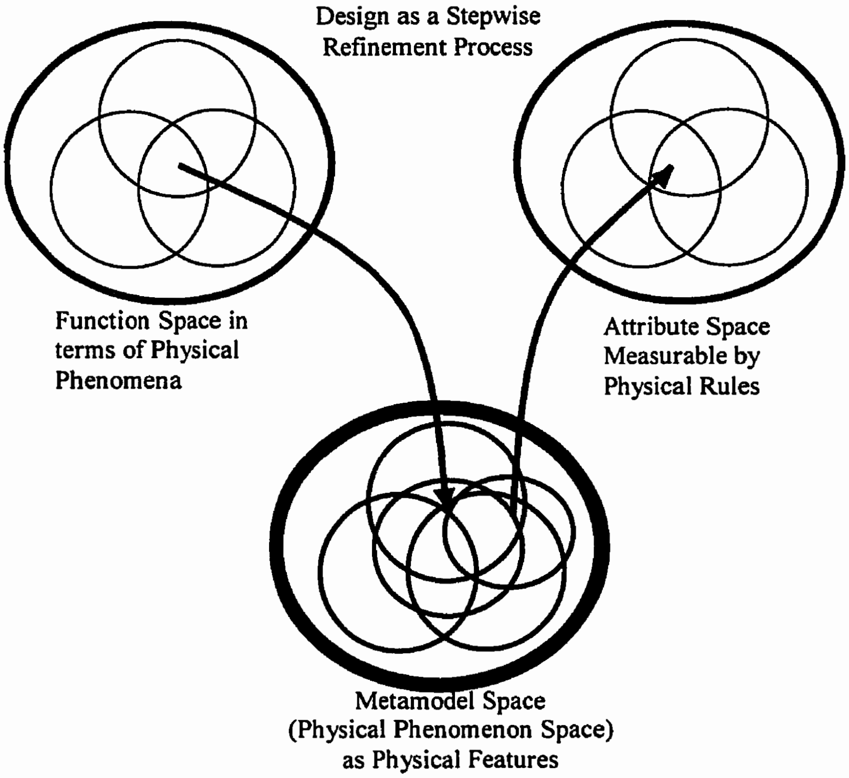


Figure 2-6 Design process in real knowledge (Tomiyama,1985)

Braha and Maimon (1998) took into account the synthesis process and presented a new assumption on design process, based on which they established a mathematical theory of design. Their scheme of design is shown in Figure 2-7.

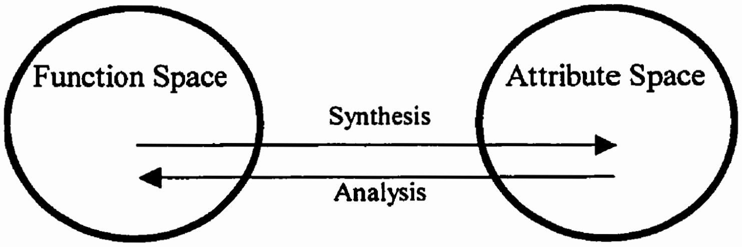


Figure 2-7 Design analysis and synthesis (Braha and Maimon,1998)

Property 4: The design requirements may initially not be precise or complete; hence, the development and elaboration of design requirements becomes an integral part of the design process. An intermediate design result often intrigues new design requirements and refines the original design problem. This is called the ill-structured characteristics of design. It is illustrated in Figure 2-8.

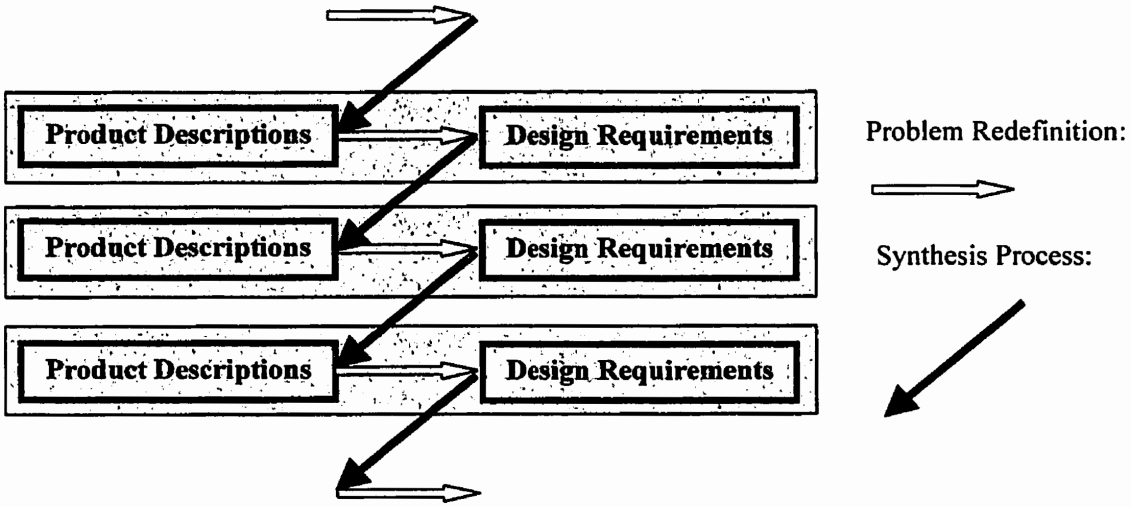


Figure 2-8 Ill-structured characteristics of design

Obviously, this property comes from the combination of the first three by taking into account the refining process of design problem. Evaluation is a sub-process of problem redefinition.

In traditional problem solving, an initial formulation of the problem and the requirements for solution are usually given, an approach is often developed to find the solution(s). In the process, the problem formulation is not changed. This kind of problem is said to have well defined structure. From the preceding discussions, the design problem itself is always changing in the process of finding solution. The change depends on the intermediate solutions. This is the so called ill-structured problem. The detailed discussion of the ill structure property of design problem can be found in Simon (1969,1971,1973).

Property 5: Design alternatives are not provided in advance in design problem solving. They must be found, developed, and synthesized by an exploration process, which is iterative and evolutionary in nature. It cannot be expected to have a deterministic solution for most design problems. There are no such things as *well defined* initial state, goal state, and state space for design problem solving.

Property 6: The designer is constantly faced with the problem of bounded rationality. In the model of bounded rationality it is self-evident that there are limitations on the cognitive and information processing capabilities of the designer's decision making. Traditional engineering design methods make much more use of satisfying criteria rather than optimal specifications (Simon, 1969, 1982).

Property 7: Design is a creative act. One designer may get different solutions when s/he does a same design in different time. Different designers may reach different solutions for the same design problem. Creative design is a random process.

Though there are much more properties about design, only the above seven are taken as they are directly related to the present research. The rest of the thesis will take these properties either as axioms (Property 6) to derive the theorems about design or criteria to test and justify the theorems.

2.4 Concluding Remarks

This chapter classifies design studies into three categories: philosophy of design, design theory and methodology, as well as applications. The research in design theory and methodology is reviewed in five subheadings: systematic design methodology, decision-based design theory, AI-based design theory, axiomatic design, and science-based design theory. Seven design properties are summarized for the purpose of deriving, testing and justifying the axiomatic system in this thesis.

Chapter 3 An Axiomatic Approach to Studying Product Design

3.1 Introduction

An axiomatic theory is a deductive theory. It is composed by a set of statements called axioms or postulates, from which conclusions or theorems can be logically derived. Everything that is not an axiom or a theorem is excluded from the theory. If a verified fact is excluded from a theory then the theory needs to be evolved.

To formulate an axiomatic theory, a formal language is required to represent the axioms and derived theorems. This chapter aims to establish such an axiomatic theory for product design. The core of this theory is two axioms about product design and a mathematical language to represent the axioms and theorems. Theorems about product design will be derived in this system following logical steps. The verification and meaning of these theorems will be discussed in Chapter 4 through Chapter 8.

In design community, there have been major attempts in using axiomatic approach to:

- study design,
- accomplish design, and
- represent design.

The task of the first category is to establish a proper axiomatic system so that design activity can be well understood from the theory. The second category aims to develop some axiom-like criteria for design, by following which good design solutions can be generated. The third one is on the dimension of data and object modeling. In whichever category, different settings of axiom(s) and representation languages lead to completely different axiomatic systems and theories. The objective of this research is to find out the best combination of axioms and representation languages, with which more meaningful and accurate theorems about product design can be logically derived.

A pioneer work in axiomatic approach to studying design was initiated by Yoshikawa in his general design theory (Yoshikawa, 1981). In that theory, Yoshikawa proposed three axioms: axiom of recognition, axiom of correspondence, and axiom of operation. These axioms defined an idealized design performed by a superman who knows everything in the real world perfectly. Topology was used as a tool to represent the theorems. By recognizing the impossibility of perfect design knowledge which solves any design problem deductively, Tomiyama and Yoshikawa(1985) developed the work into an extended general design theory. The extended theory includes a metamodel as the interface between design function and product attributes. This metamodel is supposed to embody the physical laws in real world. Since the work is too abstract and idealistic, it has not found real practical applications. Based on the above research, Tomiyama(1994) has moved to the development of a knowledge-intensive design theory which is more in the area of knowledge-based design.

In the same category, Maimon and Braha (1996) tried to deal with the more philosophical studies about design, or in general the science of artificial, proposed by Simon(1969). They also used set theory and topology. The work is more focused on the computational aspect of design.

Suh(1989) proposed to accomplish a design task following his two design axioms: independence axiom and information axiom. The independence axiom suggests maintaining functional independence. The information axiom is used to select the design candidate with the least uncertainty among design alternatives. A design with the highest probability and the lowest complexity in meeting functional requirements has the minimal information content. These two axioms are used as evaluation rules and guidelines in assisting designers in design processes.

Salustri and Venter(1992) attempted to develop a design information theory based on axiomatic set theory. Their work is a subset of axiomatic set theory by adding constraints from engineering design. Design processes were not their concern.

The present thesis is another attempt in studying design using axiomatic approach. The main difference of this approach from other similar attempts is the axioms used in the theory. The following sections will discuss: language, axioms, and theorems of the axiomatic system for product design.

3.2 Language

Basically, set theory will be used to represent the axioms and theorems in this axiomatic system. However, the following definitions are given for the specific aim of design research.

Definition 3-1 Object

Object is an entity existing in environment.

The definition of environment will be given in Sections 3.3and 3.4.

Definition 3-2 Properties of Object

Property is an observable, measurable or otherwise known characteristic related to an object (Salustri and Venter, 1992). All properties in environment constitute a property set

$$\mathbf{X} = \{\mathbf{x}_i : \mathbf{i} = 1,2,\ldots\} \qquad (3\text{-}1)$$

where $\mathbf{x}_i$: a property;

$\mathbf{X}$: property set.

A property is defined by its name $\mathbf{x}^n$ and its value $\mathbf{x}^v$. If the sets of property name and property value are denoted as $\mathbf{D}^x$ and $\mathbf{R}^x$, respectively, then a property can be further represented as

$$\forall \mathbf{x}_i \in \mathbf{X},\ \mathbf{x}_i = <(\mathbf{x}_i)^n, (\mathbf{x}_i)^v>, (\mathbf{x}_i)^n \in \mathbf{D}^x, (\mathbf{x}_i)^v \in \mathbf{R}^x \qquad (3\text{-}2)$$

$\mathbf{D}^x$ and $\mathbf{R}^x$ can be taken as the domain and the range of a property, respectively. Therefore, a property set $\mathbf{X}$ can be defined as

$$\mathbf{X} \subseteq \mathbf{D}^x \times \mathbf{R}^x \qquad (3\text{-}3)$$

Definition 3-3 Object Description

Any object in environment is recognized by human beings through its properties (adapted from Yoshikawa, 1981; Salustri and Venter, 1992). An object description is a set of properties

$$\mathbf{O} = \{\mathbf{x}_i : \mathbf{i} = 1,2,\ldots \mathbf{n}_o, \mathbf{x}_i \in \mathbf{X}\} \qquad (3\text{-}4)$$

where **O**: an object;

$\mathbf{x}_i$: a property defined by Equation (3-2);

X: property set defined by Equation (3-1);

$\mathbf{n}_o$: number of properties defining an object.

Definition 3-4 Structure Operator

Structure operator is a unary operator ⊕ that satisfies the following condition:

$$\forall \mathbf{O},\ \oplus\mathbf{O} ::= \mathbf{O} \cup (\mathbf{O} \times \mathbf{O}) \qquad (3\text{-}5)$$

⊕**O** is read as "Structuring **O**" or "the structure of **O**". "::=" means "defined by".

3.3 Axioms

Human beings deal with two worlds: objective and subjective. The subjective world is the human mind. It includes knowledge, imagination, rationality, and so on. The objective world is the natural being. Design is a human activity to create artifacts, based on the understanding of these two worlds through the subjective world. Artifacts may include products, processes, and systems. The created artifacts in turn become a part of the objective world. The scheme is shown in Figure 3-1.

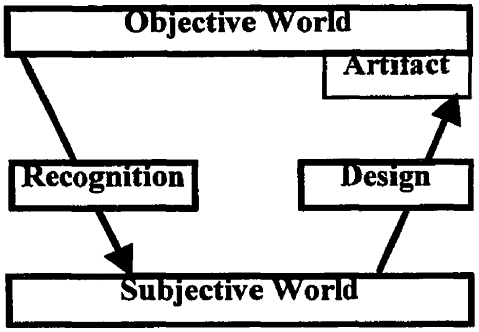


Figure 3-1 Human activities

In this thesis, the focus is on product. In consistency with terminology from product design community, the word "objective world" will be replaced by "environment", "subjective world" by "human understanding", and "artifact" by "product". The new scheme is then shown in Figure 3-2. It is called human-product-environment system, which has three objects: environment, product, human being, and two processes: recognition and design. In this system, human being plays two roles, one as a part of environment, and another as an agent that could understand environment including himself by standing out of the environment.

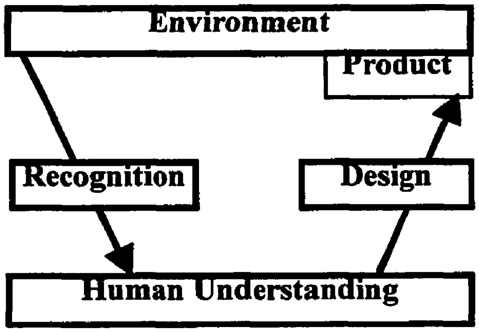


Figure 3-2 Human-product-environment system

In the system shown in Figure 3-2, environment is the whole universe. Recognition is the process through which human beings develop an understanding of the environment. Design is the process through which human beings create product into the environment to serve for prescribed purposes. The recognition precedes the design. For products to function properly in their environment, they must obey the physical laws in the environment. Therefore, the human understanding of environment as well as human being themselves as a part of environment is essential for a design to succeed. As a result, the nature of the recognition determines the nature of the design. If we have perfect recognition and in turn perfect knowledge of the environment and the design requirements, then we will have no difficulty in solving any problems at all. This was one of Yoshikawa's axioms in his general design theory (Yoshikawa, 1981). That axiom was called the axiom of correspondence which states that "the entity set and the entity-concept set have one-to-one correspondence." It assumes the existence of a superman who knows everything in the real world (Tomiyama, 1994). The design by this superman is finished immediately after design requirements are specified.

Design becomes problematic because the recognition is problematic in that human designers are not supermen and can never have the perfect understanding of environment. This constitutes the first axiom of the current axiomatic system of product design.

Axiom 1. Axiom of bounded rationality

Human beings are bounded in rationality (adapted from Simon, 1969).

This axiom is from Simon's study in organizational behavior (1969). It may have many implications. Typically, it could include:

- Causality is not symmetric.

The relation from cause to effect is deterministic in nature. However, any other forms of reasoning are plausible.

A simple example is given in Figure 3-3. The cantilever beam with a force **F** exerting at its right end could have a displacement δ. With the force and material properties given, δ is solely determined. However, if we know the displacement δ, we would not be able to determine forces on the beam and many other factors.

Figure 3-3 Causal relation

This fact of causal relationship can be more formally described as

$$\begin{array}{l} \forall \mathbf{s}, \forall \mathbf{R}, \exists \mathbf{A}, \exists \mathbf{K}, \mathbf{K} : \mathbf{A} \rightarrow \mathbf{R} \\ \textbf{K is a one - to - one correspondence} \\ \overline{\mathbf{K}} \textbf{ is not necessarily a one - to - one correspondence} \end{array} \tag{3-6}$$

where **A**: action on the product;

R: response from the product;

K: performance knowledge of the product;

L: all physical laws regarding the product;

$\overline{\mathbf{K}}$: the complement set of **K** with reference to the set **L;**

s: a product.

- Human beings can only get partial knowledge about the environment. It indicates a fundamental nature of human beings in rationality, which comes from the limitation of human recognition ability.

$$\mathbf{K} \subset \mathbf{L} \tag{3-7}$$

This essentially states that human recognition is not perfect and it is impossible for human being to know all laws governing the world. For the cantilever beam in Figure 3-3, the causal relationship between force **F** and displacement δ may be known, but the causal relation between the molecule structure and displacement δ could be unknown.

- Information might not be accurate.

$$\begin{gathered} \forall \mathbf{s}, \forall \mathbf{R}, \exists \mathbf{A}, \exists \mathbf{K}, \mathbf{K} : \mathbf{A} \rightarrow \mathbf{R} \\ \forall \mathbf{s}, \forall \tilde{\mathbf{R}}, \exists \tilde{\mathbf{A}}, \exists \tilde{\mathbf{K}}, \tilde{\mathbf{K}} : \tilde{\mathbf{A}} \rightarrow \tilde{\mathbf{R}} \\ \left\| \mathbf{K}, \tilde{\mathbf{K}} \right\| \geq \mathbf{0}, \left\| \mathbf{A}, \tilde{\mathbf{A}} \right\| \geq \mathbf{0}, \left\| \mathbf{R}, \tilde{\mathbf{R}} \right\| \geq \mathbf{0} \end{gathered} \tag{3-8}$$

where **A**: real action on the product;

R: real response from the product;

K: real performance knowledge of the product;

$\tilde{\mathbf{A}}$: recognized action on the product;

$\tilde{\mathbf{R}}$: recognized response from the product;

$\tilde{\mathbf{K}}$: recognized performance knowledge of the product;

||**X**,**Y**||: the distance between the set **X** and the set **Y**.

Still for the cantilever beam, the measurement of force and materials might not be accurate, so the final displacement δ can not be precisely obtained.

An obvious manifestation of this axiom in design is the application of safety factors.

The above design axiom has embodied the nature of human being in rationality, which is the lower part of the scheme in Figure 3-2. The focus will then be moved to product and environment in the scheme. The nature of object representation will be discussed.

Axiom 2. Axiom of Object Structuring

An object can be structured as a class of other objects with relations among these other objects.

$$\begin{aligned} &\forall \mathbf{O}_n, \mathbf{O}_n = \mathbf{O}^1_{n+1} \cup \mathbf{O}^2_{n+1} \cup \cdots \cup \mathbf{O}^m_{n+1} \\ &\oplus \mathbf{O}_n = \mathbf{O}_n \bigcup \mathbf{G}_n \\ &\mathbf{G}_n \subseteq \mathbf{O}_n \times \mathbf{O}_n \end{aligned} \tag{3-9}$$

where $\mathbf{O}_n$: an object;

$\oplus\mathbf{O}_n$: structure of $\mathbf{O}_n$;

$\mathbf{O}^i_{n+1}$: an element included in $\mathbf{O}_n$;

$\mathbf{G}_n$: the relations existing within object $\mathbf{O}_n$.

3.4 Theorems of Product Design

In this section, we are going to derive theorems about product design from the axioms and language given in Sections 3.2and 3.3. These theorems include: constitution of product-environment system, definition of product performances, partition of product properties, structure of product and environment, design requirements, design evaluation, design synthesis, and design governing equation.

3.4.1 Constitution of product-environment system

Definition 3-5 Product

Product is an object created by humans according to their knowledge of the environment. Once a product is created, it becomes a part of the environment.

Definition 3-6 Product Environment

Product environment is defined as universe excluding the product. It is categorized into: direct, close, and remote environment. Direct product environment immediately affects the product, while the remote product environment has little or no effect on the product. Close product environment is something in between. (adapted from Hubka and Eder, 1988).

Definition 3-7 Product-environment system

In product engineering context, the overall environment (universe) can be divided into two parts: product and product environment. Both together constitute a product-environment system.

If we denote the overall environment (universe), product environment, and product as **U**, **E**, and **S**, respectively, then

$$\mathbf{U} = \mathbf{E} \cup \mathbf{S} \tag{3-10}$$

According to Axiom 2, the structure of the overall environment can be represented as

$$\oplus \mathbf{U} = \oplus(\mathbf{E} \cup \mathbf{S}) \tag{3-11}$$

In terms of Equation (3-5), we have

$$\begin{aligned}\oplus \mathbf{U} &= \oplus(\mathbf{E} \cup \mathbf{S}) \\ &= (\mathbf{E} \cup \mathbf{S}) \cup ((\mathbf{E} \cup \mathbf{S}) \times (\mathbf{E} \cup \mathbf{S})) \\ &= (\mathbf{E} \cup \mathbf{S}) \cup (\mathbf{E} \times \mathbf{E}) \cup (\mathbf{S} \times \mathbf{S}) \cup (\mathbf{E} \times \mathbf{S}) \cup (\mathbf{S} \times \mathbf{E}) \\ &= (\mathbf{E} \cup (\mathbf{E} \times \mathbf{E})) \cup (\mathbf{S} \cup (\mathbf{S} \times \mathbf{S})) \cup (\mathbf{E} \times \mathbf{S}) \cup (\mathbf{S} \times \mathbf{E}) \\ &= (\oplus \mathbf{E}) \cup (\oplus \mathbf{S}) \cup (\mathbf{E} \times \mathbf{S}) \cup (\mathbf{S} \times \mathbf{E})\end{aligned} \tag{3-12}$$

The relation represented in Equation (3-12) includes four terms, as is shown in Figure 3-4.

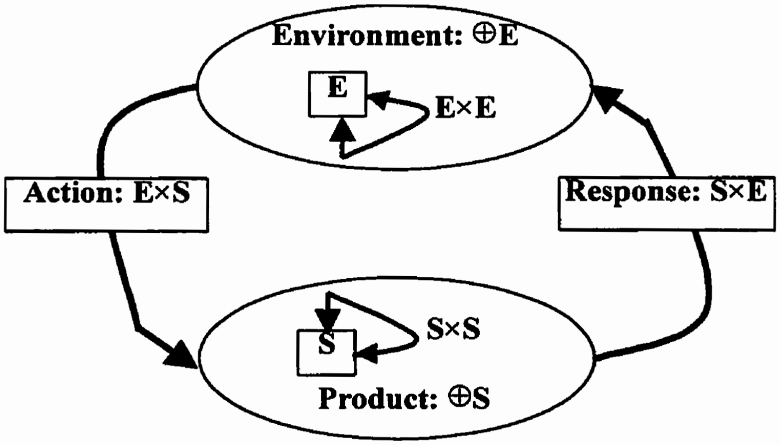


Figure 3-4 Product-environment system

In Figure 3-4, the symbols have the following physical meanings:

S:	product description;
$\oplus$**S**:	product structure. It represents relations within a product;
E:	environment description;
$\oplus$**E**:	environment structure. It represents relations within an environment;
E×**S**:	action imposed on the product **S** by environment **E**;
S×**E**:	response of the product **S** to environment **E**.

Equation (3-12) can be stated as the following theorem:

Theorem 3.1 A product-environment system is composed by product structure, environment structure, and the mutual interactions between product and environment. Both product and environment can also be structured by other objects in terms of Axiom 2.

The elements appearing in Equation (3-12) will be further defined in sections 3.4.2 and 3.4.4. The detailed factual meaning of the equation will be discussed in Chapter 4.

3.4.2 Definition of product performance

Figure 3-4 can be simplified as Figure 3-5. It actually defines product performance.

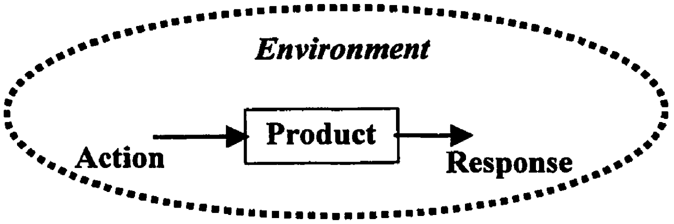


Figure 3-5 Product performance in environment

Theorem 3.2 Product performances are responses of a product to external actions from its environment. The relations from environment to product are external actions on the product. The relations from product to environment are responses of the product to its environment.

$$\begin{aligned} &\mathbf{P} \subseteq \mathbf{A} \times \mathbf{R} \\ &\mathbf{A} \subseteq \mathbf{E} \times \mathbf{S} \\ &\mathbf{R} \subseteq \mathbf{S} \times \mathbf{E} \end{aligned} \tag{3-13}$$

where **P:** product performance

A: action on product;

R: response from product;

E: environment;

S: product description.

In the above theorem, a new term, product performance, is introduced. This is not included in Equation (3-12). The following gives the proof how it can be derived from Equation (3-12).

A Cartesian product **A×B** can be defined as a set of ordered pair **<a,b>**, where **a** and **b** belong to sets **A** and **B**, respectively. Symbolically, it can be written as:

$$\mathbf{A \times B ::= \{< a, b > | a \in A, b \in B\}}$$

Ordered pair **<a,b>** is a primitive notion in set theory which satisfies the following property:

$$\mathbf{< a, b > = < c, d > \leftrightarrow a = c \wedge b = d}$$

The following definition of ordered pair is then adopted:

$$\mathbf{< a, b > = \{\{a\}, \{a, b\}\}}$$

Based on the above definition, it can be proved that for $\mathbf{a \in A}$ and $\mathbf{b \in B}$, $\mathbf{< a,b > \in A \times B \subseteq \wp\, \wp(A \cup B)}$.

Since $\mathbf{A \cup R}$ is defined in Equation (3-12), $\mathbf{\wp\, \wp(A \cup R)}$ can be defined following the above steps. Therefore, $\mathbf{A \times R}$ is defined. This is the product performance.

The equations in the above proof are not numbered because they are not to be used in the rest of the thesis.

Product performance **P** can be investigated from two aspects:

1) The focus is on the composition of each performance.

$$\begin{aligned} &\mathbf{\forall p \in P, \exists a \in A, \exists r \in R} \\ &\mathbf{p = < a, r >} \end{aligned} \tag{3-14}$$

2) The focus is on the relation from action to performance.

$$\begin{aligned}&\forall \mathbf{S}_i \in \mathbf{S}, \exists \mathbf{A}_i \subset \mathbf{A}, \exists \mathbf{R}_i \subset \mathbf{R}, \exists \mathbf{K}'_i \subset \mathbf{K}\\ &\mathbf{R}_i = \mathbf{K}'_i(\mathbf{A}_i)\end{aligned} \tag{3-15}$$

This relation is a set of product performance knowledge.

According to Axiom 1, performance knowledge is deterministic in nature. In other words, once a product is defined, the relation from actions to response is deterministic. Equation (3-15) can be written in another form as

$$\begin{aligned}&\mathbf{K}_i : \mathbf{S}_i \rightarrow \mathbf{P}_i\\ &\mathbf{P}_i \subseteq \mathbf{A}_i \times \mathbf{R}_i\\ &\mathbf{R}_i = \mathbf{K}'_i(\mathbf{A}_i)\end{aligned} \tag{3-16}$$

More detailed discussion will be made in Section 6.2.

3.4.3 Partition of product properties

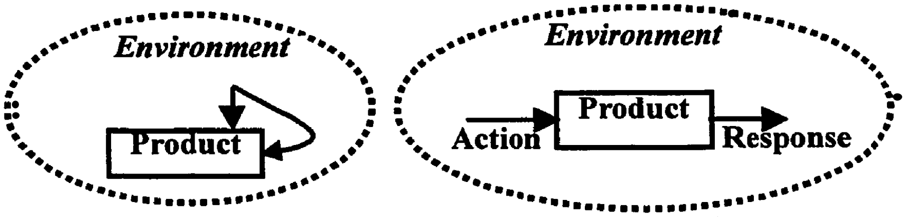
It is obvious from Figure 3-5 that two objects need to be formulated: performance (action and response) and structure (product and environment). Therefore, there exist two subsets of properties in representing products. One is related to product structure and another is associated to product performance. They are called structural property set and behavioral property set, respectively.

$$\mathbf{X} = \mathbf{X}^s \cup \mathbf{X}^b \tag{3-17}$$

where $\mathbf{X}$: product property set;

$\mathbf{X}^s$: structural property set;

$\mathbf{X}^b$: behavioral property set.



a) Product Structural Property b) Product Behavioral Property

Figure 3-6 Product property

The difference between two properties depends on the context. Generally speaking, structural properties are those that do not depend on environment while behavioral properties are always defined with reference to the enviornment. This difference is graphically shown in Figure 3-6.

For example, "color" in most cases is structural property. But if the color of a product can change according to environment, then it becomes a behavioral property.

All the properties in a design field constitute a property pool.

In terms of Equation (3-4), product descriptions, actions and responses, as objects, can be represented as

$$\begin{aligned} &\forall \mathbf{S}, \mathbf{S} = \{\mathbf{x}_i \mid \mathbf{i} = \mathbf{1}_s, \mathbf{2}_s, \cdots, \mathbf{n}_s, \mathbf{x}_i \in \mathbf{X}\} \\ &\forall \mathbf{a} \in \mathbf{A}, \mathbf{a} = \{\mathbf{x}_i \mid \mathbf{i} = \mathbf{1}_a, \mathbf{2}_a, \cdots, \mathbf{m}_a, \mathbf{x}_i \in \mathbf{X}\} \\ &\forall \mathbf{r} \in \mathbf{R}, \mathbf{r} = \{\mathbf{x}_i \mid \mathbf{i} = \mathbf{1}_r, \mathbf{2}_r, \cdots, \mathbf{p}_r, \mathbf{x}_i \in \mathbf{X}\} \end{aligned} \tag{3-18}$$

Theorem 3.3 Product property set can be partitioned into two subsets: structural and behavioral sets. Structural properties are independent of environment while behavioral properties are defined by both product and environment descriptions.

Table 3-1 gives some examples of structural and behavioral properties widely used in mechanical design.

Table 3-1 Properties in mechanical design

Name	Type	Value	Examples
direction	structural	{x,y,z}	<direction,x>
coordinate	structural	$R \times R \times R$	<coordinate,<21.0,10.0,30.0>>
shape	structural	String	<shape, circle>
density	structural	R	<density,15.0>
material_name	structural	String	<material_type,steel>
viscosity	structural	R	<viscosity,0.6>
force_name	behavioral	String	<force_name, torque>
force_magnitude	behavioral	R	<force_magnitude,500.0>
Note: in this table, R is the set of real number			

3.4.4 Structure of product and environment

In this section, we will discuss product and environment structures. Since both are the same mathematically as was manifested in Equation (3-12) and Figure 3-4, only product structure will be studied. The conclusions apply to environment structure.

There are two cases to consider:

1) Product is defined by a set of properties.

$$\mathbf{S} = \{\mathbf{x}_i : \mathbf{i} = 1,2,\ldots \mathbf{n}_s, \mathbf{x}_i \in \mathbf{X}\} \tag{3-19}$$

According to Axiom 1, the product structure ⊕S will include two parts: one is the product description itself and another is the relations from set S to S. This relation is the physical laws and geometric theorems, which define the relation between different properties of a product. It will be further addressed in Section 4.2.1, 4.3.1, and 6.2.

2) Product is defined by a set of other products.

$$\mathbf{S} = \mathbf{S}_1 \cup \mathbf{S}_2 \cup \cdots \cup \mathbf{S}_{n_s}, \mathbf{S}_i \text{ is a product}, \mathbf{i} = 1,2,\cdots \mathbf{n}_s \tag{3-20}$$

According to Equation (3-5), the product structure ⊕S is then represented as

$$\oplus \mathbf{S} = \bigcup_{i=1}^{n_s} \mathbf{S}_i \cup \bigcup_{i=1}^{n_s} \bigcup_{j=1}^{n_s} \mathbf{S}_i \times \mathbf{S}_j \tag{3-21}$$

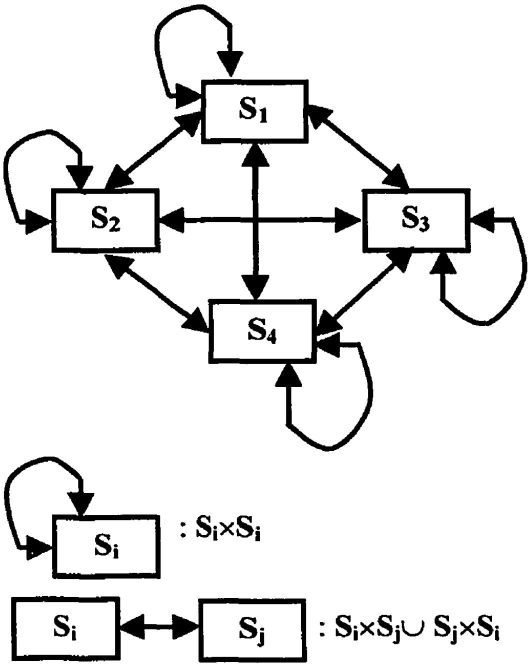


Figure 3-7 Product structure

They form a network. For a product with four components, this network is shown in Figure 3-7. In the equation, each component can still be objects, their structure can form refined networks. This structuring process continues until every component can be defined in the form of Equation (3-19). Further discussion is made in Sections 4.2.2 and 4.3.2.

Theorem 3.4 A product can be structured through a set of components. Components are also objects, which can be further structured until they can be defined by a set of properties.

The structure of environment will be given in Section 4.4.

3.4.5 Design requirements

Design requirements are the constraints on a product design so that the designed product can be manufactured to achieve its desired functions in its working environments. It is obvious from Figure 3-5 that there are only two possibilities to constrain a product: constraints on product structure or product performance.

According to Equations (3-18), a design requirement can be formally represented as

$$\mathbf{r}^{\mathbf{d}} = \lambda(\mathbf{x}_i, [\mathbf{x}_i]) \tag{3-22}$$

where $\mathbf{r}^{\mathbf{d}}$: design requirement;

$\mathbf{x}_i$: product property, structural as well as behavioral;

$[\mathbf{x}_i]$: constraint on product property;

λ: relational operator, such as =, <, >, and so on.

3.4.6 Design evaluation

In Section 3.4.4, product and environment structure is given in Equation (3-21). This section will focus on design evaluation process by considering the simplest case of Equation (3-21): decomposing a product into two parts: $\mathbf{S}'$ and $\mathbf{S}_i$. Hence,

$$\mathbf{S} = \mathbf{S}' \cup \mathbf{S}_i \tag{3-23}$$

Substitute the above equation into Equation (3-11), we get

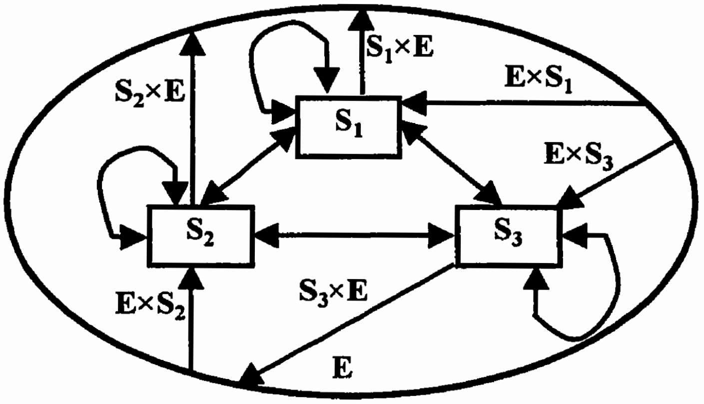
$$\begin{aligned}\oplus\mathbf{U} &= \oplus(\mathbf{E}\cup\mathbf{S}) \\ &= \oplus(\mathbf{E}\cup\mathbf{S}'\cup\mathbf{S}_i) \\ &= \oplus((\mathbf{E}\cup\mathbf{S}_i)\cup\mathbf{S}') \\ &= \oplus(\mathbf{E}'\cup\mathbf{S}') \\ &= \oplus\mathbf{E}'\cup\oplus\mathbf{S}'\cup\mathbf{E}'\times\mathbf{S}'\cup\mathbf{S}'\times\mathbf{E}' \\ &= \oplus\,\mathbf{E}'\cup\oplus\mathbf{S}'\cup\mathbf{A}'\cup\mathbf{R}' \\ \mathbf{E}' &= \mathbf{E}\cup\mathbf{S}_i\end{aligned} \tag{3-24}$$

In terms of Equation (3-13), the product performance, which was the hidden interactions between the extracted component and the left product structure, can then be specified using $\mathbf{A}'$ and $\mathbf{R}'$. This process is repeated until $\mathbf{S}'$ becomes empty. The union of all intermediate $\mathbf{A}'$ and $\mathbf{R}'$ embodies all performance information of a product in the given environment. This is analysis process. Therefore, by moving each component of a product into its environment, the performances of product can be studied. This approach has been widely adopted in engineering analysis.

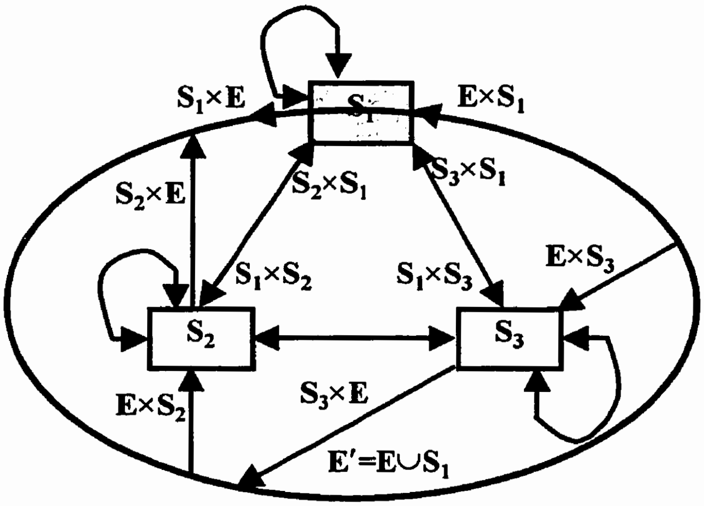
In each step, the mapping from $\mathbf{S}_i$ to $\mathbf{A}'$ and $\mathbf{R}'$ can be taken as performance knowledge $\mathbf{K}_i$

$$\begin{aligned}&\mathbf{K}_i:\mathbf{S}_i\rightarrow\mathbf{P}_i \\ &\mathbf{P}_i\subseteq\mathbf{A}'\times\mathbf{R}'\end{aligned} \tag{3-25}$$

Taking a three-component product as an example, the above process can be demonstrated in Figure 3-8.



a) Original product-environment system



b) Updated product-environment system: S_1 as a part of environment

Figure 3-8 Process of design evaluation

Theorem 3.5 A product's performances can be analyzed through performance knowledge by gradually separating each component from other components. This process is deterministic in nature.

3.4.7 Design synthesis

Another case is that product S is not well defined in terms of **Theorem 3-4**. The extreme case is that the product structure is not known at all. The objective here is then to find product structures so that the product-environment system can be completed and Equation (3-22) is satisfied.

The current product-system takes the following form

$$\oplus \mathbf{U} = \oplus \mathbf{E} \cup \mathbf{A} \cup \mathbf{R} \tag{3-26}$$

$\oplus \mathbf{S}$ is missing in the above equation.

Apply the performance knowledge in Equation (3-25) in a reverse way

$$\begin{aligned} &\mathbf{K}_i^{-1} : \mathbf{P}_i \rightarrow \mathbf{S}_i \\ &\mathbf{P}_i \subset \mathbf{A} \times \mathbf{R} \end{aligned} \tag{3-27}$$

As a result, performance will also be defined

$$\mathbf{P}_i' = \mathbf{K}_i(\mathbf{S}_i) \tag{3-28}$$

The product-environment system can then be updated as

$$\begin{aligned} \oplus \mathbf{U} &= \oplus(\mathbf{E} \cup \mathbf{S}_i) \\ &= \oplus \mathbf{E} \cup \oplus \mathbf{S}_i \cup \mathbf{P}_i' \end{aligned} \tag{3-29}$$

$\mathbf{P}_i'$ will be evaluated against $\mathbf{P}_i$ using Equation (3-22). If Equation (3-22) is satisfied, then the performance $\mathbf{P}_i$ is well defined. $\oplus(\mathbf{E} \cup \mathbf{S}_i)$ would be the structure of newly generated product-environment system. Otherwise this process will continue. In this way the set of product is augmented until no more product component can be added. This is the synthesis process. It should be noted that according to Axiom 1, $\mathbf{K}_i^{-1}$ is not deterministic but plausible in nature. Many possibilities may exist in generating $\mathbf{S}_i$ from $\mathbf{P}_i$.

The process is illustrated in Figure 3-9. The arrows in the figure represent the relations among the components. They can be action, response, or geometric relations.

3.4.8 Design governing equation

Substituting Equation (3-25) into Equation (3-27), we get

$$\mathbf{K}_i^{-1} \bullet \mathbf{K}_i : \mathbf{S}_i \rightarrow \mathbf{P}_i \rightarrow \mathbf{S}_i \qquad (3\text{-}30)$$

or

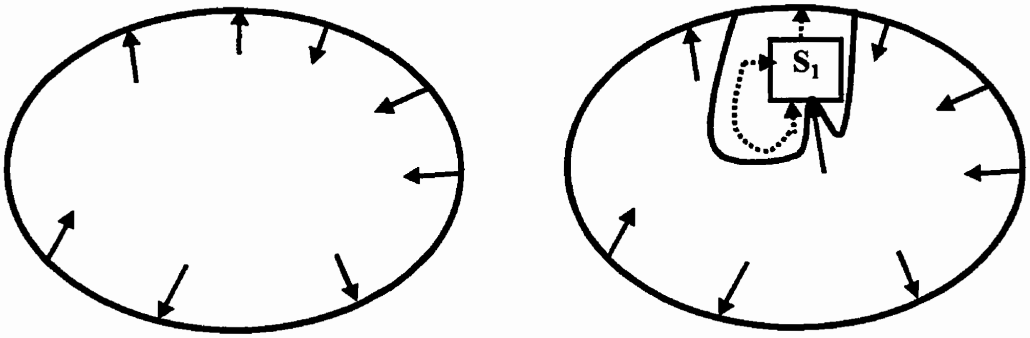
$$\mathbf{S}_i = \mathbf{K}_i^{-1} \bullet \mathbf{K}_i(\mathbf{S}_i)$$

This equation is called design governing equation. If Axiom 1 does not hold, then the following equation will always be true:

$$\mathbf{K}_i^{-1} \bullet \mathbf{K}_i = \mathbf{I} \qquad (3\text{-}31)$$

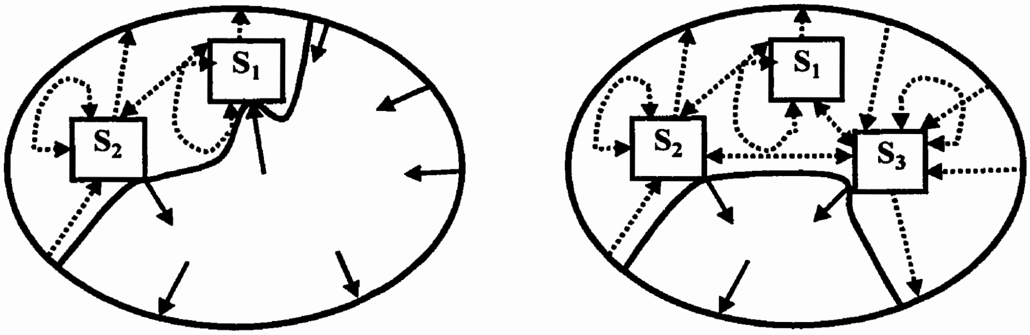
where **I** is a unit transformation.

In this case, design would become a well defined problem. The detailed implication of this equation will be discussed in Chapter 7.



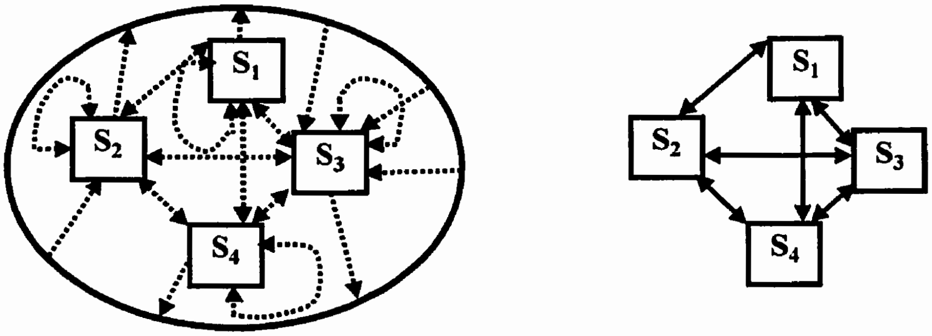
a) Original product-environment system

b) One product component added



c) Two product components added

d) Three product components added



e) Four product components added

f) Final product

Figure 3-9 Design synthesis process

3.5 Related Work

There are two groups of work that are comparable to that accomplished in this thesis. They are the General Design Theory by Yoshikawa and Tomiyama (Yoshikawa, 1981;Tomiyama and Yoshikawa, 1985) and the Formal Design Theory by Braha and Maimon (1998). Despite the common point in that all the three theories take set theory as the language of representation, the fundamental difference lies in the selection of axioms. The following two subsections will discuss the difference in details.

3.5.1 Comparison to the general design theory

Tomiyama made a straightforward summary of the general design theory in 1994 (Tomiyama, 1994). His summary is adapted in the following to compare our theory in this thesis.

GDT (General Design Theory) begins with a manifesto that our knowledge can be mathematically formalized and operated. This is represented by three axioms that define knowledge as topology and operations as set operations. GDT regards a design process as a mapping from the function space to the attribute space, both of which are defined over the entity concept set. GDT makes a distinction between an entity and an entity concept. An entity is a concrete existing object, and an entity concept is its abstract, mental impression conceived by a human being. In terms of Figure 3-1, entity is the being in the objective world while entity concept is the being in the subjective world. Yoshikawa established three axioms for the general design theory (Yoshikawa, 1981):

Axiom1. (Axiom of recognition) Any entity can be recognized or described by the attributes.

Axiom 2. (Axiom of correspondence) The entity set and the entity-concept set have one-to-one correspondence.

Axiom 3. (Axiom of operation) The abstract-concept set is a topology of the entity-concept set.

Axiom 2 guarantees the existence of a superman who knows everything. Axiom 3 signifies that it is possible to logically operate abstract concepts as if they were just ordinary mathematical sets. Accordingly, they get set operations, such as intersection, union, and negation. They can then introduce ideal knowledge that knows all the elements of the entity set perfectly and that can describe each element with abstract concepts without ambiguity. Because of this, the design solution is immediately obtained after the specifications are described.

However, the situation in the ideal knowledge does not apply to real design in many points. First, design is not a simple mapping process but rather a stepwise refinement process where the designer seeks the solution that satisfies the constraints. Second, the ideal knowledge does not take physical constraints into considerations, and it can produce design solutions such as perpetual machines. These restrictions are considered in the real knowledge where design is regarded as process in which the designer builds the goal and tries to satisfy the specifications without violating physical constraints. To formalize the real knowledge, they first defined a physical law as a description about the relationship among physical quantities of entities and the field. The concept of physical laws is one of the abstract concepts formed when one looks at a physical phenomenon as manifestation of physical laws. Based on a new hypothesis, which states that finite subcoverings exist for any coverings of the set of feasible entity concepts made of sets chosen from the set of physical law concepts, they claimed that they can prove finiteness or boundedness of our knowledge. It should be noted that the finiteness here means that a feasible entity is explicable not by an infinite number but a finite number of physical laws. With these considerations, they formalized design process as a metamodel evolution process. This model indicates that in real design, design is a stepwise transformation process from the function space to the attribute space via the metamodel space. Here, metamodel space is physical phenomenon space.

The current axiomatic theory, however, differs from GDT in the following aspects:

- Bounded rationality is taken as a fundamental axiom. This is against the axiom 2 in GDT. Metamodel introduced the real knowledge required to map design specifications to design solutions, but the nature of this knowledge was not clear.
- To formalize design in real knowledge, they introduced new hypothesizes and new definitions. Our axiomatic theory takes two axioms. In GDT's terminology, one (Axiom of object structuring) concerns the description of entity in ideal knowledge, which covers all possible relations. Another (Axiom of bounded rationality) concerns the description of entity concept. Axiom 1 in GDT was adopted to define object in our theory.

3.5.2 Comparison to the formal design theory

Braha and Maimon (1998) established a formal design theory (FDT) based on the following five postulates:

Postulate 1 (Entity-Relational Knowledge Representation): an artifact representation is built upon the multiplicity of modules (attributes) and relationships among them.

Postulate 2 (Nested Hierarchical Representation): the design of any complex system can be considered at various levels. The general direction of design is from more abstract to less. A design at any level of abstraction is a description of an organized collection of constraints (such as various structural, cause-effect, functional, and performance features) that appear in the physically implemented design.

These two premises lead directly to formulate the attribute space as an algebraic structure. The artifact is represented by the pair <*M*, *C*>. *M* stands for the set of modules that the artifact is comprised of; and *C* denotes the set of relations that represent the relationships among the modules. In order to capture the essence of design, a hierarchical construction of systems from subsystems is also developed. Consequently, the general set of modules is classified into basic and complex modules. Basic modules represent entities that can not be defined in terms of others. Complex modules are defined hierarchically in terms of other modules, where the effects of their interaction are expressed.

Postulate 1, if represented using the language in this thesis, would be:

$$\forall \mathbf{O} = \mathbf{A} \bigcup \mathbf{B},\ \oplus \mathbf{O} ::= \mathbf{O} \cup (\mathbf{A} \times \mathbf{B}) \qquad (3\text{-}32)$$

while our theory assumes:

$$\forall \mathbf{O} = \mathbf{A} \bigcup \mathbf{B},\ \oplus \mathbf{O} ::= \mathbf{O} \cup ((\mathbf{A} \bigcup \mathbf{B}) \times (\mathbf{A} \bigcup \mathbf{B})) \qquad (3\text{-}33)$$

Postulate 2 is a logical conclusion of the axiomatic system established in this thesis.

Postulate 3 (Incompleteness): any knowledge representation (as represented by the designer) is incomplete.

Postulate 4 (Bounded rationality): the designer can consider only a subset of knowledge representations at any instant of decision making.

Postulate 5 (Non-Determinism): several feasible designs can be generated to the level specified by the designer.

Postulate 3 and 4 are included in our axiom 1 (Axiom of bounded rationality). Postulate 5 is a logical conclusion of our axiomatic system.

3.5.3 Conclusions

Because of the differences in the selection of axioms, the three axiomatic theories are different. This difference also leads to the differences in the derived theorems. FDT is more mathematically and logically consistent and complete compared to GDT. Ours is more concise and easier to operate. The axiom of bounded rationality have been recognized by the three theories from different perspectives. The axiom of object structuring is unique in our axiomatic theory. It is our belief that it makes the theory more robust and logical in deriving the theorems about design compared to the other two.

3.6 Summary

This chapter established an axiomatic approach to studying product design. With this approach, design activities were investigated by deriving theorems from axioms using the mathematical language.

In the current axiomatic system, there are two axioms, one is the axiom of bounded rationality which indicates the imperfection of human recognition, another is the axiom of object structuring, which embodies all possible relations within an object. Based on this axiomatic system, models of product-environment system and design processes are derived. The design governing equation is also derived.

The significance of this work lies in that it provides a formal method to approaching design studies. In ideal situations, design studies become the derivation of mathematical equations plus the explanation of factual meaning of these equations.

The following chapters will give the concrete meaning of these equations and develop models to improve design practice. Chapter 4 will elaborate the definition of product-environment system. Chapter 5 refines the definition of design requirements. Chapter 6 will further explain the design process model. Chapter 7 and 8 are exploratory, focusing on the nature of the design problem and design activities. Chapter 9 gives an example of using this theory to develop a potential mechanism design software.

Chapter 4 Formulation and Formalization of Product-Environment System

4.1 Introduction

In Chapter 1, it was indicated that the objective of the current thesis research is to formulate and formalize the general product conceptual design process as shown in Figure 4-1.

Figure 4-1 Design activity

The design process begins with design requirements and ends with product descriptions. Therefore, in modeling this process, two basic elements: design requirements and product descriptions, need to be defined and formulated in the first place. The formulation should also take into account the following two basic factors:

1) Designing is an evolving process from simple and abstract concepts to complicated and concrete product descriptions. During this process, design requirements, product descriptions, product performances and product environment change as the design process progresses. In the design process, candidate product descriptions are generated based on a set of design requirements. The product descriptions are then evaluated against design requirements to determine if the designed products satisfy the requirements and to choose the best appropriate solution. The process iteratively generates conceptual, configuration, and detailed designs. To support this process, a

dynamic object representation scheme is proposed to embody the time-evolution factor in design. The scheme is given in Figure 4-2.

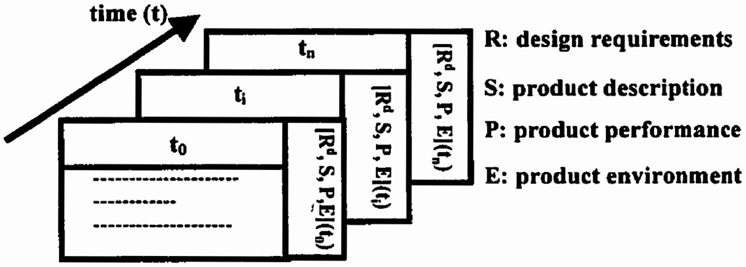


Figure 4-2 A framework of design representation

2) In a real world design, the number of potential product designs is infinite while the means of representation is always finite, especially in computer implementation. It is therefore essential for the representation scheme to have a finite means to handle the infiniteness existing in the problem. A natural way to do this in scientific research is to define a set of basic elements and combination rules. By combining the basic elements with the combination rules, any object can be constructed. An obvious example is the representation of a point in 3D space in an O-XYZ coordinate system.

Chapter 3 derived the formal representation of these two objects, namely design requirements and product descriptions. This chapter will discuss product descriptions and corresponding product performance and product environment in more detail. The modeling of each object will include primitive objects and complex objects, which embodies the dynamic nature of each representation. Design requirements will be discussed in Chapter 5. Chapter 6 focuses on the design synthesis process. The discussion in these three chapters will show that the derived equations in Chapter 3 constitute the basics of a formal design process model.

4.2 Product Descriptions

In this section, the focus will be placed on product descriptions. The discussion includes three subsections: primitive products, compound products, and levels of complexity and abstraction. The first two subsections deal with the infiniteness problem while the third shows how this approach embodies the dynamic nature of the representation.

4.2.1 Primitive product

Just like that any 3D point can be defined by its x, y, and z coordinates, any product can also be defined by basic products. In mechanical design, basic products include components and connectors such as gears, shafts, bearings, fixations, joints, fasteners, couplings, welds and so on (Gui and Mantyla, 1994). They are called primitive products, which are well defined by a set of product structural properties without turning to other product components.

$$\begin{aligned} &(\mathbf{S}^{a})_{i} = \{\mathbf{x}_{j} \mid \forall \mathbf{x}_{j} \in \mathbf{X}, \forall (\mathbf{x}_{j})^{n} \in \mathbf{D}^{x}, \exists (\mathbf{x}_{j})^{v} \in \mathbf{R}^{x}, \mathbf{x}_{j} = < (\mathbf{x}_{j})^{n}, (\mathbf{x}_{j})^{v} >^{s}, 1 \leq \mathbf{j} \leq \mathbf{n}_{s}\} \qquad (4\text{-}1) \\ &\mathbf{S}^{a} = \{(\mathbf{S}^{a})_{i} \mid \mathbf{i} = 1,2,3,\ldots,\mathbf{n}_{a}\} \end{aligned}$$

where $(\mathbf{S}^{a})_{i}$: a primitive product;

$\mathbf{S}^{a}$: a set of primitive products;

$\mathbf{x}_{j}$: a structural property;

$(\mathbf{x}_{j})^{n}$: property name;

$(\mathbf{x}_{j})^{v}$: property value;

$\mathbf{X}^{s}$: set of structural properties, the definition is given in Equation (3-17);

$\mathbf{n}_s$: the number of properties defining a primitive product;

$\mathbf{n}_a$: the number of primitive products in a design domain;

$\mathbf{D}^x$: domain of property; and

$\mathbf{R}^x$: range of property.

This definition refines that given in Equation (3-19).

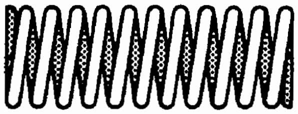


Figure 4-3 A compression coil spring SP1 (Hindhede *et al*,1983)

Figure 4-3 is an example of the straight-sided helical compression springs. Table 4-1 gives a definition of the spring according to Equation (4-1).

Two points should be noted here. First, in describing a component, the least number of properties can be used to provide a complete product description. In the above compression coil spring example, either of outside diameter, wire diameter, or inside diameter is redundant. Second, if a property value takes a range rather than one single value, then the description designates a set of same type of components. It is then called primitive product type, which is defined in Equation (4-2).

Table 4-1 Description of a compression coil spring

Informal Description		Formal Description
Property Name	Property Value	
Outside diameter (d0)	V_1=25.0mm	<d0,25.0> $V_1 \in \{7.0,8.0,\ldots,19.0,20.0,22.0,25.0,28.0,\ldots,60.0,65.0\}$
Wire diameter (d)	V_2=2.0mm	<d,2.0> $V_2 \in \{1.0,1.25,1.4,1.6,1.8,2.0,2.25,2.5,2.8,3.2,\ldots,10.0\}$
Inside diameter (d1)	V_3=21.0mm	<d1,21.0> $V_3 \in \{d_I \mid d_I = d0-2*d\}$
Free length (L_0)	V_4=234.0mm	< L_0,234.0> $V_4 \in R$ (the set of real numbers)
Active coils (N_A)	V_5=11	< N_A,11> $V_5 \in Z^+$ (the set of integer numbers)
Inactive coils (N_I)	V_6=2	< N_i,2> $V_6 \in Z^+$ (the set of integer numbers)
Pitch (p)	V_7=20.0mm	<p,20.0> $V_7 \in R$ (the set of real numbers)
Material(m)	V_8= carbon-Steel	<m, carbon_ steel> $V_8 \in$ {carbon steel, alloy, stainless steel,…}
SP1={<d0,25.0>,<d,2.0>,<d1,21.0>,< L_0,234.0>,< N_A,11>,< N_I,11>, <p,20.0>}		

$$
\begin{aligned}
(\mathbf{S}^{\mathbf{at}})_{\mathrm{i}} &= \{(\mathbf{S}^{\mathbf{a}})_{\mathrm{k}} \mid \mathbf{k} = \mathbf{1,2,\ldots}\} \\
&= \{\mathbf{x}_{\mathrm{j}} \mid \forall \mathbf{j}, \mathbf{1} \le \mathbf{j} \le \mathbf{n}, \forall (\mathbf{x}_{\mathrm{j}})^{n} \in \mathbf{D}^{\mathbf{x}}, \exists (\mathbf{x}_{\mathrm{j}})^{\mathbf{v}} \subset \mathbf{R}^{\mathbf{x}},\ \mathbf{x}_{\mathrm{j}} = < (\mathbf{x}_{\mathrm{j}})^{\mathbf{n}}, (\mathbf{x}_{\mathrm{j}})^{\mathbf{v}} >,\ \mathbf{x}_{\mathrm{j}} \in \mathbf{X}\} \qquad (4\text{-}2) \\
\mathbf{S}^{\mathbf{at}} &= \{(\mathbf{S}^{\mathbf{at}})_{\mathrm{i}} \mid \mathbf{i} = \mathbf{1,2,\ldots,m}\}
\end{aligned}
$$

where $(\mathbf{S}^{\mathbf{at}})_{\mathbf{i}}$: a primitive product type;

$\mathbf{S}^{\mathbf{at}}$: a set of primitive product types;

$\mathbf{x}_{\mathbf{j}}$: structural property;

$(\mathbf{x}_{\mathbf{j}})^{\mathbf{n}}$: property name;

$(\mathbf{x}_{\mathbf{j}})^{\mathbf{v}}$: property value;

$\mathbf{X}^{\mathbf{s}}$: set of structural properties, the detailed definition is given in Section 5.3.3;

$\mathbf{n}_{\mathbf{a}}$: the number of primitive products in a design domain;

$\mathbf{D}^{\mathbf{x}}$: domain of property;

$\mathbf{R}^{\mathbf{x}}$: range of property.

For the compression coil spring example, the product type can be described in Table 4-2.

Table 4-2 Description of compression coil springs

Informal Description		Formal Description
Property Name	Property Value	
Outside diameter (d0)	V_1	$V_1 \in \{7.0, 8.0, \ldots, 19.0, 20.0, 22.0, 25.0, 28.0, \ldots, 60.0, 65.0\}$
Wire diameter (d)	V_2	$V_2 \in \{1.0, 1.25, 1.4, 1.6, 1.8, 2.0, 2.25, 2.5, 2.8, 3.2, \ldots, 10.0\}$
Inside diameter (d1)	V_3	$V_3 \in \{x \mid x = D0 - 2*d\}$
Free length (L_0)	V_4	$V_4 \in R$ (the set of real numbers)
Active coils (N_A)	V_5	$V_5 \in Z^+$ (the set of integer numbers)
Inactive coils (N_I)	V_6	$V_6 \in Z^+$ (the set of integer numbers)
Pitch (p)	V_7	$V_7 \in R$ (the set of real numbers)
Material(m)	V_8	$V_8 \in$ {carbon steel, alloy, stainless steel,...}
$SP1=\{<d0, V_1>,<d, V_2>,<d1, V_3>,< L_0, V_4>,< N_A, V_5>,< N_I, V_5>, <p, V_7>,<m, V_8 >\}$		

Obviously primitive product type is more abstract than primitive product. But for the sake of simplicity, we will call both of them primitive product in this thesis whenever it does not cause confusion.

Figure 4-4 gives a class of primitive springs in mechanical design. They are represented as

$$\begin{aligned}&\textbf{primitive\_springs}\\&\quad=\{\textbf{compression\_coil\_spring, extension\_coil\_spring, torsion\_bar,}\\&\qquad\textbf{torsion\_coil\_spring, flat\_spring, volute\_spring, flat\_spiral\_spring,}\\&\qquad\textbf{belleville\_spring, leaf\_spring}\}\end{aligned}\tag{4-3}$$

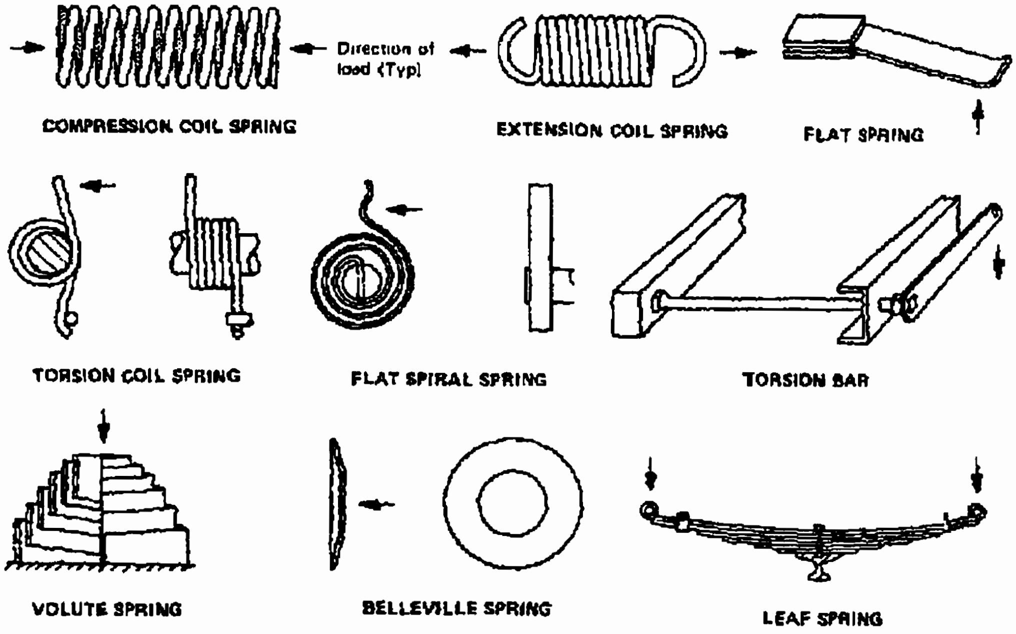


Figure 4-4 Types of mechanical springs (Hindhede *et al*,1983)

In real engineering design, the selection of primitive product is artificial, depending on designer's expertise and inclination.

4.2.2 Compound products

Very few products in real life design are primitive. They are more often a set of components related to each other through component connectors. Such a product can be formally represented as

$$\begin{aligned} &\mathbf{S} = \mathbf{S}^c \cup \mathbf{S}^r \\ &\mathbf{S}^c = \bigcup_{i=1}^{n_c} (\mathbf{S}^c)_i \\ &\mathbf{S}^r = \bigcup_{i=1}^{n_t} (\mathbf{S}^r)_i \end{aligned} \tag{4-4}$$

where S: a product;

S^c: the set of all the components included in the product;

S^r: the component connectors linking components in S^c;

$(S^c)_i$: a components included in the product **S**;

$(S^r)_i$: a component connector linking components in S^c.

n_c: the number of components in the product **S**;

n_t: the number of component relations in the product **S**.

Product S can be further defined as

$$\begin{aligned} &\mathbf{S} = \bigcup_{i=1}^{n_s} \mathbf{S}_i \\ &\mathbf{S}_i = \begin{cases} (\mathbf{S}^c)_i & 1 \le i \le n_c \\ (\mathbf{S}^r)_i & n_c < i \le n_s \end{cases} \\ &n_s = n_c + n_t \end{aligned} \tag{4-5}$$

According to Equations (3-20) and (3-21), the structure of the product defined by Equation (4-5) is

$$\begin{aligned}\oplus \mathbf{S} &= \bigcup_{i=1}^{n_s} \mathbf{S}_i \cup \bigcup_{i=1}^{n_s}\bigcup_{j=1}^{n_s} \mathbf{S}_i \times \mathbf{S}_j \\ &= \bigcup_{i=1}^{n_s} \oplus \mathbf{S}_i \cup \bigcup_{i=1}^{n_s}\bigcup_{\substack{j=1\\ j\neq i}}^{n_s} \mathbf{S}_i \times \mathbf{S}_j\end{aligned} \qquad (4\text{-}6)$$

where ⊕ is the structure operator defined in Equation (3-5). ⊕S is read as "structuring S" or "the structure of S"

The above equation means two things: first, there are interactions between component ($S_i \times S_j$) in structuring a product. This will be addressed in Section 4.3. Second, a component may also be composed by other components through their relations ($\oplus S_i$). This renders the structuring of a product recursive and brings in a hierarchical structure of the product description. It is shown in Figure 4-5.

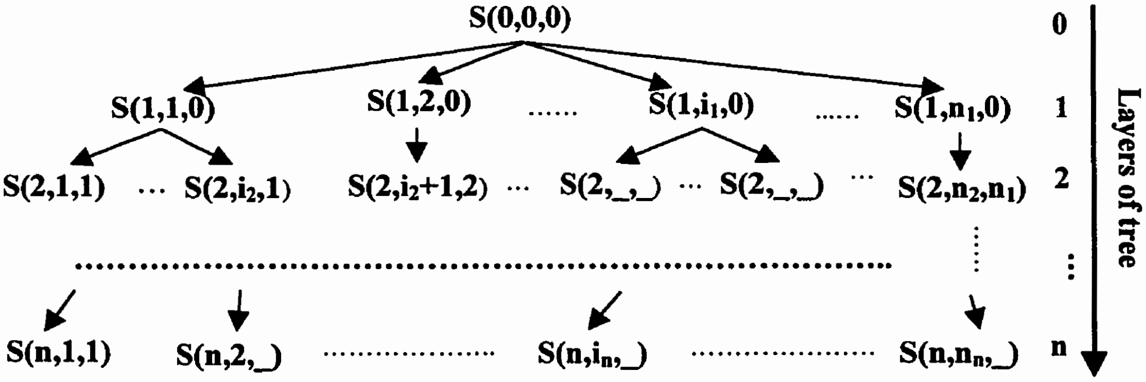


Figure 4-5 Tree structure of product description

In the above tree, by defining $\mathbf{S(k,i_k,j_{k-1})}$ as the node at the $\mathbf{i_k}$th position in the **k**th layer with a parent node at the $\mathbf{j_{(k-1)}}$th position in the **(k-1)**th layer, the product is then described recursively as

$$\begin{aligned}
&\mathbf{S(0,0,0)} = \bigcup_{i_1=1}^{n_1} \mathbf{S(1,i_1,0)} \\
&\forall \mathbf{i}_k, 1 \le \mathbf{i}_k \le \mathbf{n}_k, \mathbf{S(k,i_k,j_{k-1})} = \bigcup_{i_{k+1}=m_0+1}^{m_n} \mathbf{S(k+1,i_{k+1},i_k)} \\
&\forall \mathbf{i}_k, \mathbf{S(n,i_k,\_)} \in \mathbf{S}^a \\
&\mathbf{n}_0 = 1, \mathbf{n}_k = \sum_{i_k=1}^{n_{k-1}} \mathbf{n(k+1,i_k)}, \quad \mathbf{m}_0 = \sum_{i=1}^{i_k-1} \mathbf{n(k+1,i)}, \quad \mathbf{m}_n = \mathbf{m}_0 + \mathbf{n(k+1,i_k)}
\end{aligned} \tag{4-7}$$

where **S(0,0,0)**: root node;

n(k+1,i_k): number of child nodes of the **i_kth** node in the **kth** layer of the tree;

n_k: total number of nodes in the **kth** layer of the tree.

A typical block of the tree is shown in Figure 4-6.

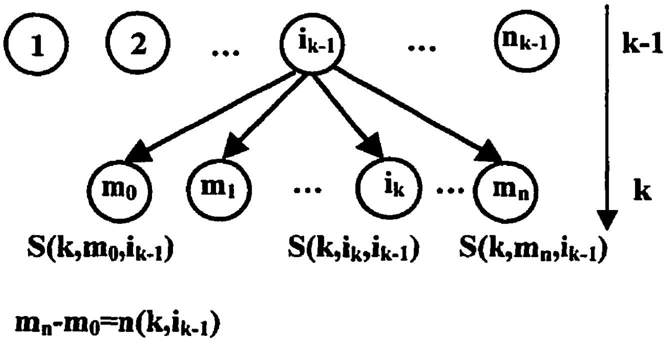


Figure 4-6 Basic block of product description tree

Indeed, Equation (4-7) has provided a formal approach to representing any product based on the primitive product set given in Equations (4-1) and (4-2). The objective of representing infinite number of products by a finite means has thus been realized.

By rewriting Equation (4-7), we get

$$\begin{aligned} & \mathbf{S = S[0] = S(0,0,0)} \\ & \forall \mathbf{k}, 1 \le \mathbf{k} \le \mathbf{n}\ \mathbf{S[k]} = \bigcup_{\mathbf{i_k}=1}^{\mathbf{n_k}} \mathbf{S(k, i_k, \_)} \\ & \forall \mathbf{i_n}, \mathbf{S(n, i_n, \_)} \in \mathbf{S^a} \end{aligned} \tag{4-8}$$

where **k**: layer of representation;

S[k]: structural representation in the **k**th layer;

S(k,i$_k$,_): the component at the **i$_k$**th position in the **k**th layer;

S^a: primitive product set.

The above equation is actually Equation (4-5) refined by considering different layers in the hierarchical structure of product description. In fact, Equations (4-1), (4-2), (4-7), and (4-8) constitute a mathematical representation of product descriptions.

Product can be further defined as

$$\begin{aligned} & \mathbf{S = S[0]} = \bigcup_{\mathbf{i}=1}^{\mathbf{n_i}} \mathbf{S_i[1]} \\ & \mathbf{S_i[n]} = \bigcup_{\mathbf{j}=1}^{\mathbf{n_{ij}}} \mathbf{(S_i)_j} \end{aligned} \tag{4-9}$$

It will be shown in Section 4.2.3 that this representation implies levels of complexity and abstraction of product description.

The primitive product in Equations (4-1) and (4-2) define the lowest level of abstraction and the highest level of complexity of a product description. In fact, the choice and the definition of primitive products vary with design domains and designers' expertise and preferences.

Needless to say, different design domains have different primitive products. Meanwhile, experienced designers may have more complex primitive products in mind compared with naïve ones.

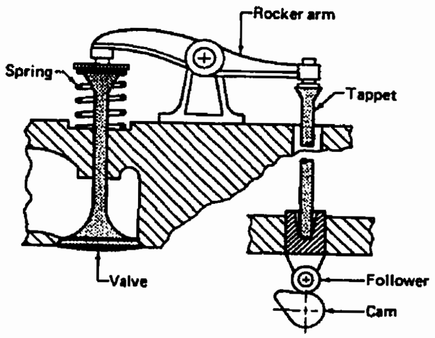


Figure 4-7 A rocker arm assembly (Hindhede *et al*,1983)

A rocker arm assembly example shown in Figure 4-7 can be used to explain the above notions. It consists of the following components:

RM_1=valve

RM_2=spring

RM_3=rocker arm

RM_4=tappet

RM_5=follower assembly

RM_6=cam assembly

The rocker arm can then be represented according to Equation (4-5) as

$$\mathbf{RM} = \bigcup_{i=1}^{6} \mathbf{RM}_i \tag{4-10}$$

where every component can be further defined by other components until the component becomes a primitive product.

In the description given in Equation (4-10), $\mathbf{RM}_2$ is not necessarily to be refined since spring is one of the primitive products. In contrast, however, $\mathbf{RM}_6$ must be further described since it is not primitive by itself. It is shown in Figure 4-8, which is composed of

$\mathbf{CAM}_1$=Cam body

$\mathbf{CAM}_2$=Cam shaft

$\mathbf{CAM}_3$=Cam key

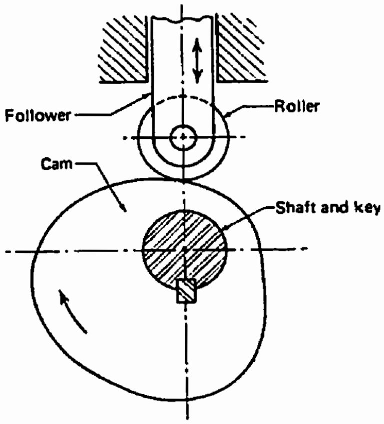


Figure 4-8 Cam assembly (Hindhede *et al*,1983)

Accordingly, the cam can be formally described as

$$\mathbf{CAM} = \bigcup_{i=1}^{3} \mathbf{CAM}_i \tag{4-11}$$

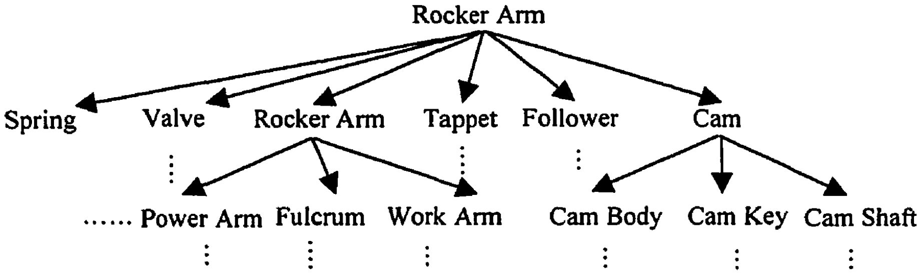


Figure 4-9 Tree structure of rocker arm

In this way, we have a tree structure of the rocker arm as is shown in Figure 4-9. The cam body, shaft, key and so on can be defined with a set of structural properties, respectively.

4.2.3 Levels of complexity and abstraction

As was requested in Figure 4-2, product descriptions should evolve with the design process. So it is essential to verify if Equation (4-8) satisfies the condition. Indeed, in Equation (4-8), each **S[k]** gives a representation of product with respect to the components of different levels of complexity. As the value of index **k** increases, product description **S[k]** becomes more detailed. This means that index **k** represents the degree to which the definition of product description is detailed. Therefore, index **k** is called the complexity level of product description. **S[k]** is the $\mathbf{k}^{\text{th}}$ order product description. In the tree structure of product description in Figure 4-5, if a component can not be further decomposed, the number of its successors becomes one. This component is thus a primitive product. If all the leaf nodes in Figure 4-5 become primitive products, then the product is said to be well defined. Otherwise the product description is partially defined, which is indeed an abstraction of a well-defined product description. Correspondingly, we have the notion of abstraction levels for product descriptions. The levels of complexity and abstraction for product descriptions are shown in Table 4-3. Obviously, lower order product description is the abstraction and the type of a higher order ones. This fact matches with a basic logical principle: the more a concept's intention is, the narrower the concept's extension has.

Table 4-3 Levels of complexity and abstraction of product description

Product Definition	**Complexity Level**	**Abstraction Level**
S[0]	0	n
S[1]	1	n-1
⋮	⋮	⋮
S[n]	n	0

The design process usually begins with a partially defined product description. In extreme cases, it begins from **S[0]**. The final design is obtained by expanding the leaf nodes into components with greater level of complexity. It is shown in Figure 4-10 how this process is supported with the model in this chapter. In the figure, t is the time that represents the progresses of design problem solving. This representation scheme provides a top-down approach of supporting the dynamic design process from generic and simple to concrete and complex product descriptions. An example will be given in Chapter 6 showing the application of this representation to the dynamic design process, which evolves as the design progresses.

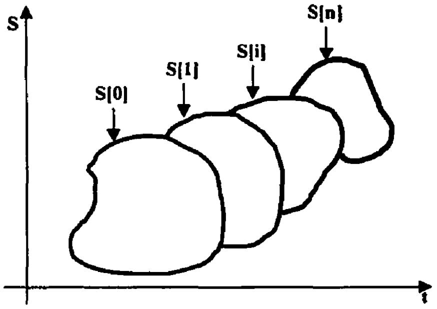


Figure 4-10 Product descriptions in evolving design process

4.3 Product Performance

In this section, we are going to discuss product performances. It includes three subsections: performances of primitive product, product performance network, and levels of complexity and abstraction of product performance.

4.3.1 Performances of primitive product

Product performance is the interaction between product and product environment. It represents product's response to actions from the environment. Figure 3-5 shows the basic form of product performances. It was also defined mathematically in Equations (3-13) and (3-18).

For the cam assembly example in Figure 4-8, with reference to the cam, the rotation of shaft and key is an action on the cam while the rotation of the cam itself is a response of the cam to the action. If the rotational velocity of the shaft is ω_s, then the rotational velocity of the cam should also be ω_s. This pair of velocities become a performance of the cam.

$$\mathbf{p}^{1}_{\mathbf{cam}} = <\mathbf{roration\_of\_shaft\_\&\_key, rotation\_of\_cam}> \quad (4\text{-}12)$$

According to Equation (3-18), they can be defined by a set of product properties.

$$\begin{aligned}
&\underline{\textbf{Action on the cam :}}\\
&\mathbf{rotation\_of\_shaft\_\&\_key} =\\
&\{\mathbf{shaft\_\&\_key\_description,\ cam\_description,\ contact\_point,}\\
&\qquad\qquad <\mathbf{rotation\_rate},\omega_S>,\ <\mathbf{rotation\_direction, clock\_wise}>\}\\
&\underline{\textbf{Response of the cam :}}\\
&\mathbf{rotation\_of\_cam} =\\
&\{\mathbf{cam\_description}, <\mathbf{rotation\_rate},\omega_S>, <\mathbf{rotation\_direction, clock\_wise}>\}
\end{aligned} \quad (4\text{-}13)$$

Moreover, if we consider the follower, one of its performances is represented as

$$\mathbf{p}^{1}_{\mathbf{follower}} = <\mathbf{rotation\_of\_cam,\ slide\_of\_follower}> \quad (4\text{-}14)$$

It indicates that the follower will slide vertically with velocity **v** following the rotation of cam with rotational rate ω_S. The rotation of cam is an action on the follower and the slide of follower becomes a response.

$$\begin{aligned}
&\underline{\textbf{Action on the follower :}}\\
&\mathbf{rotation\_of\_cam} =\\
&\{\mathbf{cam\_description,\ follower\_description, contact\_point,}\\
&\quad <\mathbf{rotation\_rate},\omega_S>, <\mathbf{rotation\_direction, clock\_wise}>\}\\
&\underline{\textbf{Response of the follower :}}\\
&\mathbf{slide\_of\_follower} =\\
&\{\mathbf{follower\_description}, <\mathbf{slide\_direction, vertical}>, <\mathbf{slide\_velocity, v}>\}
\end{aligned} \quad (4\text{-}15)$$

As was pointed out in Section 3.4.2, product performance knowledge exists to relate a product's response to actions. Figure 4-11 is the basic form of performance knowledge.

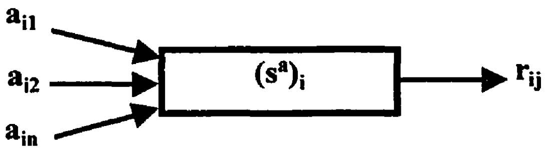


Figure 4-11 Basic form of performance knowledge

The mathematical representation is given in Equation (4-16).

$$\begin{aligned} &\forall (\mathbf{S}^a)_i \in \mathbf{S}^a, \forall \mathbf{r}_{ij} \in \mathbf{R}_i \subset (\mathbf{S}^a)_i \times \mathbf{E}, \exists \mathbf{A}_{ij}, \exists \mathbf{k}'_{ij} \\ &\mathbf{A}_{ij} = \bigcup_{p=1}^{n} \mathbf{a}_{ip} \\ &\mathbf{r}_{ij} = \mathbf{k}'_{ij}(\mathbf{a}_{i1}, \mathbf{a}_{i2}, \cdots, \mathbf{a}_{in}) \end{aligned} \tag{4-16}$$

where $(\mathbf{S}^a)_i$: a primitive product;

$\mathbf{S}^a$: primitive product set;

E: environment;

$\mathbf{A}_{ij}$: a set of actions;

n: the number of actions in the set $\mathbf{A}_{ij}$;

$\mathbf{R}_i$: a set of responses;

$\mathbf{a}_{ip}$: an action on the product from the environment, which is a subset of product properties **X** as was defined in Equation (3-18);

$\mathbf{r}_{ij}$: a response to the environment from the product, which is a subset of product properties **X** as was defined in Equation (3-18);

$\mathbf{k}'_{ij}$: a piece of product performance knowledge.

Obviously, **k'**$_{ij}$ is a many-to-one mapping. It means that under certain circumstances, the value of a product response is merely determined. In other words, a product can not respond to external actions in two different ways at the same time. The number of actions in set $\mathbf{A}_{ij}$ varies from case to case.

The objective of most scientific explorations is to find the law governing this relation by which the responses of product under certain actions can be found out. Some of the laws are deterministic, such as Newton's law, the others are nondeterministic, such as chaotic dynamics.

For the cam example, one piece of knowledge can be written as

$$\forall \mathbf{cam\_shaft\_assembly}, \vec{\omega}_{\mathbf{cam}} = \vec{\omega}_{\mathbf{shaft}} \tag{4-17}$$

However, one product could have more than one type of responses to the same external actions. Again for the cam shown in Figure 4-8, the vertical velocity of the contacting point between the cam and the roller, denoted as **v**, is a function of the cam profile C_p and the cam rotation ϖ_{cam}.

$$\forall \mathbf{cam\_shaft\_assembly}, \mathbf{v} = f(\vec{\omega}_{\mathbf{cam}}, C_p) \tag{4-18}$$

This fact can be generally represented in Figure 4-12.

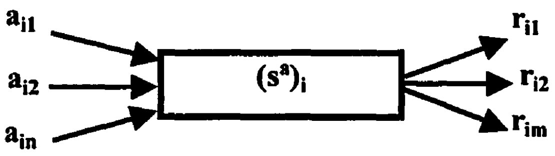


Figure 4-12 Product performances

Therefore, for each primitive product $(\mathbf{S}^a)_i$, there could be a set of performance knowledge $\mathbf{K}_i$ like that in Equation (4-16).

$$\begin{aligned} &\forall(\mathbf{S}^a)_i \in \mathbf{S}^a, \exists \mathbf{K}_i, \mathbf{K}_i = \{\mathbf{k}'_{ij} \mid \mathbf{j} = 1,2,\ldots \mathbf{n}_k\} \\ &\mathbf{R}_i = \mathbf{K}'_i \mathbf{A}_i \end{aligned} \tag{4-19}$$

where $\mathbf{K}'_i$: a set of performance knowledge associated with a primitive product;

$\mathbf{k}'_{ij}$: a piece of product performance knowledge;

$\mathbf{n}_k$: the number of known performance knowledge associated with a primitive product;

$\mathbf{A}_i$: a set of actions;

$\mathbf{R}_i$: a set of responses.

Engineering practice has established a set of such products, whose performances can be defined by performance knowledge directly. It is not necessary to turn to its components' performances. They are primitive products as was defined in Section 4.2.1. For all these primitive products, a set of performance knowledge $\mathbf{K}'$ exists.

$$\forall \mathbf{S}^a, \exists \mathbf{K}', \mathbf{K}' = \{\mathbf{K}'_i \mid \mathbf{i} = 1,2,\ldots \mathbf{n}_p\} \tag{4-20}$$

Therefore, all products with their performance knowledge constitute a set of design knowledge for solving a class of design problems. Examples are given in Section 6-2 for a rivet setting tool design problem.

4.3.2 Product performance network

In most cases it is not a trivial task to establish direct and explicit relations between actions and responses for all products. However, as was discussed in Section 4.2.2, any product can be ultimately represented by a set of primitive components. In this way, the performances of a complicated product can be studied based on the performances of primitive products included in this product. The mutual interactions among a product's constituent components establishes a product performance network such as that shown in Figure 4-13.

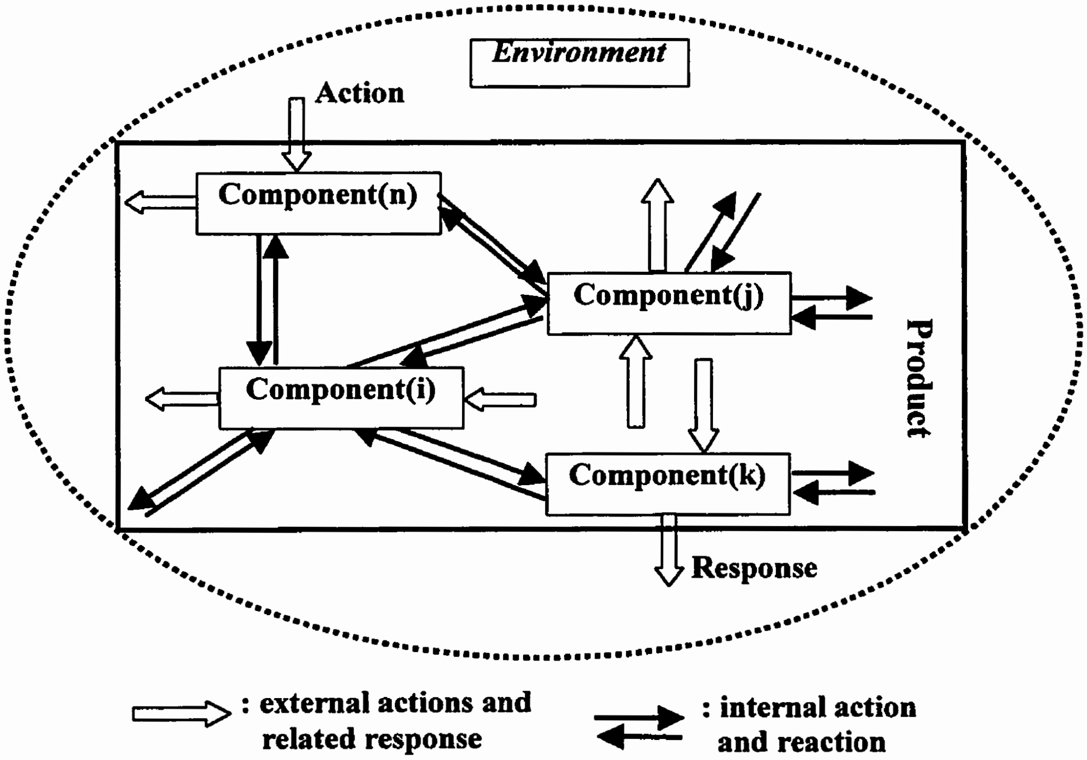


Figure 4-13 Product performance network

This network was also implied in Equation (4-6) ($\bigcup_{i=1}^{n_s}\bigcup_{\substack{j=1\\ j\neq i}}^{n_s} S_i \times S_j$). An example is given in Figure 4-14, which represents the performance of the cam assembly in Figure 4-8. The virtual rectangular embodies the product structure.

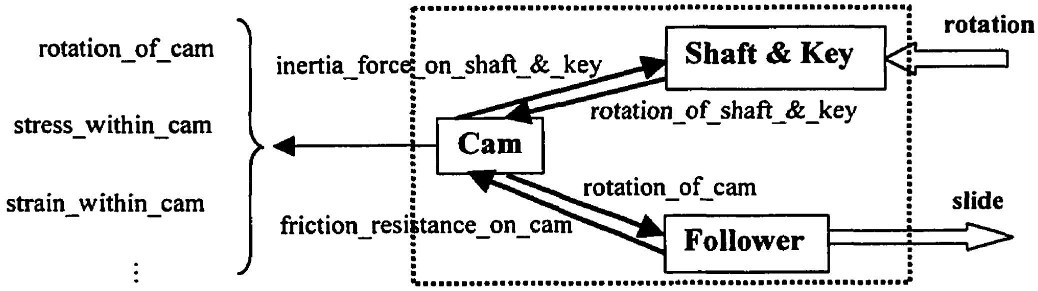


Figure 4-14 Performance network of cam assembly

Through the performance network, actions on a product are transmitted to responses to its environment. The basic block is separated to relate the fundamental performance pattern of a component to the definition of performance given in Equations (3-13) and (3-14). For the component(i) in Figure 4-13, its performance can be illustrated in Figure 4-15 by separating the group of actions on and responses from the component(i).

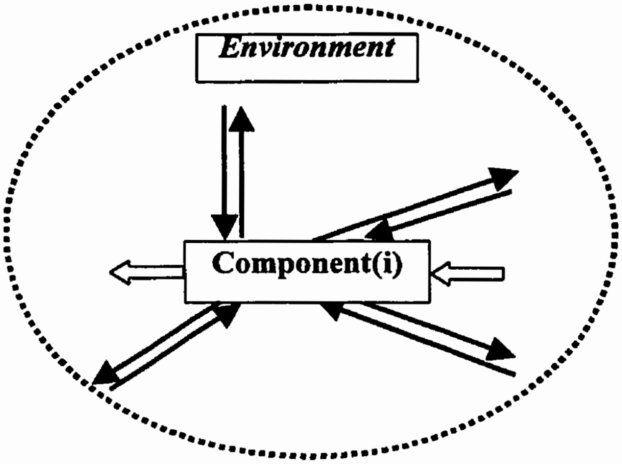


Figure 4-15 Performance of components in a product

Figure 4-16 gives a typical form of the performance of a component in product network. The expression "action(k)" is a representative of actions on component(i) with the expression "reaction(k)" as the corresponding response. The expression "action(j)" is a representative of component(i)'s actions on its connected components with the expression "reaction(j)" as the corresponding response from the connected components which in turn acts on component(i) from its connected components due to its actions on those connected components. The expression "response(i)" is a representative of the response of component(i) to its environment due to all the actions on and reactions to it.

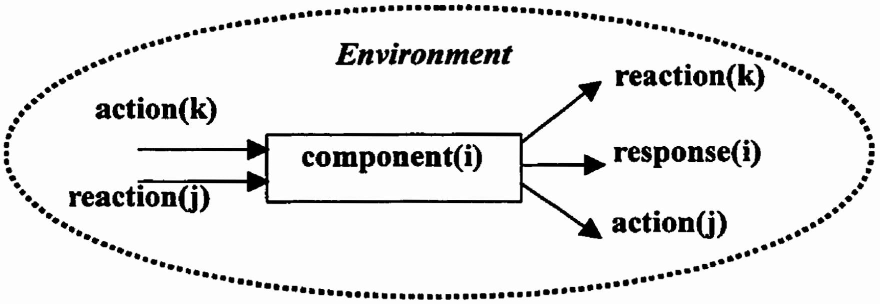


Figure 4-16 Typical constituent of influence network of components

In Figure 4-16, action(k) and reaction(j) together constitute the actions imposed on component(i) whereas reaction(k), action(j), and response(i) become the responses from component(i). The figure indeed shows the performance pattern of components included in a product. It is through its components' interactions that a product exhibits global performance.

The product performance network can be represented by simply substituting Equation (4-8) into Equation (3-13):

$$\mathbf{P[k]} \subseteq \mathbf{A} \times \mathbf{R} \subseteq (\mathbf{E} \times \mathbf{S}) \times (\mathbf{S} \times \mathbf{E}) = (\mathbf{E} \times \mathbf{S[k]}) \times (\mathbf{S[k]} \times \mathbf{E}) \tag{4-21}$$

where **k**: layer of complexity and abstraction;

P[k]: product performance in the kth layer;

E: environment;

S[k]: product descriptions in the kth layer.

This equation means that product performance also has levels of abstraction and complexity corresponding to the product descriptions.

4.3.3 Performance in different levels of complexity and abstraction

According to the definition of product performance in Equation (4-21), product performances should also be studied from three aspects: primitivity, complexity and abstraction, as was done to product descriptions. Primitivity of product performances has been discussed in Section 4.3.1.

Just as any product can be decomposed into subassemblies and components until all components become primitive products, the performance network in Figure 4-13 can also be decomposed into component performances until all performances become performances of primitive products. Similar to Equation (4-8), we have the following representation of performances

$$\begin{aligned} &\mathbf{P[0] = P(0,0,0)} \\ &\forall \mathbf{k}\ \mathbf{1 \le k \le n}\ \mathbf{P[}\boldsymbol{k}\mathbf{]} = \bigcup_{\mathbf{i}} \mathbf{P(}\boldsymbol{k}\mathbf{,}\boldsymbol{i}_{\mathbf{k}}\mathbf{,\_)} \\ &\forall \mathbf{i}_{\mathbf{n}}\ \mathbf{P(n,i_n,\_)} \in \mathbf{P_a} \end{aligned} \tag{4-22}$$

Equations (4-10) and (4-11) for the cam assembly example have actually shown how this equation supports the modeling of compound products.

According to Equation (4-22), the levels of abstraction for product performances are embodied in **P[k]**, as is given in Figure 4-17. The levels correspond to different stages of design processes. This is illustrated in Figure 4-17. It can be seen as the equivalence of Figure 4-10 in the case of product performances. From Figure 4-17 and Table 4-4, it is obvious that Equation (4-22) embodies the evolving process of product performances as the design progresses. Equations (3-13), (3-18) and (4-22) have given a complete representation of any product performance in any stage of design.

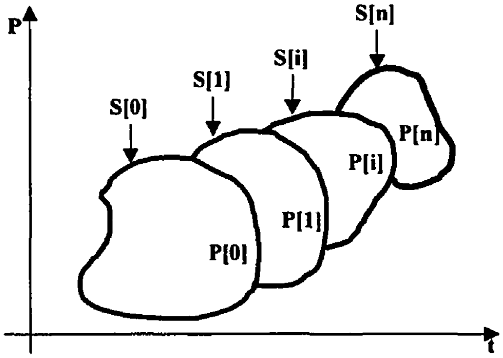


Figure 4-17 Product performances in evolving design process

Table 4-4 Product performance in different levels of abstraction and complexity

Product Description	**Action**	**Response**	**Product Performance**
S[0]	**A[0]**	**R[0]**	**P[0]**
S[1]	**A[1]**	**R[1]**	**P[1]**
S[2]	**A[2]**	**R[2]**	**P[2]**
⋮	⋮	⋮	⋮
S[n]	**A[n]**	**R[n]**	**P[n]**
Notes: index **n** is the level of complexity $\mathbf{A[n]} \subset \mathbf{X}$ $\mathbf{R[n]} \subset \mathbf{X}$			

4.4 Product Environment

Product environment is also an important part of product-environment system. But not enough attention has been given to its modeling in current literature.

Theoretically speaking, everything except the product itself can be seen as its environment. In the context of design engineering, the environment is where the product is supposed to work in. Human is one environment for many products. Ergonomics is a subject studying the nature and properties of this environment. Other environments include natural environment, function environment, financial environment, manufacturing environment, and so on. Corresponding to these different environments, product exhibits different performances according to Equation (3-13). Still for the cam assembly system, in the context of function environment, the performance is the output velocity in the follower due to the input velocity in the shaft. In the context of natural environment, however, the performance could be the weight of the follower due to the gravity field. Other performances include the change of cost of the cam assembly system with the change of market prices of raw materials, the stress distribution of the shaft under environmental forces, and so on.

In a product performance network, any product component may be subject to the actions from components that connect to it. In this network, each product component is other components' environment as is shown in Figure 4-18. As a result, the component may respond to resist the actions by acting on and/or reacting to the connected components.

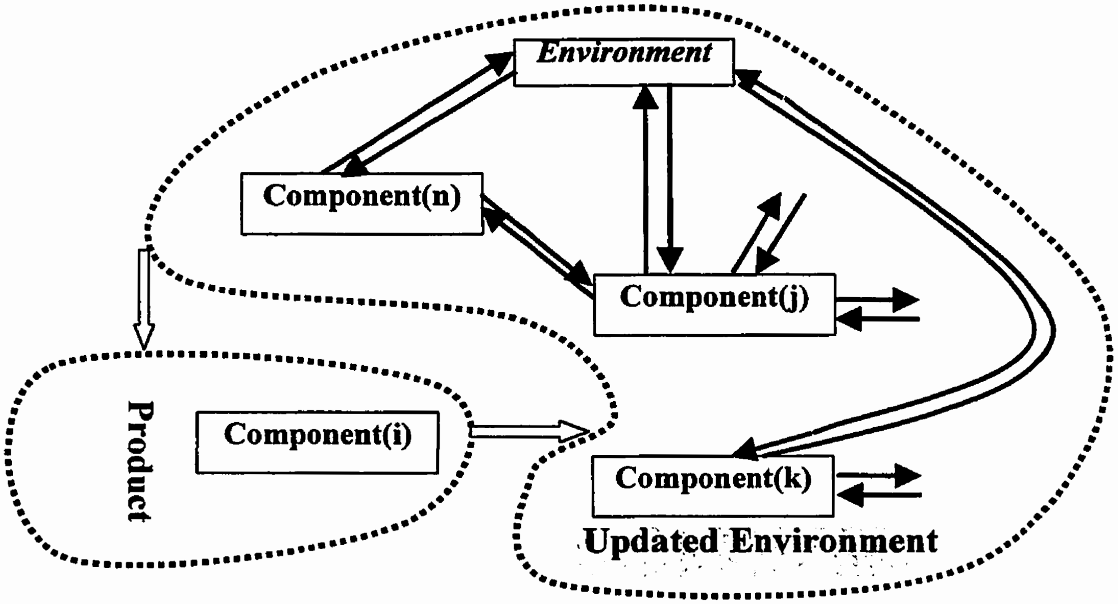


Figure 4-18 Relativity of product environment

For the cam assembly in Figure 4-8, if the focus is on the cam itself, then all other parts of the assembly will become the environment of the cam. Compared to the performance network given in Figure 4-14, an updated performance scheme is given in Figure 4-19.

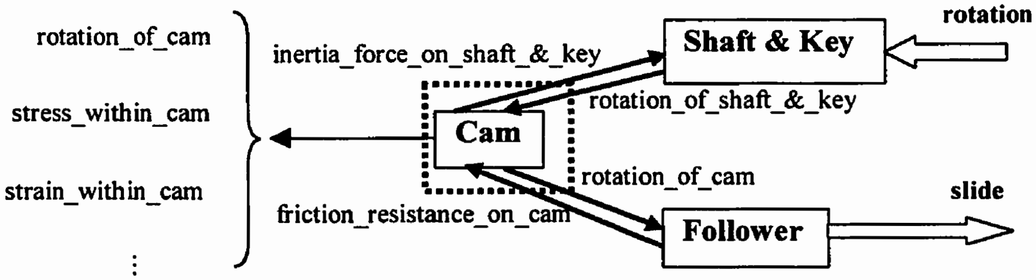


Figure 4-19 Updated performance network of cam assembly

In Figure 4-19, the follower and shaft, which were originally a part of product, become parts of the environment now. The fact means that there is no strict boundary between the definitions of product and environment. Product and environment are two relatively defined concepts. Environment can be defined by just replacing the symbol **S** with the symbol **E** in the Equations (4-1) through (4-10).

$$(\mathbf{E}^{a})_{i} = \{\mathbf{x}_{j} | \forall \mathbf{x}_{j} \in \mathbf{X}, \forall (\mathbf{x}_{j})^{n} \in \mathbf{D}^{x}, \exists (\mathbf{x}_{j})^{v} \subset \mathbf{R}^{x}, \mathbf{x}_{j} = < (\mathbf{x}_{j})^{n}, (\mathbf{x}_{j})^{v} >, 1 \leq \mathbf{j} \leq \mathbf{n}\} \tag{4-23}$$

$$\mathbf{E}^{a} = \bigcup_{i=1}^{m} (\mathbf{E}^{a})_{i}$$

$$\mathbf{E}(0,0,0) = \bigcup_{i_1=1}^{n_1} \mathbf{E}(1,\mathbf{i}_1,0) \tag{4-24}$$

$$\forall \mathbf{i}_k \ 1 \leq \mathbf{i}_k \leq \mathbf{n}_k \ \mathbf{E}(\mathbf{k},\mathbf{i}_k,\mathbf{j}_{k-1}) = \bigcup_{i_{k+1}=m_0+1}^{m_n} \mathbf{E}(\mathbf{k}+1,\mathbf{i}_{k+1},\mathbf{i}_k)$$

$$\forall \mathbf{i}_k, \mathbf{E}(\mathbf{n},\mathbf{i}_k,\_) \in \mathbf{E}_a$$

$$\mathbf{n}_0 = 1, \mathbf{n}_k = \sum_{i_k=1}^{n_{k-1}} \mathbf{n}(\mathbf{k}+1,\mathbf{i}_k), \quad \mathbf{m}_0 = \sum_{i=1}^{i_k-1} \mathbf{n}(\mathbf{k}+1,\mathbf{i}), \quad \mathbf{m}_n = \mathbf{m}_0 + \mathbf{n}(\mathbf{k}+1,\mathbf{i}_k)$$

$$\mathbf{E}[0] = \mathbf{E}(0,0,0) \tag{4-25}$$

$$\forall \mathbf{k} \ 1 \leq \mathbf{k} \leq \mathbf{n} \ \mathbf{E}[\mathbf{k}] = \bigcup_{i_k=1}^{n_k} \mathbf{E}(\mathbf{k},\mathbf{i}_k,\_)$$

$$\forall \mathbf{i}_n \ \mathbf{S}(\mathbf{n},\mathbf{i}_n,\_) \in \mathbf{E}_a$$

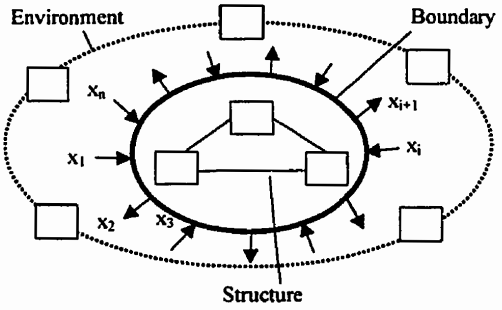


Figure 4-20 Updated product-environment system

As was indicated in Figure 3-5, the interactions between structure and environment are actions and responses. They can be seen as their mutual boundaries. Another element of the boundary is the structural properties of the contacting parts between environment and product. For the example in Figure 4-19, the contour of cam can be a boundary like this.

Graphically, an engineering system can be represented in Figure 4-20. x_i can be the action, response, or geometric contact property between product and environment. Symbolically, this product-environment boundary can represented by **B**, which is further defined as

$$\mathbf{B} = \mathbf{A} \cup \mathbf{R} = (\mathbf{E} \times \mathbf{S}) \cup (\mathbf{S} \times \mathbf{E}) \qquad (4\text{-}26)$$
$$\mathbf{B} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_n\}$$

where **B**: product-environment boundary;

$\mathbf{x}_i$: either of action, response, and product structural property;

E: product environment;

S: product descriptions;

A: actions of environment on product;

R: response of product to environment.

By considering Equation (4-25), we have

$$\mathbf{B[n]} = \{\mathbf{x}_i\mathbf{[n]} \mid \mathbf{i} = \mathbf{1}, \ldots, \mathbf{m}\} \qquad (4\text{-}27)$$

where **B[n]**: product-environment boundary in the $\mathbf{n}^{\text{th}}$ layer;

$\mathbf{x}_i\mathbf{[n]}$: product property in the $\mathbf{n}^{\text{th}}$ layer.

This means that product-environment boundary also evolves with time. Examples will be given in Chapter 5 and Chapter 6.

In engineering practice, only the boundary between product and environment is important rather than the constituents of the environment. For instance, the cam's color has nothing to do with the motion transmission given by the mechanism. As a result, the environment in most cases refer to the boundary between product and actual product environment. This thesis will follow this engineering convention whenever it does not cause confusions. But it should be noted that two symbols: **E** and **B**, will always be used to refer to real product environment and product-environment boundary so that logical inconsistency can be avoided. This shows an advantage of using mathematical approach to studying design.

4.5 Related Work

It should be noted here that the tree structure of product descriptions in this chapter is different from traditional representations in product modeling. The first and the most essential point lies in that the base of the present representation is the property set which might be point set (geometrical information) or concept set (feature information, physical information and any other linguistic information). But the point set in topology is the base of CSG solid modeling. Another profound difference is that the traditional product modeling can only support the representation of product information after the product has been designed whereas the present approach represents a product along the whole dynamic design process from design concept to final detailed geometry and dimensions. The third difference lies in that component connectors are viewed as a product component which are separately handled in many other similar models. This minor departure makes it much easier to get a uniform mathematical representation to support the dynamic design process.

4.6 Summary

This chapter formulated product descriptions, product performances, and product environment in terms of two requirements from supporting design process: one is evolution nature of design and another is the finite representation of infinite number of possible products. Three items were represented in the same scheme: primitive objects, combination rules and compound objects. The scheme embodies the levels of complexity and abstraction. It will be shown in later chapters that this scheme supports the design process naturally. Chapter 5 will deal with another design object: design requirements, based on the model of product descriptions and product performances presented in this chapter. Both Chapters 4 and 5 will be the base of the design process model in Chapter 6.

Chapter 5 Formulation and Formalization of Design Requirements

5.1 Introduction

This chapter discusses the formulation and formalization of the other end of the design process shown in Figure 4-1: design requirements. Design requirements are constraints on a product design so that the designed product can be manufactured to achieve its desired functions in its working environments. They can be motives or demands for creating a completely new product, complaints about the performance of existing products, or the failure due to malfunctions of existing products. The first step in product design is to define design requirements by specifying customer needs and wants (Otto, 1996). However, because of the complexity and randomness of this process due to too many human factors involved, the formulation and formalization of this process is extremely difficult and is not the concern of this thesis. Instead, the focus of this chapter will be on the formal modeling of the results from this process: design requirements. It is the starting point for the design process model in Chapter 6.

In the current literature, there have been many approaches to modeling design requirements. Among all those approaches, there are two extreme cases. One is from and for design engineering. Another is for the scientific exploration of design. Most of the research from design engineering has been focusing on documenting design practice of design engineers. The terminology and jargon have been kept as closely as possible to what they are in real design practice. On the other hand, however, scientific approach focuses more on the formal representation and modeling of the problem. Engineering representation helps engineers to describe design problems whereas formal representation supports a more logical and rationalized realization of the design process. A successful scientification of design must be able to transform the engineering representation to the formal representation.

This chapter includes six sections. In section 5.2, a running example is given to illustrate the concepts in this chapter and Chapters 6 through 8. Section 5.3summarizes the engineering representation of design requirements distributed in literature. A formal representation of design requirements is proposed in Section 5.4, followed by a normalization process of design requirements to transform engineering representation of design requirements into formal representations. Section 5.6 is a summary of this chapter.

5.2 Running Example

This section gives a running example to help illustrate the concepts proposed in chapters 5 through 8. The example is about the design of a rivet setting tool, which is adopted from the book by Hubka *et al*(1988).

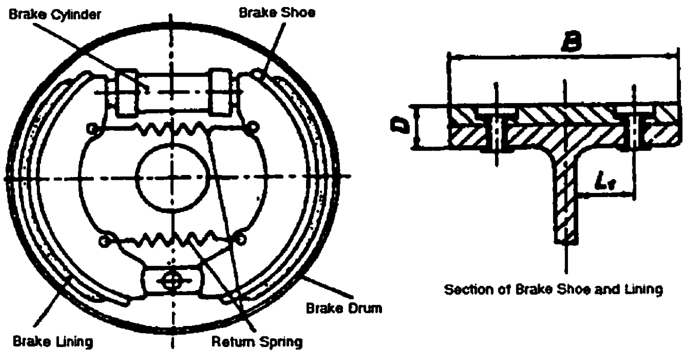
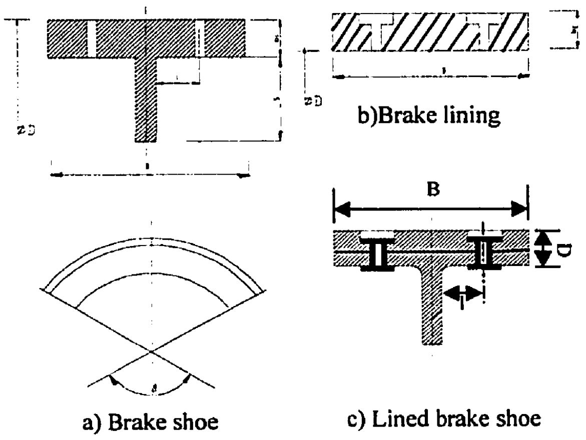


Figure 5-1 Internal drum brake(Hubka *et al*,1988)

The task of this problem is to design a tool for riveting brake linings onto brake shoes for internal drum brakes as shown in Figure 5-1.

Figure 5-2 gives the details and dimension of brake shoe, brake lining and rivets.



Data:

- Brake drum:
  inside diameter = 250~400mm
- Rivet to DIN standard 7338
  Shank diameter = 3~5mm
  Material: C_u
- Maximum shoe width
  B_{max} = 100mm
- Maximum total thickness of parts:
  D_{max} = 15mm
- minimum distance between rivet center line and rib
  L_{1min} = 100mm
- Riveting stroke: ~3mm
- Riveting force: ~500N

Figure 5-2 Form of brake shoe and lining(adapted from Hubka *et al*,1988)

The data are summarized in Table 5-1.

The following is a list of the design requirements for this design problem:

R-1: Functional requirements:

R-1-1. To rivet brake lining into brake shoe

Table 5-1 Dimensions of brake shoe and brake lining

	Brake Drum	**Brake Lining**	**Brake Shoe**	**Rivet**
ϕD	250~400mm	250~400mm	250~400mm	3~5mm
β_{max}	150°	N/A	150°	N/A
B	80~100mm	80~100mm	80~100mm	N/A
D_{max}	15mm	N/A	N/A	N/A
b_1	N/A	N/A	6~8mm	N/A
b_2	N/A	6~8mm	N/A	N/A
s_{max}	N/A	N/A	50mm	N/A
l_{min}	N/A	N/A	10mm	N/A
Stroke	N/A	N/A	N/A	4mm
Material	N/A	Special friction lining	Al-alloy	Al-alloy
Maximum mass	N/A	200g	2kg	N/A
Maximum Force	N/A	N/A	N/A	500N

R-2: Physical requirements:

R-2-1. The form and dimension of the tool must be consistent with the form and dimension of brake shoe and brake lining, which is given in Figure 5-2 and Table 5-1.

R-3: Ergonomic Requirements

R-3-1. User: car mechanic

R-3-2. Hand force: ~200N

R-3-3. Foot force: ~400N

R-3-4. Working height: follow ergonomic standards

R-3-5. Safety against accidents: follow related industry standards

R-4: Operational Requirements

R-4-1. Service life 5 years

R-4-2. Good transportability

R-4-3. Maintenance free

R-5: Appearance requirements

No special requirements

R-6: Manufacturing requirements

R-6-1. Manufacturable in workshop of ... Co. Ltd

R-7: Cost: Financial Requirements

R-7-1. Maximum manufacturing costs: CAD$190

5.3 Engineering Definition of Design Requirements

In design practice, design requirements come from the clarification of design tasks. From the product life cycle point of view, any product design must take into account a number of requirements regarding functionality, safety, manufacturability, assembly, testing, shipping, distribution, operation, services, re-manufacturing, recycling and disposal (Gu and Sosale, 2000). They are necessary functions and task-specific constraints, which can be listed under the following headings (Pahl and Beitz, 1988): geometry, kinematics, forces, energy, material, signal, safety, ergonomics, production, quality, control, assembly, transport, operation, maintenance, costs, and schedules. Basically, there are only two types of design requirements: structural requirements and performance requirements, as is shown in Figure 5-3. For the running example, structural requirements include R-2-1, R-3-1, R-3-2, R-3-3, and R-3-4 whereas performance requirements include R-1-1, R-3-5, and R4 to R7.

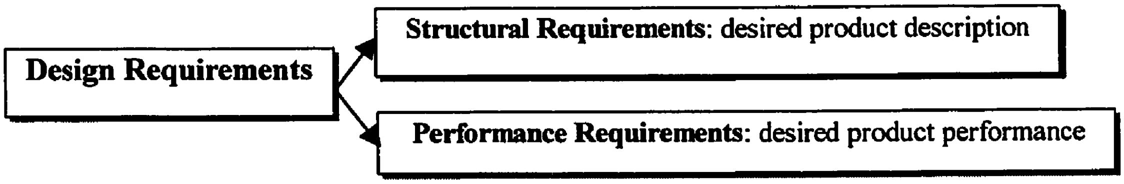


Figure 5-3 Classification of design requirements

Among the two design requirements in Figure 5-3, the requirements regarding product descriptions are very much straightforward, and can be directly represented as the constraints on product structure in the way given in Equation (3-22). Symbolically, they can be defined as

$$\mathbf{r}^{\mathbf{d}} = \lambda(\mathbf{x},[\mathbf{x}]), \mathbf{x} \in \mathbf{X}^{\mathbf{s}} \tag{5-1}$$

where $\mathbf{X}^{\mathbf{s}}$: a set of product structural properties, defined in Equation (5-14);

$\mathbf{x}$: product structural property;

$[\mathbf{x}]$: constraint on product structural property $\mathbf{x}$;

λ: logic operator such as $=, \geq, \leq$;

$\mathbf{r}^{\mathbf{d}}$: design requirement.

For the running example, the distance between the brake lining and brake shoe assembly and the ground should equal to the comfortable working height. Denote the working height and comfortable working height range as $\mathbf{h}$ and $[\mathbf{h}_{\mathbf{w}}^{\mathbf{l}}, \mathbf{h}_{\mathbf{w}}^{\mathbf{u}}]$, respectively, the design requirement can be represented as

$$\mathbf{h} \in [\mathbf{h}_{\mathbf{w}}^{\mathbf{l}}, \mathbf{h}_{\mathbf{w}}^{\mathbf{u}}] \tag{5-2}$$

where $\mathbf{x} = \mathbf{h}$, product structural property;

$[\mathbf{x}] = [\mathbf{h}_{\mathbf{w}}^{\mathbf{l}}, \mathbf{h}_{\mathbf{w}}^{\mathbf{u}}]$: constraint on product structural property;

$\lambda =$ '$\in$': logic operator.

It should be noted here that the requirement can also be defined as a property $\mathbf{x[0]}$ in terms of Equation (3-2):

$$\begin{aligned} &\mathbf{x}^{n} = \mathbf{h} \\ &\mathbf{x}^{v} = [\mathbf{h}_{w}^{l}, \mathbf{h}_{w}^{u}] \\ &\mathbf{x[0]} = < \mathbf{x}^{n}, \mathbf{x}^{v} > \end{aligned} \tag{5-3}$$

If the final height is decided, say $\mathbf{h}_{w}$, then this property can be further defined as

$$\begin{aligned} &\mathbf{x}^{n} = \mathbf{h} \\ &\mathbf{x}^{v} = \mathbf{h}_{w} \\ &\mathbf{x[1]} = < \mathbf{x}^{n}, \mathbf{x}^{v} > \end{aligned} \tag{5-4}$$

Compared to Equations (4-1) and (4-2), these two equations correspond to the properties of product type and product. In other words, product type can be taken as the constraint on product.

The requirements regarding product performance have long-since been described by functions. The modeling of function in an engineering context has been addressed by many researchers. These include: the definition and use of function in the mechanical design process by Grabowski and Benz (1989), Pahl and Beitz (1988), and Ullman (1992); the structure of function representation by Kota and Lee (1990), Mashburn and Anderson (1991), and Rane and Issac (1990). However, the understanding of functional modeling is still confusing and can not be represented directly in the way given in Equation (3-22). Among many approaches to modeling functions, two major function models are widely accepted in design research community. One model defines function as a relation between input and output of energy, material, and information (Pahl and Beitz, 1988). Another model represents a function in the form of verb-noun phrase (Miles, 1972). Function words are the core of these two definitions. Since the definitions of product function and functional knowledge are subjective and domain-dependent in nature, ontology has been an important research subject in design community (Kirschman *et al*, 1996). The input-output approach has the advantage to maintain the relationships among functions of components in a product by functional structure of the product. A general structure of this representation can be illustrated with an example in Figure 5-4. V_i is a verb in the figure. For the riveting tool design example, the functional structure based on motion analysis and force analysis is given in Figure 5-5.

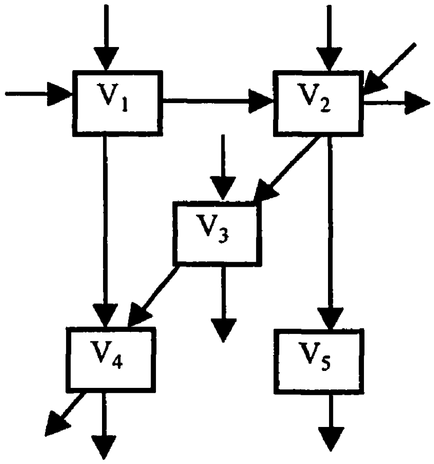


Figure 5-4 Sample functional structure

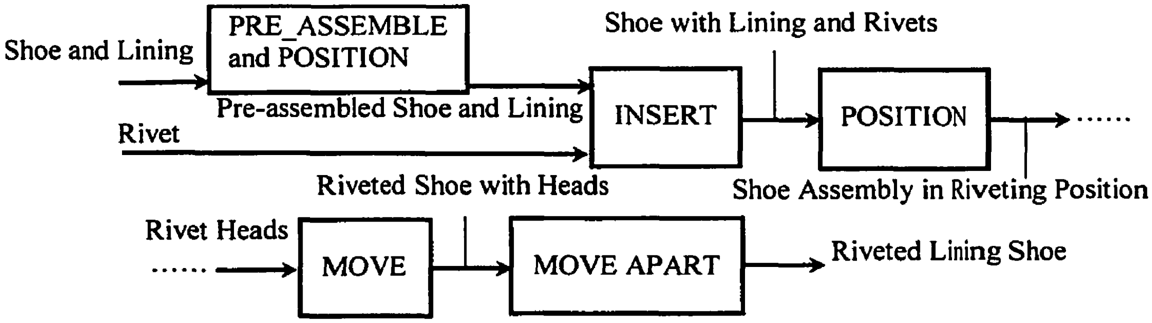


Figure 5-5 Functional structure of rivet setting tool (adapted from Hubka *et al*,1988)

But it is argued that the input-output model lacks the expressive power to describe the functions of many mechanical devices such as "to ENABLE Insertion of shoes", "to ENABLE transport", and so on, which have no explicit input and output in them.

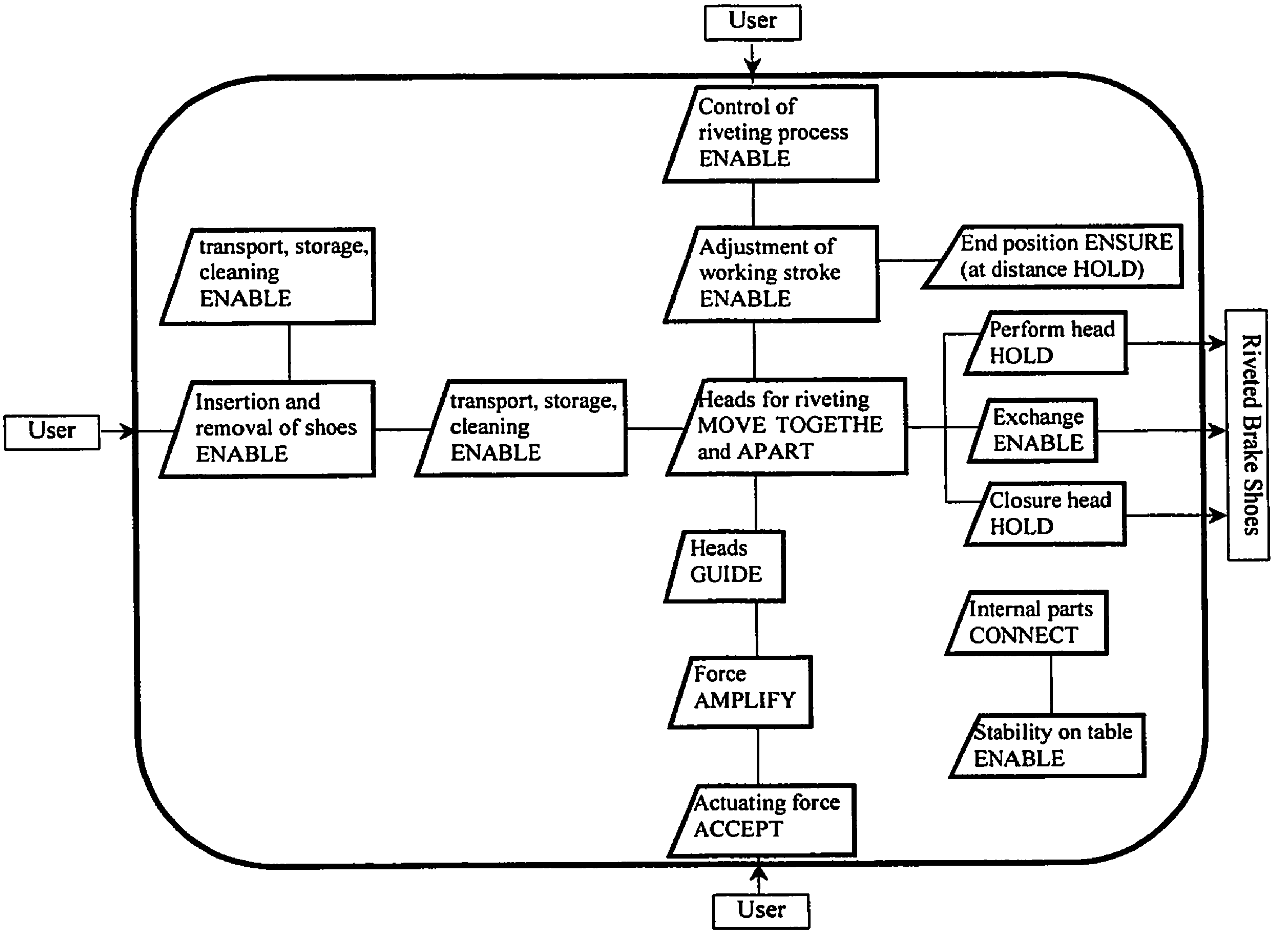


Figure 5-6 Subfunctions of rivet setting tool (adapted from Hubka *et al*,1988)

The "to verb noun" approach represents functions very clearly and in a fashion familiar to designers. For example, functions of shaft include those such as "to transmit power" and "to support pulleys, sprockets, gears, cranks, levers, and other attachments". The drawbacks, however, include the lack of a standard vocabulary, poor maintenance of functional relationships in a function structure, and the lack of a formal method for decomposition. Figure 5-6 is such a representation of rivet setting tool. The mutual relations among functions are not clear.

Recently, Lossack *et al*(1998) attempted to solve this problem by capturing the knowledge responsible for transforming design requirements (in some cases, functions) into product descriptions or behavior to obtain a function structure based on Umeda and Tomiyama (1996)'s work on a function-behavior-state model. Still function words are the core of this approach. A similar idea was proposed by Altshuller (1988) where behaviors were connected to functions by means of a library. About 30 function words were used.

From the above discussions, it can be seen that current functional modeling approaches have the following problems:

- Functional modeling is not consistent with structural requirements, so separate representation schemes are required for structural and performance requirements.
- The modeling of function and in turn functional structure is still an open question. This makes concept generation challenging.

5.4 Formal Representation of Design Requirements

To solve the problems listed at the end of Section 5.3, the representation of design requirements in Equation (3-22) will be refined, based on which functional requirements would also be able to be represented in the same scheme. Equation (5-5) rewrites Equation (3-22) to start this refining process

$$\mathbf{r^d = \lambda(x,[x]), x \in X} \tag{5-5}$$

where $\mathbf{r^d}$: design requirement;

$\mathbf{x_i}$: product property, structural as well as behavioral;

$[\mathbf{x_i}]$: constraint on product property;

λ: relational operator, such as =, <, >, and so on.

In this way, a design requirement is represented as a predicate. Equation (5-5) assigns a Boolean value to design requirement predicate $\lambda(\mathbf{x},[\mathbf{x}])$.

$$\mathbf{r^d} = \begin{cases} \mathbf{1} & \textbf{if } \mathbf{r^d} \textbf{ is satisfied} \\ \mathbf{0} & \textbf{if } \mathbf{r^d} \textbf{ is unsatisfied} \\ \mathbf{-1} & \textbf{if } \mathbf{r^d} \textbf{ can not be decided for satisfaction} \end{cases} \tag{5-6}$$

Boolean values 0, 1 and –1 represent false, true and undetermined, respectively.

Then, design requirements for a design problem can be written as

$$\begin{aligned} \mathbf{R^d} &= \{(\mathbf{r^d})_i \mid \mathbf{i} = \mathbf{1}, \cdots \mathbf{n_r}\} \\ (\mathbf{r^d})_i &= \lambda_i(\mathbf{x_i}, [\mathbf{x_i}]), \mathbf{x_i} \in \mathbf{X} \end{aligned} \tag{5-7}$$

where $\mathbf{R^d}$ is a set of design requirements;

$(\mathbf{r^d})_i$: design requirement;

$\mathbf{x_i}$: a product property;

$[\mathbf{x}_i]$: constraint on product property $\mathbf{x}$;

λ_i: relational operator, such as =, >, <;

$\mathbf{n}_r$: the number of design requirements for a design problem.

In Equation (5-7), $\mathbf{x}_i$ can only assume either structural or behavioral property of the product to be designed. However, $[\mathbf{x}_i]$ could be one of four different elements: 1) constraint on the action imposed on the product by the environment; 2) constraint on the response to the environment by the product; 3) constraint on the structural property of the product from the contacting environment element; 4) constraint on the structural property of the product by other requirements.

The design process is to evolve the set of design requirements, product descriptions, as well as product environment until all design requirements are satisfied, as is shown in Figure 5-7.

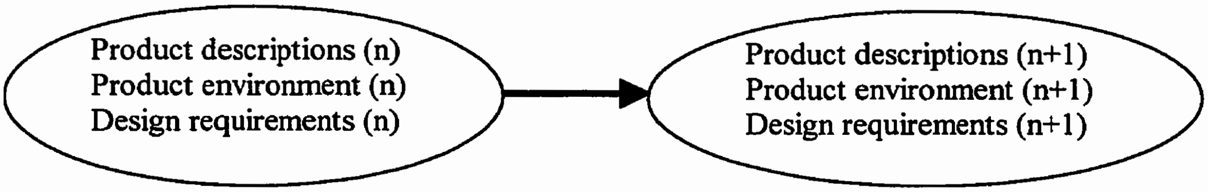


Figure 5-7 Evolution process of design

In the above evolution process, two notes should be made: 1) the earlier environment and product description can be seen as the constraints and requirements on the later design. This can be seen from Equations (5-3) and (5-4). 2) The accomplished partial product descriptions can be seen as a part of environment for the later stage of design. In this way, design requirements are always constraints on the boundary elements. Therefore, design requirements can be seen as the current state of the design process, which is the only set of properties known to designers at each step of the design process. They constrain the properties to be designed. Denoted a current property as $\mathbf{x}_i[\mathbf{n}]$, it is refined as $\mathbf{x}_i[\mathbf{n+1}]$ in a later design stage. Then a design requirement can be defined as:

$$\forall \mathbf{x}_i[\mathbf{n}] \in \mathbf{B}[\mathbf{n}],\ \exists \mathbf{x}_i[\mathbf{n}+\mathbf{1}] \in \mathbf{B}[\mathbf{n}+\mathbf{1}],\quad (\mathbf{r}^d)_i = \lambda_i(\mathbf{x}_i[\mathbf{n}+\mathbf{1}], \mathbf{x}_i[\mathbf{n}]) \tag{5-8}$$

where **B[n]**, **B[n+1]** is defined in Equation (4-27).

The difference between $\mathbf{x}_i[\mathbf{n}]$ and $\mathbf{x}_i[\mathbf{n+1}]$ can be further defined as

$$\begin{aligned}
&\forall \mathbf{x}_i[\mathbf{n}] \in \mathbf{B}[\mathbf{n}], \exists \mathbf{x}_i[\mathbf{n}+\mathbf{1}] \in \mathbf{B}[\mathbf{n}+\mathbf{1}],\\
&\mathbf{x}_i[\mathbf{n}] =< (\mathbf{x}_i[\mathbf{n}])^n, (\mathbf{x}_i[\mathbf{n}])^v >\\
&\mathbf{x}_i[\mathbf{n}+\mathbf{1}] =< (\mathbf{x}_i[\mathbf{n}+\mathbf{1}])^n, (\mathbf{x}_i[\mathbf{n}+\mathbf{1}])^v >\\
&(\mathbf{x}_i[\mathbf{n}+\mathbf{1}])^v \in (\mathbf{x}_i[\mathbf{n}])^v
\end{aligned} \tag{5-9}$$

According to Equation (5-8), any design requirement is transformed into constraint on either actions, or responses, or physical properties of product-environment boundary element. Therefore, considering the evolution nature of design process, the design requirements in Equation (5-7) can be represented as a set of environment elements:

$$\begin{aligned}
&\forall \mathbf{R}^d, \mathbf{R}^d = \mathbf{B}[\mathbf{n}]\\
&\mathbf{B}[\mathbf{n}] = \{\mathbf{x}_i[\mathbf{n}] \mid \mathbf{i} = \mathbf{1}, \cdots, \mathbf{n}_r\}
\end{aligned} \tag{5-10}$$

The evaluation of these design requirements depends on later design stages according to Equation (5-8).

Still for the riveting design example, to formulate design requirements, the product-environment system at the beginning of the design process needs to be formulated. It is shown in Figure 5-8.

The environment can be written as

$$\mathbf{B} = \{\mathbf{F}_f \cup \mathbf{F}_h, \mathbf{F}_r^u, \mathbf{F}_r^d, \mathbf{W}_b, \mathbf{G}_b, \mathbf{h}_w, \mathbf{x}_{mf}, \mathbf{x}_f, \mathbf{x}_s, \mathbf{x}_t, \mathbf{x}_{mt}\} \tag{5-11}$$

where $\mathbf{F_f}$: foot force;

$\mathbf{F_h}$: hand force;

$\mathbf{F}_r^u, \mathbf{F}_r^d$: riveting forces;

$\mathbf{W_b}$: weight of brake shoe, lining and rivets assembly;

$\mathbf{G_b}$: geometric model of brake lining and brake shoe;

$\mathbf{h_w}$: comfortable operation height.

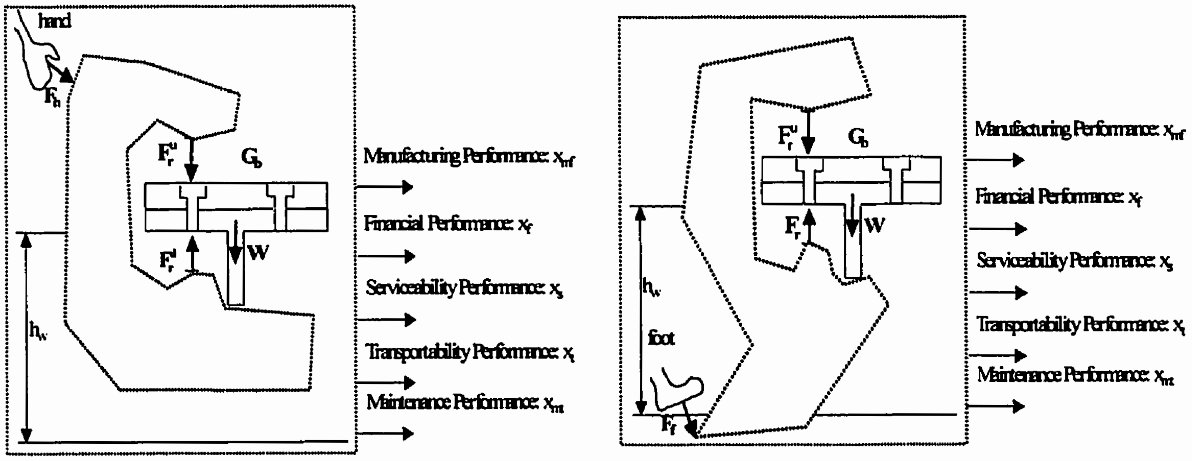


Figure 5-8 Initial environment for rivet setting tool design problem

All design requirements can be represented in Table 5-2 and Table 5-3 as follow:

Table 5-2 Performance requirements

Type	**Input**	**Output**
Functional	$\mathbf{F_h}$ **or** $\mathbf{F_f, F_h \leq 200N, F_f \leq 400N}$	$\mathbf{F_r^u, F_r^d, [F_{rmin}] \leq F_r^u, F_r^d \leq [F_{rmax}]}$
Financial	market information	x_f, $x_f \leq$CAN$190.0
Manufacturing	Manufacturing factors	Error between design and product

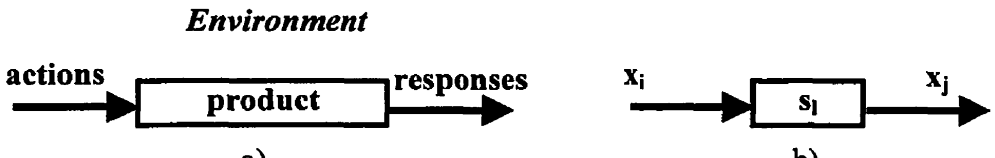
Table 5-3 Structural requirements

Type	**Environment**	**Product Descriptions**
Physical	geometric model of brake lining and shoe	Forms and dimensions of closure and perform heads
Ergonomic	Comfortable working height	h_w

5.5 Normalisation of Design Requirements

Section 5.4has established a uniform representation scheme of design requirement. However, it is not realistic to force design engineers to define a design problem in this way. Therefore, it is necessary to develop an approach to transforming design requirements from engineering representation to formal representation so that functional requirements can also be embodied in the scheme. This is called requirement normalization process. Through this process, design requirements are represented as a subset of environment **B**.

The basis of the formal representation of design requirements is the performance model of a product, which is shown again in Figure 5-9. Therefore, the objective of normalizing a functional requirement is to reduce the functional models given in Section 5.3.



a) b)

Figure 5-9 Basic form of product performance

The major difference between performance model and existing function models lie in that function is defined centered on a function word which is a verb, while performance is defined centered on a product which is a noun. As a result, in function-based design, a function structure precedes product structures. In most cases, function structure and product structure are not isomorphic. In performance-based design, performance network and product structures share the same structure. Therefore, instead of generating a functional structure for further design decomposition, functions are transformed into performances so that a performance network is handled in the later design stages. This provides a way to avoid the complexity and difficulty for solving a conceptual design problem.

To transform the design requirements in function based engineering representation into that in performance-based representation, those two cases in Section 5.3need to be processed: input-output function model and to verb-noun model.

The basic form of input-output function model is shown in Figure 5-10. It has the same form and structure as performance model.

Figure 5-10 Input-output function model

For the lever shown in Figure 5-11, the performance model is given in Equation (5-12).

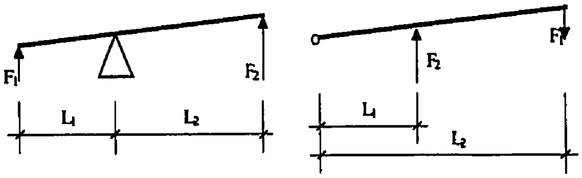


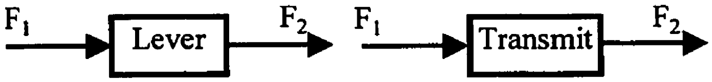
Figure 5-11 Lever

$$\forall \mathbf{lever}, \mathbf{F}_2 = \frac{\mathbf{L}_2}{\mathbf{L}_1}\mathbf{F}_1 \wedge \mathbf{F}_1 \rightarrow \mathbf{F}_2 \tag{5-12}$$

Meanwhile, the function model is given in Equation (5-13).

$$\forall \mathbf{lever}\, \exists \mathbf{transmit} \subseteq \mathbf{F}_1 \times \mathbf{F}_2 \tag{5-13}$$

They can be illustrated graphically in Figure 5-12.



a) Performance model b) Function model

Figure 5-12 Function and performance models of lever

Obviously, the name of the above relation is artificial, subjective, and domain dependent. The word 'transmit' can be replaced by 'change to' and many others. As will be seen in Chapter 6, only actions and responses as well as structure will be involved in solving design problems. The name of performance is not crucial. This is different from function based approaches.

However, there is fundamental difference between these two representations. In input-output function model, the input and output can be anything, including all those from product property set. Performance model, on the other hand, only deals with the output that is related to the properties of considered product. For the riveting tool design example, the function was defined in Figure 5-13 with input-output function model. These input and output have nothing to do with the properties of the rivet setting tool. Therefore, they can not be the input and output in the performance based representation.

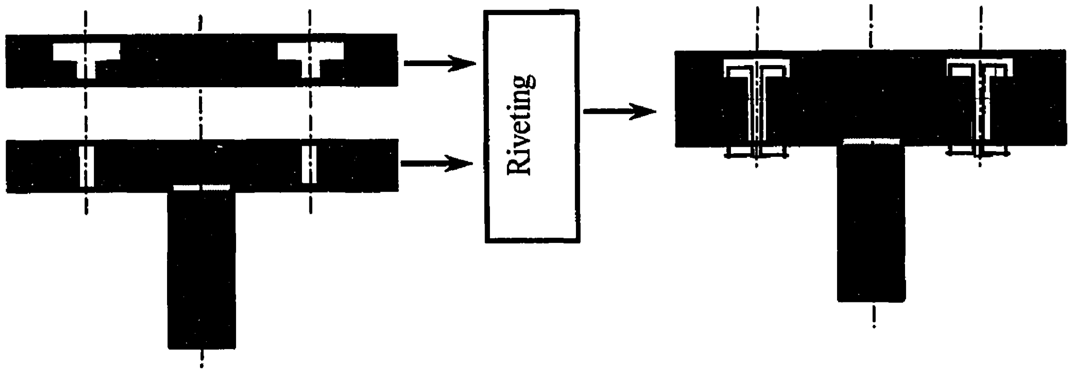


Figure 5-13 Function of rivet setting tool with input-output function model (Hubka *et al*,1988)

A black box diagram of riveting was given to illustrate the overall functionality of the rivet setting tool (adapted from Hubka *et al*,1988), as is shown in Figure 5-14.

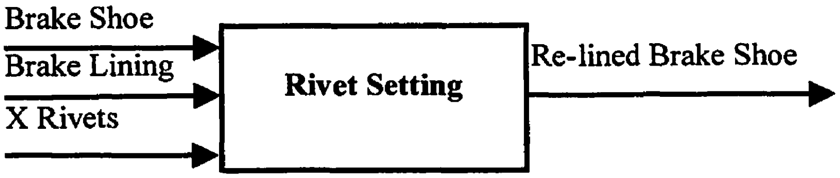


Figure 5-14 Overall functionality of rivet setting tool (adapted from Hubka *et al*,1988)

However, with the performance-based approach, this function can be described in Figure 5-15. Each output is a property of the related product.

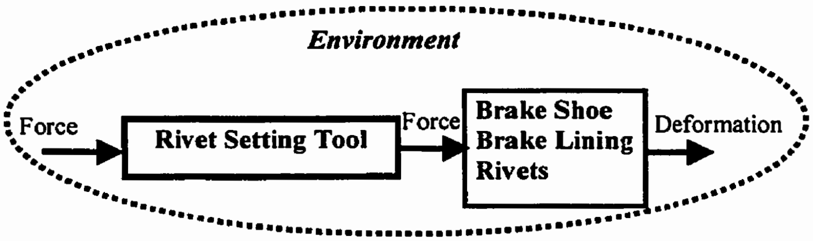


Figure 5-15 Active function of rivet setting tool in environment

The "to verb-noun" function model can be divided into two categories: passive and active. Passive functions show how a product should respond to external actions by accepting or allowing the actions whereas active functions show how a product acts on other products.

Passive functions have the pattern "to resist external actions passively". It is shown in Figure 5-16. The actions may come from human being, other external agencies, or other parts of the product.

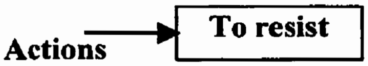


Figure 5-16 Pattern of passive function

To transform it into the form of product performance, the components involved need to be studied. Obviously, in this case, only one component is involved. It can be transformed into the normalized performance scheme shown in Figure 5-17 directly, which can be read as "this type of product can resist, accept, or allow xx action by giving xx response"

Figure 5-17 Pattern of passive function

For example, one of rivet's function is "to deform under external forces. This function is illustrated in Figure 5-18.

Figure 5-18 A function of rivet

Active functions have the pattern "to act on the surrounding products". Two components are involved. The general form of this function can be represented as in Figure 5-19.

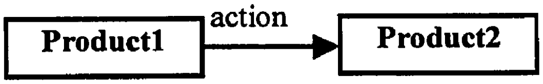


Figure 5-19 Pattern of active function

They can be transformed into two passive functions by adding the missing input and output as in Figure 5-20. In the figure, "Response 1" becomes an action on "Product 2". This is the simplest product performance network.

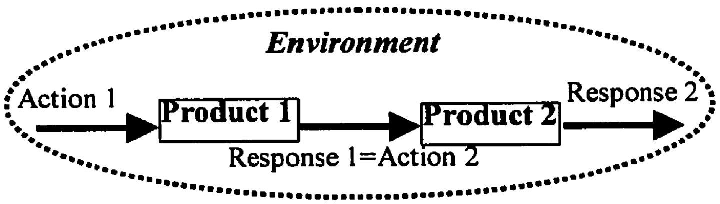


Figure 5-20 Active function

Still for the rivet example, it has another function "to connect components", as in Figure 5-21.

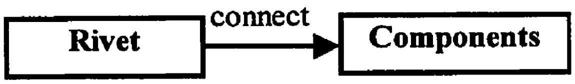


Figure 5-21 "Connect" function of rivet

Based on this framework, the function of rivet is shown in Figure 5-22.

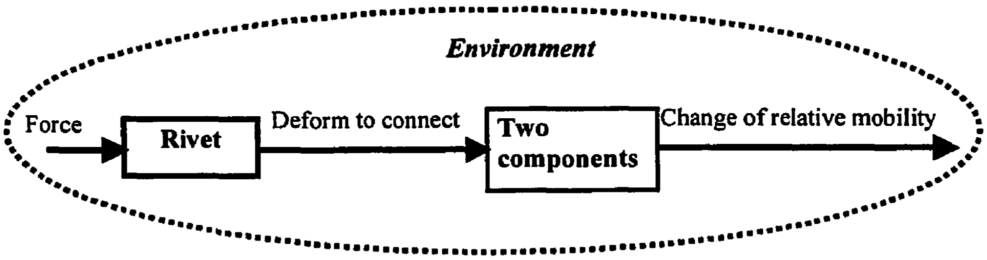


Figure 5-22 Active function of rivet

Obviously, it is difficult to define this type of function of "product 1" in terms of input-output approach. However, it can be interpreted as the addition of two performance schemes. Here, both input and output are physical effects which are the interactions between objects.

With the above manipulation, any functional requirement can be transformed into performance based formal design requirements.

Table 5-4 gives the normalized requirements of riveting tool design example.

Based on the above result, the initial design requirement of this example is shown in Figure 5-8 and given in Equation (5-14).

$$\mathbf{B} = \{\mathbf{F}_f \cup \mathbf{F}_h, \mathbf{F}_r^u, \mathbf{F}_r^d, \mathbf{W}_b, \mathbf{G}_b, \mathbf{h}_w, \mathbf{x}_{mf}, \mathbf{x}_f, \mathbf{x}_s, \mathbf{x}_t, \mathbf{x}_{mt}\} \qquad (5\text{-}14)$$

By comparing Table 5-4 to Table 4-2, it can be seen that an abstract product is defined by those parameters.

$$\mathbf{S} = \{< \mathbf{h}, [\mathbf{h}_w^l, \mathbf{h}_w^u] >\} \qquad (5\text{-}15)$$

Those parameters are not well defined in the beginning of the design process.

Table 5-4 Formal representation of design requirements in rivet setting tool design

R^d	λ	X	[X]
R-1-1	$\leq$	$\mathbf{F}_r^u, \mathbf{F}_r^d$	400N
R-2-1	Fit-in	Perform head, closure head	Shape and dimensions of contacting parts among rivet, brake shoe, and brake lining
R-3-2	$\leq$	F_h	200N
R-3-3	$\leq$	F_f	400N
R-3-4	$\approx$	h	h_w
R-3-5	follow	Failure mode	Safety standards
R-4-1	$\geq$	Service life	5 years
R-4-2	is	transportability	good
R-4-3	is	maintenance	free
R-6-1	is	manufacturability	good
R-7-1	$\leq$	Cost	CAD$190.00

5.6 Summary

A unified representation of design requirements was developed in this chapter. This representation scheme defines a design requirement as a constraint on a property of product. A functions is defined in this way by transforming its engineering representation into input-output performance representation. Through this approach, a set of design requirements is mapped into a set of environment **B**. It will be shown in the next chapter that this model together with the models for product descriptions and product performances in Chapter 4 result in a new conceptual design process model.

Chapter 6 Formulation and Formalization of Design Process

6.1 Introduction

This chapter focuses on modeling the design process illustrated in Figure 4-1 to complete a formal model of design. A product design process can be divided into conceptual, configuration, and detailed design phases. The main objective of conceptual design is to develop concepts to meet design requirements. Configuration design refines design concepts to concrete product architectures and components. Key design parameters for critical design features are also determined at this stage. Detailed design determines all detailed parameters including dimensions, tolerances and other design parameters of all components where a product is described by engineering drawings or geometric models (Gu, 1998). Correspondingly, three types of product descriptions are involved: conceptual, configuration, and detailed design.

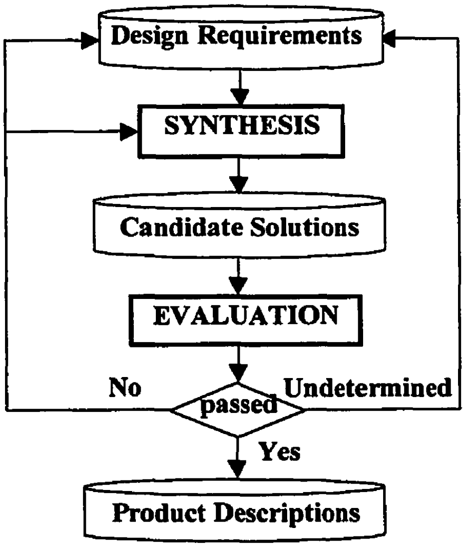


Figure 6-1 Basic design process

Each of the above design sub-tasks is generally accomplished in two phases: synthesis and evaluation, as is shown in Figure 6-1. At the beginning of the design process, a set of design requirements are given, which are defined with respect to the description of the product to be designed. To satisfy these requirements, some possible design proposals are generated for further justification against the requirements. If a design proposal satisfies the requirements, then it might be accepted as a design solution. If the product description is not detailed enough to be evaluated, then it can be kept as a potential solution and should be refined further. Otherwise the proposal must be modified or new design proposals should be created or the whole design problem should be reformulated.

Conceptual design is one of the critical design stages where some of the most important design decisions are made. Models and methods have been proposed to generate design concepts, based mostly on the systematic design approach. Conceptual design process is generally divided into the following steps: 1) identify design requirements, 2) establish function structures, 3) search for solution principles for fulfilling the subfunctions, 4) combine solution principles to fulfill the overall function, 5) select suitable combinations to define design concepts, and 6) evaluate concepts against technical and economic criteria (Pahl and Beitz, 1988). Steps 1) to 5) constitute the synthesis process whereas step 6) is the evaluation process. The process iterates until satisfactory design concepts are found. Clearly, the definition of function and the establishment of function structure are fundamentally important for this conceptual design process model. However, one of the difficult aspects is that the generation of design concepts and the development of function structures are closely coupled. Establishing function structure is especially hard for original design where no product structure exists. This makes conceptual design challenging. Umeda *et al* (1990) proposed to establish a function structure by capturing the design knowledge that transforms design requirements (in some cases, functions) into product descriptions or behaviors. However, definitions of product function and functional knowledge are subjective and domain-dependent.

This chapter proposes an environment decomposition based approach to the generation of design concepts from design requirements, with the uniform representation of design requirements given in Chapter 5. Evaluation process will not be the concern of this thesis. There have already been many results in modeling this process (Suh, 1990; Lee and Thornton, 1996; Law and Antonsson, 1996; Simpson *et al*, 1996).

In fact, a design process model is an algorithmic description of the process. An algorithm, according to traditional understanding, is a finite, unambiguous description of a procedure to solve a class of problems. Fundamentally, it consists of primitive recursive functions and unification operations. Any complex problem can be solved by the unification of primitive recursive functions (Davis and Weyuker, 1983). Similarly, a formal design process model can also include two parts: 1) primitive designs; 2) unification of primitive designs. This process generates design concepts from design requirements represented in Equation (5-7). It is represented in Figure 6-2. The objective of this chapter is to establish a formal model of this process.

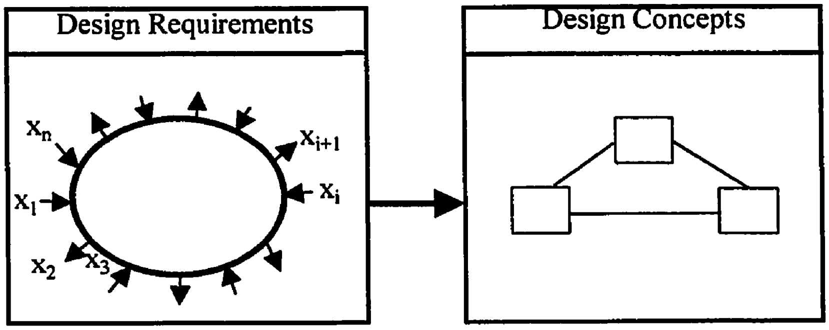


Figure 6-2 Design process

This chapter is organized as follows. In Section 6.2, primitive designs will be discussed, followed by an environment decomposition based design process model, which consists of the decomposition of design requirements and the combination of primitive designs. Some comparisons are made between existing design process models and the current ones before summarizing this chapter.

6.2 Primitive Designs

According to the classification of design requirements in Figure 5-3, two cases need to be taken into account in generating primitive products. One is related to structural requirements while another is to performance requirements. Structural requirements impose direct constraints on the form or dimensions of the product to be designed. Performance requirements lead to the generation of primitive products based on performance knowledge, which was defined in Equations (4-16) through (4-20).

In solving a design problem, there are two ways to make use of Equation (4-16), corresponding to the two aspects addressed in Equations (3-14) and (3-15): 1) the focus is on the pair of action and response without looking into their relations. It only needs to recognize the existence of the relation, which actually defines a performance of the product.

Considering the hierarchical definition of performances in Equation (4-22), multiple levels of product performances can be generalized. Therefore, product function is a special case of this representation. This perspective can then be used for design synthesis purpose. 2) the focus is on the relations between actions and responses so that the performances can be quantified in terms of the current representation of product structure. The result can then be used for evaluation purpose.

All primitive products with their performance knowledge constitute a set of design knowledge for solving a class of design problems. They form the foundation of product design in the domain. Following are some examples from riveting tool design. Performance knowledge is not given. Only input (actions) and output (responses) are shown.

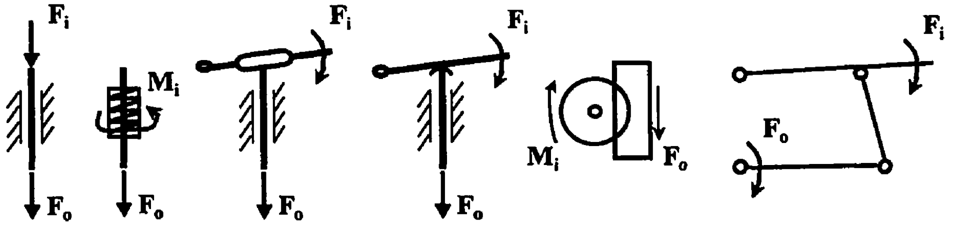


Figure 6-3 Primitive designs for moving heads together

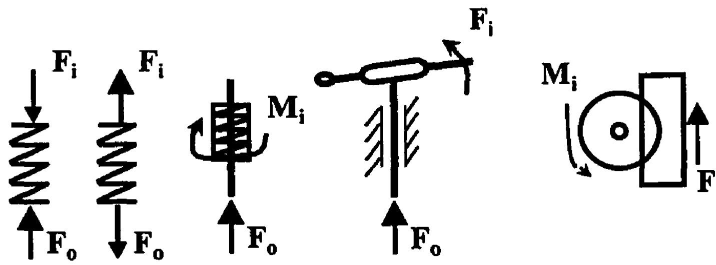


Figure 6-4 Primitive designs for moving heads apart

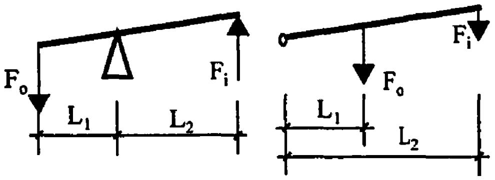


Figure 6-5 Primitive designs for amplifying forces

Table 6-1 Primitive designs for operator-tool interface

Foot			**Hand**				
Pedal	Stirrup	...	lever	wheel	push rod	pull rod	...

$$\mathbf{S^a = \{Rod, Lever, Eccentric\ cam, Wedge, Screw, Spindle, Spring, wheel, pedal,...} \tag{6-1}$$

The following algorithm is proposed to solve the primitive design problem.

Step1: determine a set of candidate primitive products S_a. This is done by matching the given design requirements R^d, which is represented by product-environment boundary $B[n]$, to the actions A_i and responses R_i in performance knowledge set K' as defined in Equation (4-20). Any product $(S^a)_i$ attached to the matched performance knowledge k'_{ij} as defined in Equation (4-16) is taken as a candidate solution. All matched primitive products constitute the set of candidate primitive products S_a.

$$\begin{aligned} \mathbf{S_a} = \{ & \mathbf{(S^a)_i[n+1] \mid \forall R^d \exists B[n], B[n] = A_{ij}[n] \cup r_{ij}[n],} \\ & \mathbf{R^d = B[n],} \\ & \mathbf{\forall (S^a)_i[n+1], \exists k'_{ij}, k'_{ij} \subset A_{ij}[n] \times r_{ij}[n],} \\ & \mathbf{i = 1,2,...n_a} \} \end{aligned} \tag{6-2}$$

At this stage, since **B[n]** comes from design requirement, $(\mathbf{S}^a)_i[\mathbf{n}]$ is not defined yet. According to Equation (3-13), action and response can not be defined. Therefore, $\mathbf{A}_{ij}[\mathbf{n}]$ and $\mathbf{r}_{ij}[\mathbf{n}]$ need to be further specified.

Step 2: determine performances of each candidate product $(\mathbf{S}^a)_i[\mathbf{n+1}]$ related to design requirements by the matched product performance knowledge $\mathbf{k}'_{ij}$. The performances can be represented as **B[n+1]**,

$$\begin{aligned} \mathbf{A}_{ij}[\mathbf{n}+1] &= \mathbf{E}[\mathbf{n}+1] \times (\mathbf{S}^a)_i[\mathbf{n}+1] \\ \mathbf{r}_{ij}[\mathbf{n}+1] &= \mathbf{k}'_{ij}(\mathbf{A}_{ij}[\mathbf{n}+1]) \\ \mathbf{B}[\mathbf{n}+1] &= \mathbf{A}_{ij}[\mathbf{n}+1] \cup \mathbf{r}_{ij}[\mathbf{n}+1] \end{aligned} \tag{6-3}$$

Here, $\mathbf{A}_{ij}[\mathbf{n+1}]$ and $\mathbf{r}_{ij}[\mathbf{n+1}]$ have been refined in terms of the definition of product $(\mathbf{S}^a)_i[\mathbf{n+1}]$ and updated environment **E[n+1]** compared to those in the step 1.

Step 3: evaluate the candidate products against design requirements

$$\mathbf{R}^d = \lambda(\mathbf{B}[\mathbf{n}+1], \mathbf{B}[\mathbf{n}]) \tag{6-4}$$

Step 4: If design requirements are satisfied, then end the process. Otherwise go to step 1.

This four-step process is named after the primitive design process. Mathematically, primitive design can be defined as:

If for a set of given design requirements $\mathbf{R}^d$, there is a map $\mathbf{f}^n$ projecting $\mathbf{R}^d$ into a product-environment boundary set **B**, which can be partitioned into an action set **A** and a response set **R** so that there is at least one product description that has a set of performance knowledge **K′**,

$$K' : A \rightarrow R \tag{6-5}$$

then this design problem is primitive.

Graphically, primitive design can be illustrated in Figure 6-6.

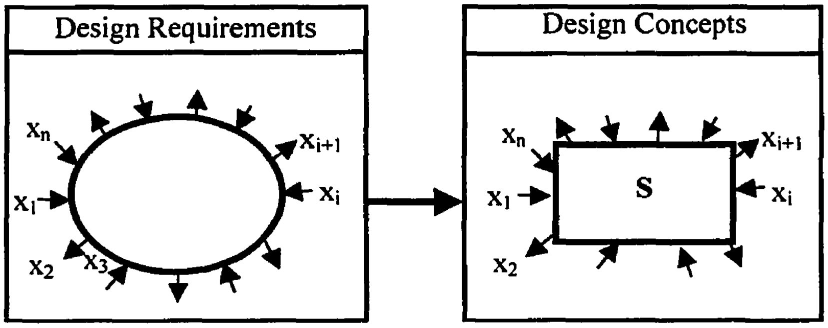


Figure 6-6 Primitive design

In the above process, the choice of primitive design is in fact artificial and relies on designers' expertise and knowledge as well as the state of the art of technology. For example, experienced designers have more primitive designs in mind so that their designs are usually more alive and flexible, compared with naïve designers. They also have more complex primitive designs and in turn more sophisticated input and output, which make their generations of design faster in many cases. Figure 6-7 is such an example for the rivet setting tool design. It can deal with three elements instead of two at one time. In this way, it satisfies two functions: "Move heads close" and "Move heads apart".

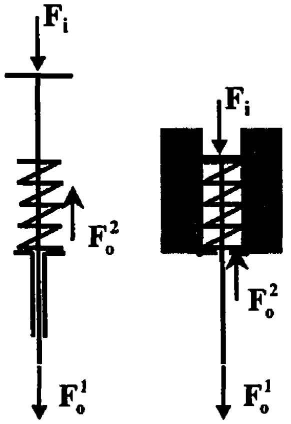


Figure 6-7 More sophisticated primitive design

6.3 Environment Decomposition Based Design Concept Generation

Since design requirements can be mapped into an environment set through Equation (5-10), the design process shown in Figure 6-2 is represented as the following mapping:

$$\mathbf{D} : \mathbf{B} \rightarrow \mathbf{S} \tag{6-6}$$

where **D**: design process;

B: environment defined in Equation (4-27);

S: product descriptions, defined in Equation (4-9).

For the environment **B** in the above equation, if a set of performance knowledge **K'** as in Equation (6-5) can be found, then the design problem is primitive. However, in real life design problems, it is not always easy to find a proper set of design knowledge to match the whole set of actions and responses included in the environment. It is therefore essential to transform a design problem into primitive ones if algorithmic solutions are expected. To do so, the constitution of the environment needs to be analyzed with reference to primitive designs.

As was implied in Equation (4-27), an environment set consists of three parts: action, response, and contacting environment elements. Based on primitive designs, there are three corresponding possibilities to generate tentative product descriptions as follows:

1) Starting from action(s). Search through all available performance knowledge, pick up those that can accept or allow the action(s). The product descriptions attached to the set of matched knowledge are candidate solutions corresponding to the action(s);

2) Starting from response(s). Search through all available performance knowledge, pick up those that generate the response(s). The product descriptions attached to the set of matched knowledge are candidate solutions corresponding to the response(s);

3) Contacting environment elements. They directly limit and constrain the form or dimensions of product, therefore the candidate product descriptions can be derived directly.

As was indicated in Section 3.4.7, the newly generated product components can be taken as a part of the environment for the succeeding design. In each of the above cases, the environment will be updated in the following ways:

1) If the component was generated by matching action(s), then replace the action(s) by the corresponding response defined by the selected primitive designs and replace the contacting environment element when necessary.

2) If the component was generated by matching response(s), then replace the response by the corresponding action(s) defined by the selected primitive designs and replace the contacting environment element when necessary.

3) If the component was generated by matching contacting environment elements, then refine the structural definition of product components as well as environment.

Following this process, the generated partial products need to be evaluated using the selected performance knowledge. Design solutions are those passing all the tests set by design requirements, as is shown in Figure 6-8.

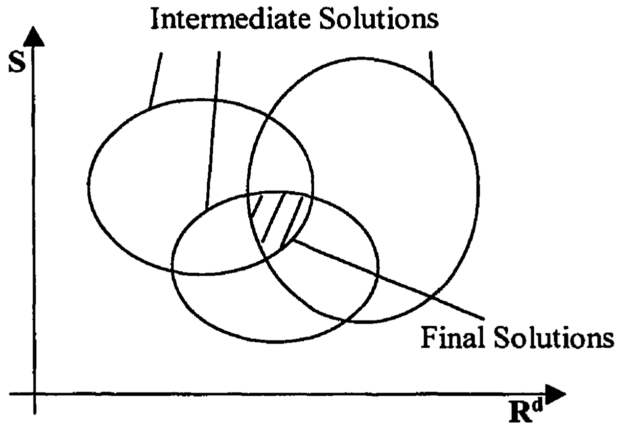


Figure 6-8 Design solution space

In summary, this design process can be described as follow:

1) extract a subset from the current environment set;

2) if there is a piece of design knowledge mapping the extracted environment subset to another environment subset, then the product structure S_i attached to this knowledge could be a component. The extracted environment subset is replaced by the

corresponding environment subset defined by the design knowledge. The current environment is updated;

3) Add component S_i to already existing product S. The product descriptions are updated;

4) Detect the performance conflicts between the newly generated product component and already existing product, update the current environment;

5) If no more environment decomposition can be done, then go to succeeding design stages, else go to step 1.

Figure 6-9 gives the scheme of this model.

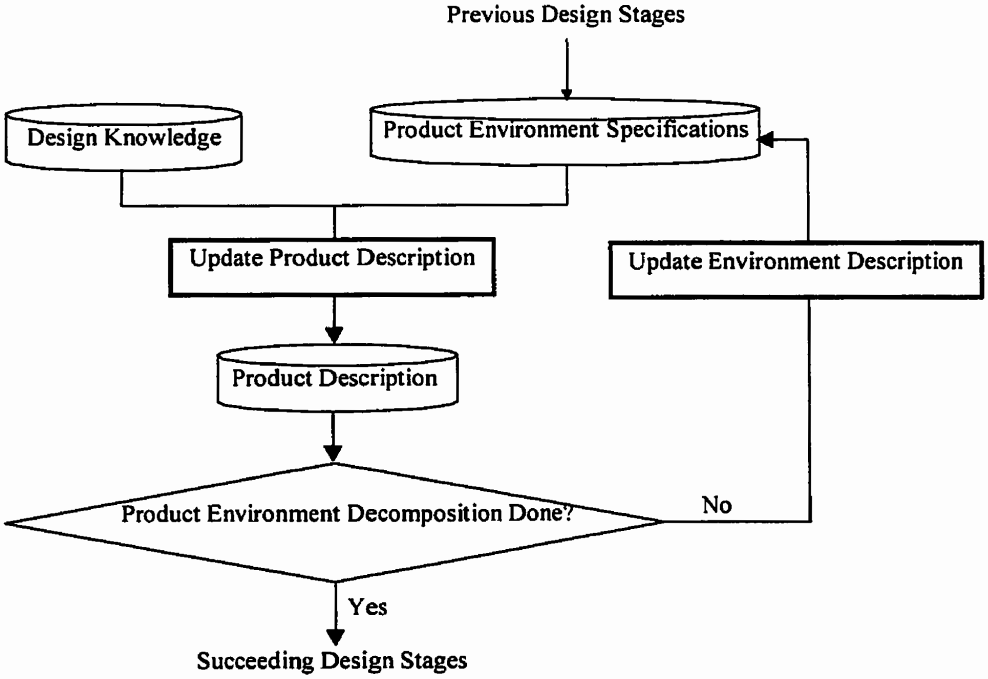


Figure 6-9 Environment decomposition based conceptual design process

Here, the criteria for verifying if the environment decomposition is done is that no single action or response is not connected to the final product performance network.

Mathematically, the process can be formulated as

```
Design($R^d$,S)
  {
   $i = 0;$
   $S[i] = S;$
   $R^d = B[i]$
  repeat
  {
            $\exists X_k \in B[i];$   //decomposition of product environment
            if $\exists S_l\ \exists k_{lm}, k_{lm} : X_k \rightarrow X_n$ or $k_{lm} : X_n \rightarrow X_k$ //apply performance knowledge
            $B'[i] = (B[i]/X_k) \cup X_n;$ //update product environment
            $S[i+1] = S_l \cup S[i];$ //combination of component into partial product
            $\forall S_l, \exists K_l, X_l = K_l(S_l);$
            $\forall S[i], \exists K_p, X_p = K_p(S[i]);$   // properties of component and partial product
            $B''[i] = X_l \updownarrow X_p;$ //conflicts between partial product and component
            $B[i+1] = B'[i] \cup B''[i];$ //update product environment
  until no more environment decomposition can be done
}
```

In the above algorithm, "/" is the difference of sets, "$\updownarrow$" represents the conflicts between sets. The newly generated environment **B[i+1]** consists of two parts. One is the environment given by subtracting the primitive designs from the original environment **B[i]**. Another is the conflict between the generated primitive component and the existing product description. A specific design methodology called TIPS (Altshuller, 1988) focused on this problem. Other research includes Haroud *et al* (1995), Bahler *et al* (1995), Otto and Antonsson(1991), Oh and Sharpe(1995), Brazier, *et al*(1995), Reddy *et al*(1996). No further details will be given here.

This process can be graphically described in Figure 6-10:

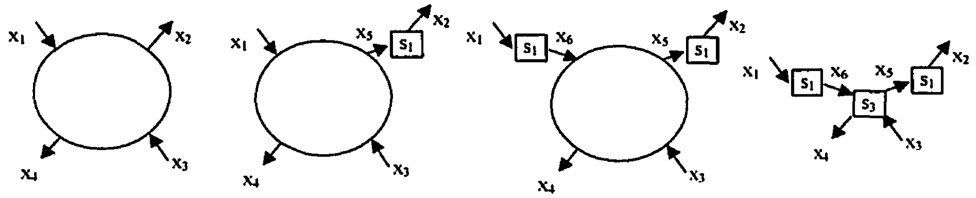


Figure 6-10 Graphic explanation of environment decomposition based conceptual design process

Figure 6-11 gives how a product concept for rivet setting tool is generated incrementally. In this example, in each step, there could have many alternatives. But to save space, we only list one alternative. The environment is gradually decomposed into primitive ones, which allows product description to be generated incrementally. However, a product may have some performances besides those contributed to defining and satisfying design requirements. They are called unintended behavior in some literature (Deng, *et al*, 2000). These "unintended" performances may conflict with those of the already generated product components. This fact makes the problem subject to continuous redefining. This is one of sources of so called ill-structure of design problems (Simon, 1973). Therefore, the environment decomposition process in the above algorithm is recursive, dynamic, and incremental.

Figure 6-11 gives one updated environment and product descriptions are:

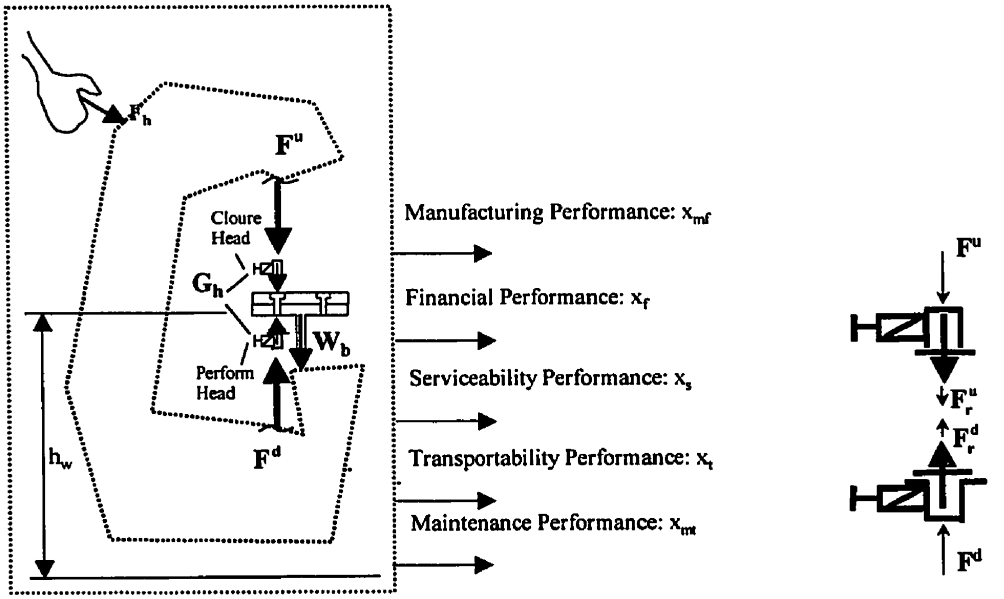
$$\begin{aligned} &B = \{F_h, F^u, F^d, W_b, G_h, x_{mf}, x_f, x_s, x_t, x_{mt}\} \\ &S = \{h_w, G_h\} \end{aligned} \tag{6-7}$$

Figure 6-12 gives another updated environment and product descriptions are:

$$\begin{aligned} &B = \{F_l, F^u, F^d, W_b, G_h, G_l, x_{mf}, x_f, x_s, x_t, x_{mt}\} \\ &S = \{h_w, G_h, G_l\} \end{aligned} \tag{6-8}$$

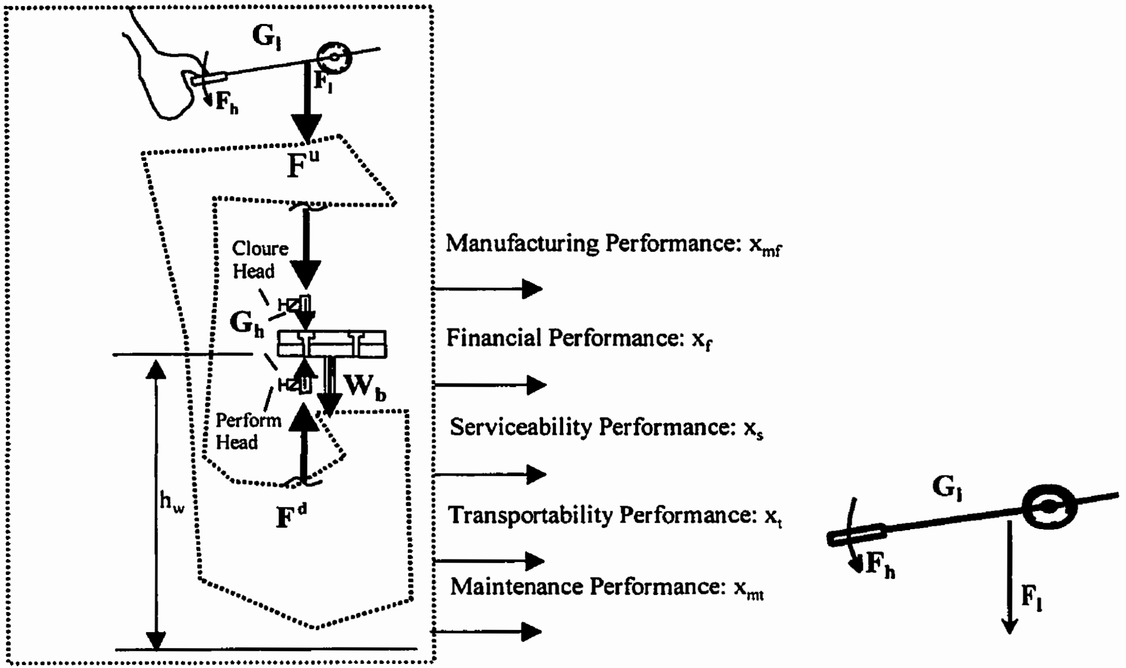
It should be noted that only forces contributing to the function of components are given in Figure 6-11 and Figure 6-12.

This is a formal realization of the design process required by Figure 4-2 and Figure 5-7. The process progresses by refining the abstract and simple ideas of product to concrete and complex descriptions of product until the final solutions are found.



a) Intermediate product descriptions and environment b) Related performance knowledge

Figure 6-11 One intermediate design for the riveting tool design example



a) Intermediate product descriptions and enviornemnt b) Related performance knowledge

Figure 6-12 Another intermediate design for the riveting tool design example

6.4 Relation to Current Research in Design Process

The proposed mathematical representation of primitive design fits well with the existing design process models. They correspond to different levels of complexity and abstraction of product description in the definition of performance knowledge in Equation (6-3). When the product description is well defined, it becomes a case-based design. When the product description is just partially defined, it is a knowledge-based design.

In the above process, if actions, responses, and product descriptions that define the performance knowledge are sophisticated and can be standardized, then this process becomes the modular design (Gu *et al*, 1997). The process could end within very few loops. The extreme case is parametric design in which only local modifications are needed for design.

6.5 Conclusions

Based on the formulation of design objects given in the previous chapters, this chapter presented a new design process model. The core of this model is environment decomposition. With this model, design concepts can be generated step by step. The only knowledge required in this process is performance knowledge, which has been studied in different scientific investigations.

Chapter 7 The Nature of Design Problem: Design Governing Equation

7.1 Introduction

The last three chapters demonstrated that the results, derived from the established axiomatic system in Chapter 3, can be used for solving the design problem. They constitute a formal model of process **D** from design specifications to design solutions. This process is given in Figure 4-1. According to the research scheme given in Figure 1-3, it is crucial to demonstrate the nature of this process and compare it against the real life design practice, which was summarized in Section 2.3. This is the objective of this chapter and chapter 8. In this chapter, the design governing equation will be formulated to explore the dynamic nature of design. This equation implies that the force driving a design to evolve comes from within the design problem, from each state of design progress.

Like any other events that have happened in the evolution of science, a governing equation embodying the basic nature of design would be beneficial for the progress of the science of design. This equation addresses the basic mechanism governing the activities in design by covering the relations among design requirements, environment, product descriptions, product performance, and design knowledge. Chapter 8 will focus on the nature of solutions of the design governing equation. This research could be an important part of scientific exploration of design phenomenon. It can also provide further clues for improving the understanding of design and design processes. However, due to the complex nature of design, it is exploratory and by no means conclusive.

In Section 7.2, the design governing equation will be formulated, followed by the discussion of the nature and characteristics of the equation in Sections 7.3 and 7.4. Section 7.5 will discuss another form of design governing equation and its implication. The related work will be compared in Section 7.6. Section 7.7 gives some remarks on this work.

7.2 Design Governing Equation: Formulation

The core of design processes is given in Figure 6-1. It includes two phases: synthesis and evaluation. They were further formalized in Equations (6-2) and (6-3).

Equation (6-2) can be rewritten in logical form as

$$\begin{array}{l} \mathbf{B[n] = A_{ij}[n] \cup r_{ij}[n]} \\ \mathbf{\exists (S^a)_i[n+1], \exists k'_{ij} : A_{ij}[n] \rightarrow r_{ij}[n]} \\ \hline \mathbf{(S^a)_i[n+1]} \end{array} \tag{7-1}$$

Alternatively, according to Equations (3-15) and (3-16), Equation (7-1) can also be written as

$$\begin{array}{l} \mathbf{B[n]} \\ \mathbf{k_{ij} : (S^a)_i[n+1] \rightarrow B[n]} \\ \hline \mathbf{(S^a)_i[n+1]} \end{array} \tag{7-2}$$

Alternatively,

$$\mathbf{(S^a)_i[n+1] = k_{ij}^{-1}(B[n])} \tag{7-3}$$

Meanwhile, Equation (6-3) can be rewritten in logical form as

$$\begin{array}{l} \forall (\mathbf{S^a})_i[\mathbf{n}], \mathbf{A}_{ij}[\mathbf{n}] = \mathbf{E} \times (\mathbf{S^a})_i[\mathbf{n}] \\ \forall (\mathbf{S^a})_i[\mathbf{n}], \exists \mathbf{k}'_{ij} : \mathbf{A}_{ij}[\mathbf{n}] \rightarrow \mathbf{r}_{ij}[\mathbf{n}] \\ \hline \mathbf{r}_{ij}[\mathbf{n}] \end{array} \tag{7-4}$$

And

$$\mathbf{B}[\mathbf{n}] = \mathbf{A}_{ij}[\mathbf{n}] \cup \mathbf{r}_{ij}[\mathbf{n}] \tag{7-5}$$

Alternatively, according to Equations (3-15) and (3-16), Equation (7-4) can also be written as

$$\begin{array}{l} (\mathbf{S^a})_i[\mathbf{n}] \\ \mathbf{k}_{ij} : (\mathbf{S^a})_i[\mathbf{n}] \rightarrow \mathbf{B}[\mathbf{n}] \\ \hline \mathbf{B}[\mathbf{n}] \end{array} \tag{7-6}$$

Which can be further represented as a mathematical equation as

$$\mathbf{B}[\mathbf{n}] = \mathbf{k}_{ij}((\mathbf{S^a})_i[\mathbf{n}]) \tag{7-7}$$

Substitute Equation (7-7) into Equation (7-3), we get

$$(\mathbf{S^a})_i[\mathbf{n}+1] = \mathbf{k}_{ij}^{-1}(\mathbf{k}_{ij}((\mathbf{S^a})_i[\mathbf{n}]) \tag{7-8}$$

Ideally, if $(\mathbf{S^a})_i[\mathbf{n}]$ is a solution of the above equation, then

$$(\mathbf{S^a})_i[\mathbf{n}] = (\mathbf{S^a})_i[\mathbf{n}+1] \tag{7-9}$$

Hence, Equation (7-8) can be written as

$$(\mathbf{S^a})_i = \mathbf{k}_{ij}^{-1}(\mathbf{k}_{ij}((\mathbf{S^a})_i) \tag{7-10}$$

As was indicated in Section 6.2 (Figure 6-7 is an example), the definition of primitive product is relative. It is logical to replace $(\mathbf{S^a})_i$ by S, therefore a more general form of the above equation is obtained

$$\mathbf{S} = \mathbf{k}_j^{-1}(\mathbf{k}_j(\mathbf{S})) \tag{7-11}$$

This equation conforms to the form given in Equation (3-30). It is called the ***Design Governing Equation***. It is a recursive equation and is the mathematical form of the logic of design proposed by Zeng and Cheng(1991) and Roozenburg(1992). It governs design activities and underlies design processes. Defining *design function* **D** as

$$\mathbf{D} = \mathbf{k}_j^{-1} \bullet \mathbf{k}_j \tag{7-12}$$

Then the design governing equation becomes

$$\mathbf{S} = \mathbf{D}(\mathbf{S}) \tag{7-13}$$

It should be noted that here the design function is a concept different from the functions used to define design requirements, as was discussed in Chapter 5. The design governing equation makes design problem solving as a search for fixed points (Shashkin, 1991) under the design function **D**.

7.3 Nature of Design Function

It was indicated Section 3.4.2 that for a given product, a product performance is solely determined under certain circumstances. According to Section 3.4.2, there are two ways to represent performance knowledge, one is Equation (3-15). Another is Equation (3-16). Both are isomorphic. This also supported the transformation from Equations (7-1) and (7-4) to Equations (7-2) and (7-6). The performance knowledge could be disclosed scientific principles, designers' expertise, and so on. The relation of product performance to product description is illustrated in Figure 7-1. It means that a product property can only assume one value for defined product description and environment.

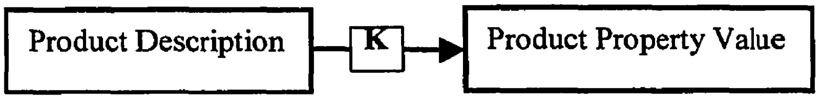


Figure 7-1 Relation between the value of a product property and product description

On the contrary, however, the inverse mapping of performance knowledge is plausible in most cases because it is the reverse of causal law. Therefore, the product description is not merely determined. For one product property, there may exist multiple product descriptions relating to it, as is shown in Figure 7-2. This is a divergent process. The more product descriptions are generated, the more chances the final design is optimal, and in turn, the more resources the design processes will consume. This fact is also implied in the axiom of bounded rationality. If human designers have perfect knowledge of synthesis, then the evaluation process can be removed from design process. This is the case in Yoshikawa's general design theory(1981).

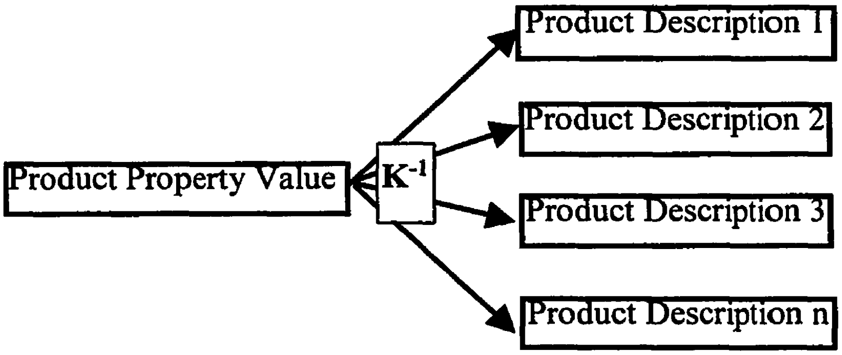


Figure 7-2 Mapping from product property to product descriptions

These two processes can be seen as two operators: synthesis and evaluation operators, acting on the solution space of design problems. The synthesis operator tries to stretch the solution space whereas the evaluation operator attempts to shrink the space. The interaction of both operators gives rise to the final balanced design solutions as shown in Figure 7-3.

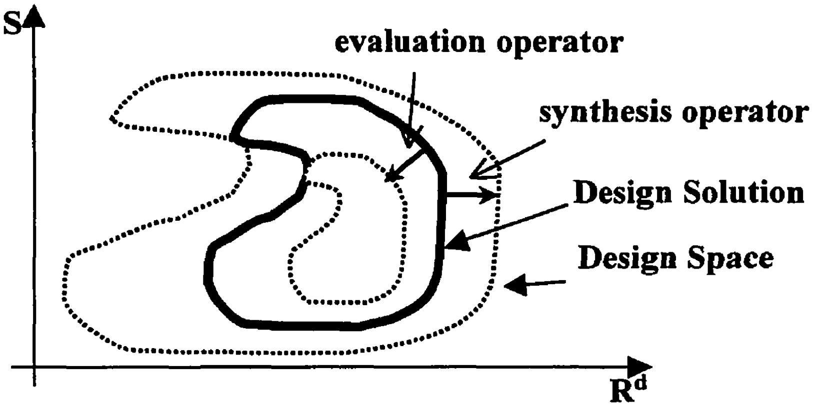


Figure 7-3 Design space under synthesis and evaluation

7.4 Characteristics of Design Governing Equation

As Equation (7-13) suggested, design governing equation makes design problem solving a search for fixed points under the design function **D**. Prior to the discussion of the characteristics of equation, the fixed point theory will be briefly reviewed.

7.4.1 Fixed point theory

Fixed point theory is a branch of topology. It plays an important role in looking for roots of an equation (Shashkin, 1991). Basically, the problem can be posed as:

Let f be a function of real variable x, continuous in closed interval [a,b] and mapping point x from interval [a,b] into point $y=f(x)$ of the same interval [a,b]. If a point x_0 is a solution of the following equation (Equation (7-14)), then it stays where it was. This point is a fixed point (Figure 7-4).

$$f(x) = x \tag{7-14}$$

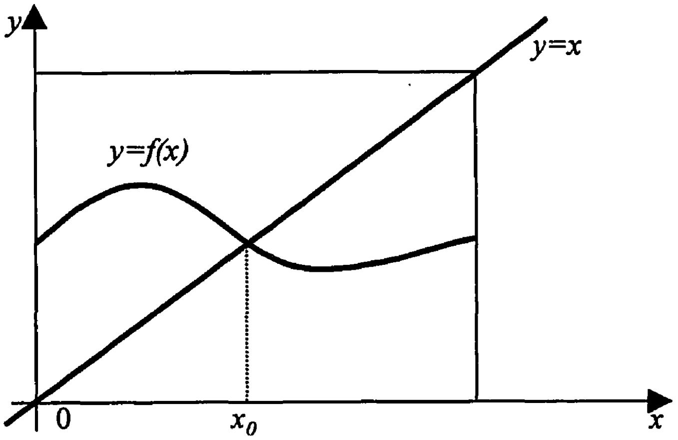


Figure 7-4 Fixed point

The solving of Equation (7-14) is usually accomplished by an iterative method. For the "initial approximation", take an arbitrary number, x_1, inside the closed interval of the mapping and substitute it into the right hand side of the equation. Expression $x_2=f(x_1)$ is then taken for the second approximation. In general, for each successive approximation, x_n, the next one, x_{n+1}, is found by the formula

$$x_{n+1} = f(x_n) \tag{7-15}$$

If a sequence of numbers $\{x_n\}$ has a limit x_0, then x_0 is a root of Equation (7-11). This process is illustrated in Figure 7-5.

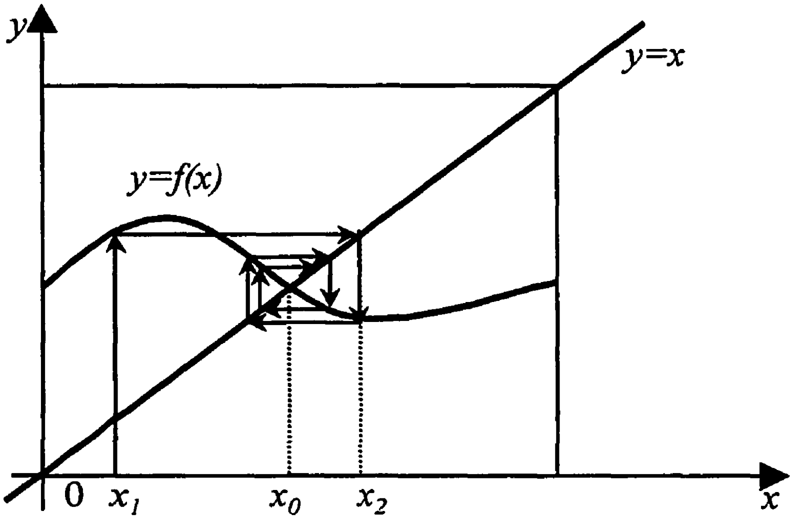


Figure 7-5 Finding fixed point

7.4.2 Structure of design function

In this section, we will discuss the constitution of design function **D** presented in Equation (7-12). As was implied in Equation (6-2), the mapping $\mathbf{k}'_{ij}$ and in turn $\mathbf{k}_{ij}$ is dependent on the product description $(\mathbf{S}^a)_i[\mathbf{n}]$ and the performance requirement $\mathbf{B}[\mathbf{n}]$, which is the union of $\mathbf{A}_{ij}[\mathbf{n}]$ and $\mathbf{r}_{ij}[\mathbf{n}]$. Furthermore, it is a subset of all performance knowledge **K** in its working environment. Mathematically, $\mathbf{k}_{ij}$ can be represented as a correlation class of all performance knowledge **K** with respect to product descriptions $(\mathbf{S}^a)_i$ and the design requirements $\mathbf{B}[\mathbf{n}]$.

$$\mathbf{k}_{ij} = [(\mathbf{S}^a)_i[\mathbf{n}] \cup \mathbf{B}[\mathbf{n}], \mathbf{K}]_{\approx} \tag{7-16}$$

Correlation class is defined based on correlation relation "≈" between two sets **Y** and **Z**

$$\forall \mathbf{y} \in \mathbf{Y}\, \forall \mathbf{z} \in \mathbf{Z}\ (\mathbf{y} \approx \mathbf{z} \rightarrow \mathbf{z} \approx \mathbf{y}) \tag{7-17}$$

The correlation class of set **Z** with respect to set **Y**, denoted as $[\mathbf{Z},\mathbf{Y}]_{\approx}$, is a subset of **Y** consisting of all those elements in **Y** that has correlation relation with each element in **Z**.

$$\mathbf{[Z,Y]_{\approx} = \{y \in Y \mid \forall z \in Z, y \approx z\}} \tag{7-18}$$

For example, assume

$$\mathbf{B[n] = \{Input\_force, Output\_force\}} \tag{7-19}$$

If $\mathbf{(S^a)_I[n]}$ is the spring in Figure 6-4,

$$\mathbf{(S^a)_i[n]} = \mathit{spring} \tag{7-20}$$

Then $\mathbf{k_{ij}}$ is defined as

$$\begin{aligned} &\mathbf{k'_{ij} : F_i \rightarrow F_o} \\ &\mathbf{F_o = F_i} \end{aligned} \tag{7-21}$$

It can also be denoted as

$$\mathbf{k'_{ij} = [spring \bigcup F_i \cup F_o, K]_{\approx}} \tag{7-22}$$

If, however, $\mathbf{(S^a)_I[n]}$ is the lever in Figure 5-11,

$$\mathbf{(S^a)_i} = \mathit{lever} \tag{7-23}$$

Then $\mathbf{k'_{ij}}$ is defined as

$$\begin{aligned} &\mathbf{k'_{ij} : F_1 \rightarrow F_2} \\ &\mathbf{F_2 = \frac{L_2}{L_1} F_1} \end{aligned} \tag{7-24}$$

If no $\mathbf{k'_{ij}}$ is found, then the related performance knowledge of a product does not exist and should be acquired to solve the problem. This is why modeling, analysis, experiments and prototyping are often required in accomplishing a design task.

Substituting Equation (7-16) into Equation (7-12), we get

$$\mathbf{D} = [(\mathbf{S}^{a})_{i}[\mathbf{n}] \cup \mathbf{B}[\mathbf{n}], \mathbf{K}]_{\approx}^{-1} \bullet [(\mathbf{S}^{a})_{i}[\mathbf{n}] \cup \mathbf{B}[\mathbf{n}], \mathbf{K}]_{\approx} \tag{7-25}$$

Obviously, there are two cases in the process of design problem solving: 1) the form of **D** keeps changing upon each newly generated product descriptions; 2) the generated design function, **D**, has a fixed form in the process. The first case corresponds to what happens in early design stages like conceptual design where the design process jumps among different design concepts. The second one occurs to the later design stages like detailed design where the form of product does not change and only parametric adjustment happens.

In the first case, each product description defines a domain for succeeding design process to search for solutions. In the progress of design process, new type of product descriptions always correlates to different forms of performance knowledge. The design process is to look for fixed points over different design concepts. Once a design concept is decided, the remaining design tasks will keep on using the same performance knowledge to adjust product descriptions **S** so that an optimal design solution can be defined.

The process is shown in Figure 7-6. In the figure, those candidates that do not satisfy design requirements, such as f, do not have interactions with the line *y*=**S**.

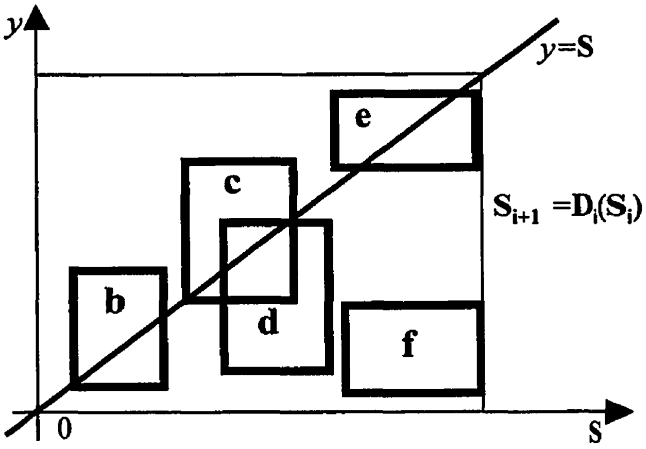


Figure 7-6 Early stages of design

An example is designing a structure to bear loading so that the structural deformation is within a limit. This is a structural topology and shape design problem. The domain in this case is entire structure types, including bar, beam, column, plate, shell, and their mixtures. If, for instance, beam is chosen as a candidate, then a new domain for design solution will be created. It can be different types of beams including simply supported beam, fixed-ends beam, cantilever beam, and so forth. The process continues until no more domain is created. The domains are the fixed points of the design function.

The above process can be represented as

$$\mathbf{S}_{i+1} = \mathbf{D}_i(\mathbf{S}_i) \tag{7-26}$$

where $\mathbf{D}_i$ is the design function depending on the type of the product descriptions $\mathbf{S}_i$.

However, in the later design stages, since the concepts have been defined, the form of performance knowledge will not change in the design process. Thus the design process is looking for fixed points under the well defined design function in the domain defined by the chosen design concept. The iteration process can be represented as

$$\mathbf{S}_{i+1} = \mathbf{D}(\mathbf{S}_i) \tag{7-27}$$

For the example of structural design, if the final domain is fixed-end beam, then the design becomes a parametric design problem subjected to given constraints. The process is shown in Figure 7-7.

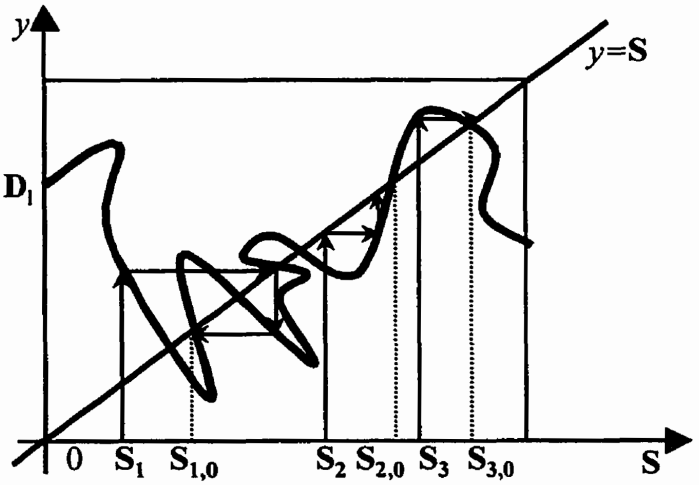


Figure 7-7 Parametric design

From the above discussion, it can be seen that in the early stages of design, the design function **D** generates product description **S** and the product description **S** in turn redefines the design function **D**. They interact with each other. This fact makes the process nonlinear and design governing equation a nonlinear dynamic equation. Chapter 8 will discuss this nonlinearity and its implication in design creativity.

7.5 Discussion

An alternative way to that leading to Equation (7-8) is substituting Equation (7-3) into Equation (7-7). Then we have

$$\mathbf{B}[\mathbf{n}] = \mathbf{k}_{ij}(\mathbf{k}_{ij}^{-1}(\mathbf{B}[\mathbf{n}-1])) \qquad (7\text{-}28)$$

This equation means that design process also leads to fixed points of environment **B[n]** which includes design specifications and product performances. By combining Equations (7-8) and (7-28), a more general form of the design governing equation is obtained,

$$\begin{Bmatrix} \mathbf{S}[\mathbf{n}+1] \\ \mathbf{B}[\mathbf{n}+1] \end{Bmatrix} = \begin{bmatrix} \mathbf{k}_{ij}^{-1}(\mathbf{k}_{ij}) & \mathbf{0} \\ \mathbf{0} & \mathbf{k}_{ij}(\mathbf{k}_{ij}^{-1}) \end{bmatrix} \begin{Bmatrix} \mathbf{S}[\mathbf{n}] \\ \mathbf{B}[\mathbf{n}] \end{Bmatrix} \qquad (7\text{-}29)$$

Graphically, it can be described as in Figure 7-8.

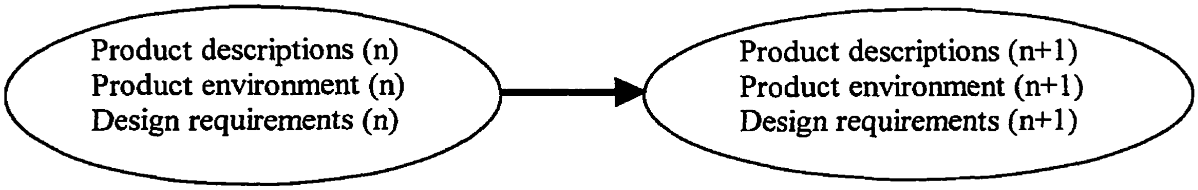


Figure 7-8 Mapping in design process

Based on this observation, design can be further defined as:

> Design is a transition process from one state of design to another. Each state is composed of current product environment, current design requirements, current product descriptions, and the product performances derived from the product descriptions.

The design state space is composed of the attribute space and function space which are addressed by Yoshikawa and Tomiyama(1994), and Braha and Maimon(1998).

More accurately, the process in Figure 7-8 can be represented in Figure 7-9, where each state might intrigue several candidate states. At each stage of state transition, a new design problem is being solved.

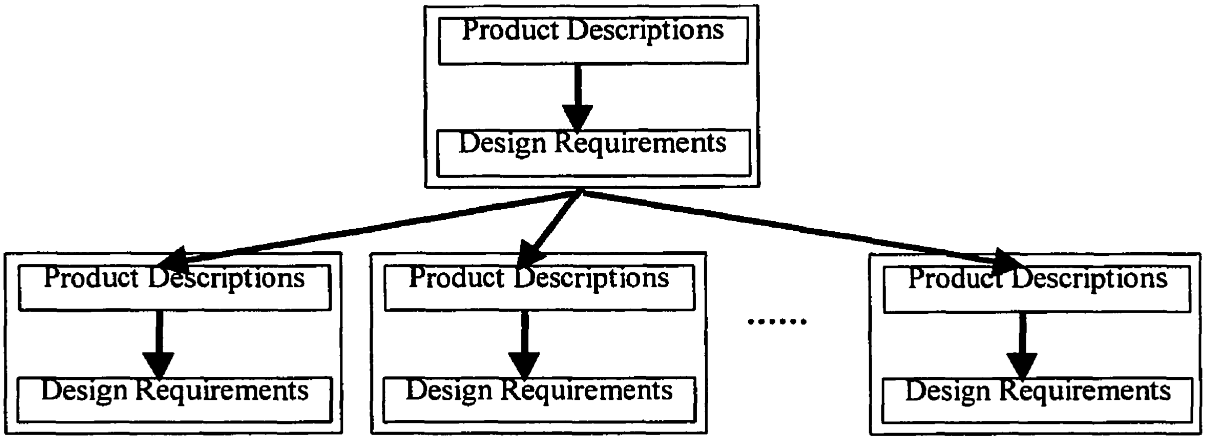


Figure 7-9 Refined mapping in design process

7.6 Related Work

Several other authors also tried to establish the similar equation. Fauvel(1991) proposed a representative design process as follows:

(Activity(n), Embodiment(n))→Activity(n+1)

An **Activity(i)** is some component process of the overall design process and an **Embodiment(i)** is the physical manifestation of the result of the completion of an activity. It indicated that given an initial design activity, the results of that activity drive the selection and execution of the other activities.

Salutri and Venter(1992) proposed a functional model of design process as

$S_{i+1}=d(S_i+P)$

In this model, for each iteration i+1, the design process **d** is applied to the problem plus the solution at iteration i.

The same idea was also discussed in the logic of design (Zeng and Cheng, 1991; Roozenburg, 1992).

7.7 Conclusion

This chapter discussed the general design governing equation and its nature and characteristics. It is argued that design problem solving is looking for the fixed points of the design function defined by the governing equation. Corresponding to different design stages, the general design function has different meanings. In the conceptual design stage, the form of general design function evolves as design process advances. In the configuration and detailed design stages, the general design function is the created solution principles.

Based on the design governing equation, design process is defined as a transition process from one state of design to another. Each state is composed of current environment, current design requirements, current product descriptions, and the product performances derived from the product descriptions.

Chapter 8 The Nature of Design Process: Design Creativity

8.1 Introduction

In discussing conceptual design, the design creativity is a problem that can not be avoided. It is important to verify if a design theory can naturally embody this nature of design. This chapter aims to compare the axiomatic theory to this design property and explore the position of creativity in the design process model established in Chapter 6. It will be seen that the design process model does not mechanize design creativity. Instead, it indicates how creativity plays a role in the design process.

Conceptual design process is not a process modifying some parameters of an existing design, which can be done by turning to numerical methods. Instead, it generates ideas and concepts just from general and abstract specifications about what the product should function and so on. This has been taken as an uncertain process full of randomness, style, and creativity.

In the Encyclopedia Britannica, creativity is defined as 'the ability to make or otherwise bring into existence something new, whether a new solution to a problem, a new method or device, or a new artistic object or form.' In the study of design cognition, what we are interested in are the basic natures of design creativity and the elements which lead to design creativity. In explicating the conditions and mechanisms that give rise to creativity, Oxman (1990) attributes the design creativity to 'the classification of prior solutions in memory as abstract and generalized knowledge stored in a structure of abstraction level'. She argued that the high level abstraction of knowledge can contribute to the creative application of prior experience in design. In accomplishing a design task, through the process of typification according to some structured representation of precedents the goals and constraints of the design are redefined. A novel reformulation of goals and constraints may make a creative design connection with precedents. Thereupon, she established a dynamic model of design. Furthermore, she discussed two mechanisms: refinement and adaptation, and the organization approach of precedent knowledge to facilitate the task (Oxman and Oxman, 1992; Oxman, 1994). Akin (1990) also explored the importance of expertise in design creativity, and he listed three elements the expertise contributes to creativity: recognition skill, problem restructuring, and procedural knowledge. Based on these and many other studies (such as Amarel, 1966), design creativity can be characterized as:

Conditions: creativity arises under special conditions;

Product: the product of a creative act is novel and unusual in some sense;

Act: a creative act appears to be random and motivated by inspiration.

In the observation, creativity is assumed to be measured through a product. The product should be new and novel compared with what have existed. There are two points to be noted for the comparison here. First, if the comparison is made between the product and the artifacts already existing in the environment, then we may be able to conclude that a novel product, of social value, may have been obtained. Second, if one can create a product which did not exist to her/his knowledge then s/he can be seen as being creative. The focus in this chapter is on the way of creative thinking. Therefore, the latter case will be used as the criteria to evaluate or define creativity in design.

People have been arguing that there is a fundamental flaw in the scientific exploration of designing: if design can be studied scientifically, then randomness will disappear and in turn there will be no room for design creativity. The problem can be more exactly stated as: why do human beings create new things merely with the existing deterministic knowledge at their hands. Or in Oxman's words (1990): "The knowledge of the precedent is, by definition, of the past. How can it be used not only to explain, but to generate, the new?"

This chapter, however, will attempt to show how the randomness and creativity in design are implied in the design process model given in this thesis. This explanation would be based on two basic observations: 1) nonlinear science has demonstrated that chaos and randomness exist for a system governed by deterministic law. This was stated by Crutchfield *et al* more directly and clearly in one of their papers (1986): "Innate creativity may have an underlying chaotic process that selectively amplifies small fluctuations and models them into macroscopic coherent mental states that are experienced as thoughts. In some cases the thoughts may be decisions, or what are perceived to be the exercise of will. In this light, chaos provide a mechanism that allow for free will within a world governed by deterministic laws." 2) design governing equation embodies such a nonlinear mechanism and design has the same underlying mechanism with chaotic dynamics. However, this does not mean that design creativity is mechanized. Instead, it merely indicates how human creativity plays a role in the design process.

In the following section, the basic principle of dynamic system theory will be introduced, followed by the study of the design process as a nonlinear dynamic process. Starting from this argument, three potential routes to creative designs are investigated with the riveting tool design as an example. In Section 8.5concluding remarks are made.

8.2 Basic Concepts of Dynamic Systems Theory

Classical sciences were built up on the basis of deterministic law, from which the Laplace's statement (Crutchfield, 1986) was deduced.

> "The present state of the system of nature is evidently a consequence of what it was in the preceding moment, and if we conceive of an intelligence, which at a given instant comprehends all the relations of the entities of this universe, it could state the respective positions, motions, and general affects of all these entities at any time in the past or future."

The literal application of Laplace's dictum to human behavior led to the philosophical conclusion that human behavior was completely predetermined: free will did not exist, no speaking of creativity. Then a question may be posed as what is the origin of random behavior and further human creativity.

At the turn of the century the French mathematician Poincaré (1929) argued that certain mechanical systems whose time evolution is governed by deterministic law may display chaotic motion. Chaotic dynamics provides a major reason for the randomness of natural behavior.

> "A very small cause which escapes our notice determines a considerable effect that we cannot fail to see, and then we say that the effect is due to chance. If we knew exactly

> the laws of nature and the situation of the universe at the initial moment, we could predict exactly the situation of that same universe at a succeeding moment. But even if it were the case that the natural laws had no longer any secret for us, we could still only know the initial situation approximately. If that enabled us to predict the succeeding situation with the same approximation, that is all we require, and we should say that the phenomenon had been predicted, that it is governed by laws. But it is not always so; it may happen that small difference in the initial conditions produce very great ones in the final phenomena. A small error in the former will produce an enormous error in the latter. Prediction becomes impossible, and we have the fortuitous phenomenon."

By now, it is widely recognized that this phenomenon is abundant in nature and has far reaching consequences in many branches of science. These observations were embodied in the two Axioms given in Section 3.3.

One can understand chaos from the theory of dynamical systems. A dynamical system consists of two parts: a state and a dynamic. The state is the essential information about the system, the components of which are the coordinates of an abstract construct of a state space. The change of state with time is described as the orbit in the state space. The evolution of a system can be visualized in the state space by representing the behavior of the system in geometric form. In general the coordinates of the state space vary with the context; for a mechanical system they might be position and velocity, but for design they are design specifications, environment, product descriptions, and product performances, which have been explained in previous chapters. The dynamic is a rule that describes how the state evolves with time. The temporal evolution may happen in either continuous time or in discrete time. The former is called a flow, the latter a mapping. Apparently, the dynamic in design is a mapping defined by the design governing equation.

Any system that comes into stable motion with the passage of time can be characterized by an attractor in state space, in which rest is an extreme case. Roughly speaking, an attractor is what the behavior of a system settles down to, or is attracted to. A system may have several attractors, different initial conditions may evolve to different attractors. Chaos is an attractor which corresponds to unpredictable motions and have a complicated geometric form. The most fundamental characteristic of chaos is its sensitive dependence on the initial conditions. A small fluctuation can be amplified in its time evolution, in which the qualitative change occurs to the considered system. The fact can be clearly explained by considering the mapping g

$$x_{n+1} = g(x_n) \tag{8-1}$$

which leads to chaotic motion. The initial difference ε between states is amplified to the separation $\varepsilon e^{n\lambda(x_0)}$, as shown in Figure 8-1.

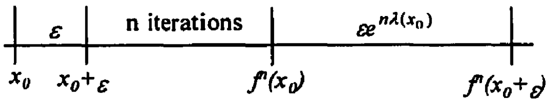


Figure 8-1 Chaotic process

However, not all nonlinear dynamical equations will bring about the chaotic motion. An important element in chaotic dynamics is the existence of a simple stretching and folding operation in the state space. The stretching operation makes an orbit in state space diverge exponentially whereas the folding operation makes the orbit pass close to one another. The orbits on a chaotic attractor are shuffled by this process. The randomness of the chaotic orbits is the result of the shuffling process.

8.3 Dynamics of Design System

It was indicated in Section 7.2 that the design governing equation defines a nonlinear dynamic process. Solving of this equation is essentially looking for fixed points under the design function **D** defined by Equation (7-8).

Since design function is nonlinear, the design governing equation, Equation (7-8), is a nonlinear dynamical equation representing the dynamic mechanism of evolving process of design. It makes designing a nonlinear dynamic system. The state of evolving design process is the union of design requirements, product environment, product descriptions, and product performances, as is shown in Figure 8-2.

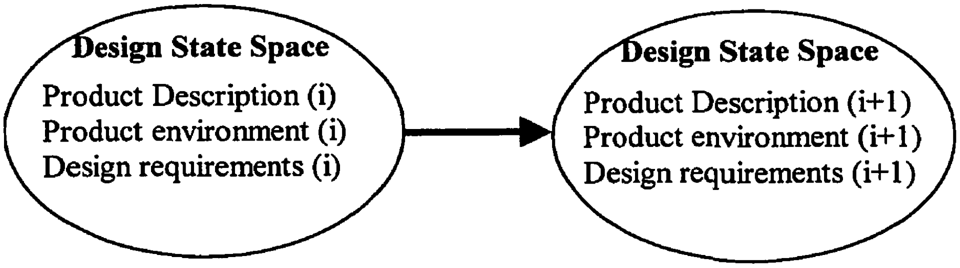


Figure 8-2 Design evolution process

It is obvious from Figure 7-3 that in the design governing equation the synthesis operator acts like a stretching operation which stretches a solution of design while the evaluation operator plays the role of folding operation which folds a solution of design. The shuffling process made up of these two operators will bring the designing into a stable state which is the final product description. As a result, a chaotic dynamic process may be implied in the design processes, depending on the property of the synthesis operator. The final attractors, that are design solutions, depend on the specification of the initial condition. The creativity of design depends on the extent of divergence that a synthesis operator will bring to the process. "Chaos" will occur only if the synthesis operator stretches the state space "exponentially". This means that if a proposed form is far from existing ones, creative designs may be obtained.

In general, every design problem may have many and even infinite number of product descriptions which satisfy the design governing equation. The final "attractors" depend on the initial conditions of the design problem. Basically, the initial conditions for a design include the initial definition of product descriptions, design requirements, and environment. But since design is an ill-defined problem, each step of the problem solving will redefine the problem and make the problem solving process start from a set of modified initial conditions. The way that the "initial conditions" change depends on the choice of every intermediate design solution, which is completely each designer's free will. This makes a design process fluct very easily and the fluctuation might be eventually amplified like what is shown in Figure 8-1. This may explain why design solutions could be so different from designer to designer. It is also in agreement with the present understanding of design creativity (Eder, 1995).

8.4 Routes to Creative Design

In the environment decomposition based design process model, as the initial conditions for a design problem are included in the corresponding environment set, the ways to change the environment are essential to get different design proposals. It can be seen from the formal design process model **Design(R^d,S)** in Section 6.3 that there are three possible ways to update the environment set: 1) extending primitive knowledge set; 2) changing the sequence of environment decomposition; and 3) altering the way of conflict resolution. They provide potentials for creative designs by diverging to different design attractors. However, these three ways largely depend on human intuition, though limited automation can be reached to implement them. We speculate that this might be where some of human design creativity would display in the design process.

8.4.1 Extending primitive design knowledge set

If the primitive knowledge set is extended, then according to the design process **Design(R^d,S)** formulated in Section 6.3, some new and different primitive product $\mathbf{S_l}$ may be generated for a picked environment subset, $\mathbf{X_k}$. As a result, updated environment **E**" will be different, since $\mathbf{X_l}$ could be different for different products. The design may go into a branch which is different from the original one, and subsequently, new product structure may be generated. Naturally in this way different novel design proposals may be obtained. Therefore, it can be assumed that the definition as well as the amount of primitive design knowledge embedded in designer's mind may directly control the quality and efficiency of the final design through redefining the design problem in the dynamic design process.

It can be seen from Equation (4-16) that there are three basic ways to extend the primitive knowledge set.

1) extend primitive product set $\mathbf{S^a}$.

Denote the primitive product set extension operator as $\mathbf{O}_s^e$, then

$$\mathbf{O}_s^e : \mathbf{S}^a \rightarrow (\mathbf{S}^a)^e \tag{8-2}$$

where $(\mathbf{S}^a)^e$ is the extended primitive product set.

For the rivet setting tool design example, if the hammer is added into the primitive product set in Equation (6-1), then a different design proposal could be generated as is shown in Figure 8-3.

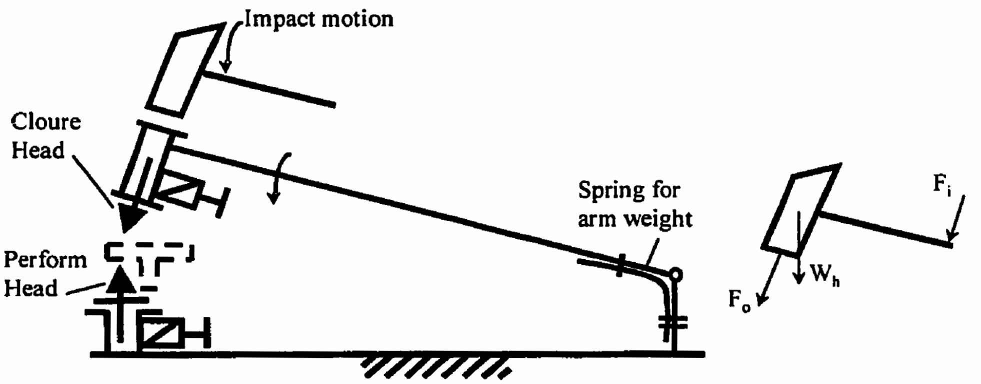


Figure 8-3 A design concept based on the primitive product: hammer

2) extend the environment set (actions and responses) related to a primitive product.

Denote the environment set extension operator as $\mathbf{O}_e^e$, then

$$\mathbf{O}_e^e : \mathbf{B}_{ij} \rightarrow (\mathbf{B}_{ij})^e \tag{8-3}$$
$$\mathbf{B}_{ij} = \mathbf{A}_{ij} \cup \mathbf{r}_{ij}$$

where $\mathbf{B}_{ij}$: environment;

$\mathbf{A}_{ij}$: a set of actions;

$\mathbf{r}_{ij}$: a response to the environment from the product, which is a subset of product properties **X** as was defined in Equation (3-18);

$(\mathbf{B}_{ij})^e$: extended environment.

For the rod in Figure 6-3, its original function is to transform forces. However, if gravity field and its weight are put into its environment set, then a piece of new performance knowledge about the rod is defined, based on which it might be used to prevent a paper from being blown away by the wind. This is shown in Figure 8-4.

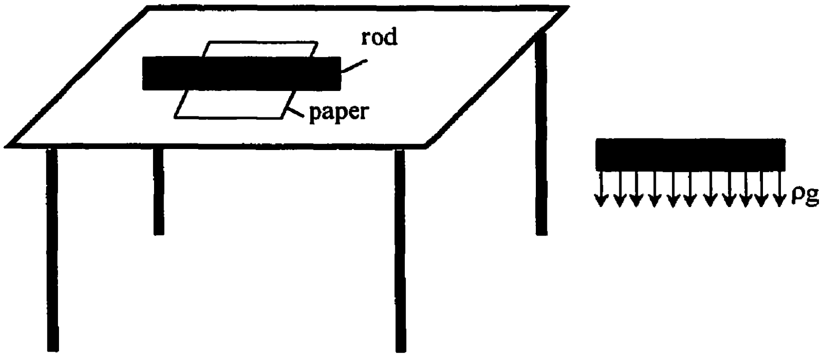


Figure 8-4 Extended performance knowledge of rod

3) extend the relations between the existing primitive products and environment set.

Denote the primitive design knowledge set extension operator as $\mathbf{O}_k^e$, then

$$\mathbf{O}_k^e : \mathbf{K}_i \to (\mathbf{K}_i)^e \tag{8-4}$$

where $\mathbf{K}_i$ is a set of performance knowledge associated with a primitive product. It was defined in Equation (4-19).

Figure 8-5 Beam of composite materials

For the beam with composite materials in Figure 8-5, it is known that it has a performance which is a relation between external loading **F** and displacement δ as was suggested by Equation (3-14):

$$\mathbf{p} = < \mathbf{F}, \boldsymbol{\delta} > \qquad (8\text{-}5)$$

However, the relation required by Equation (3-15) might not be known due to the complexity of materials. But theoretical and experimental research may help establish this relation as

$$\boldsymbol{\delta} = \mathbf{k}(\mathbf{F}) \qquad (8\text{-}6)$$

In this way, the primitive knowledge set is expanded.

8.4.2 Changing the sequence of environment decomposition

Different sequences of environment decomposition will select different primitive knowledge subsets which result in different partial designs and in turn give different redefinitions of environment for the succeeding design problem. As such, the final design solution may be quite different. This serves the same purpose with the restructuring of the problem in Akin(1990) and coincides with an often recommended technique that changing the perspective of seeing a problem may lead to an unusual solution which might be very difficult to obtain.

At each stage of design process, the power set of environment set **B[n]** can be denoted as

$$\mathbf{P(B[n])} = \{\mathbf{x} \mid \mathbf{x} \subseteq \mathbf{B[n]}\} \tag{8-7}$$

In solving the design problem, any element belonging to the power set **P(B[n])** can be picked up to find a match in **K′** defined in Equation (4-19). In the case that several matches are found, different choices decompose the environment in different ways subsequently. This might redefine the design problem differently. For the rivet setting tool design example, the following set is a part of the power set of set **B** defined in Equation (5-10):

$$\begin{aligned}\mathbf{P(B)} = \{&\{\mathbf{F_f}\},\{\mathbf{F_h}\},\{\mathbf{F_r^u}\},\{\mathbf{F_r^d}\},\{\mathbf{W_b}\},\{\mathbf{G_b}\},\{\mathbf{F_f},\mathbf{F_r^u},\mathbf{F_r^d}\},\{\mathbf{F_h},\mathbf{F_r^u},\mathbf{F_r^d}\},\\ &\{\mathbf{h_w}\},\{\mathbf{x_{mf}},\mathbf{x_f},\mathbf{x_s},\mathbf{x_t},\mathbf{x_{mt}}\},\{\mathbf{F_r^u},\mathbf{F_r^d}\}\cdots\}\end{aligned} \tag{8-8}$$

Obviously, there are many ways to start the succeeding problem solving. Generally speaking, it can start from any element of power set **P(B)**. Some elements make sense and others do not. If $\{\mathbf{F_r^u},\mathbf{F_r^d}\}$ is extracted from the environment set, the design can proceed as that in Figure 6-12. The environment and product description are updated in Equation (6-7). If $\{\mathbf{W_b}\}$ is extracted instead, then the design might look like that in Figure 8-6.

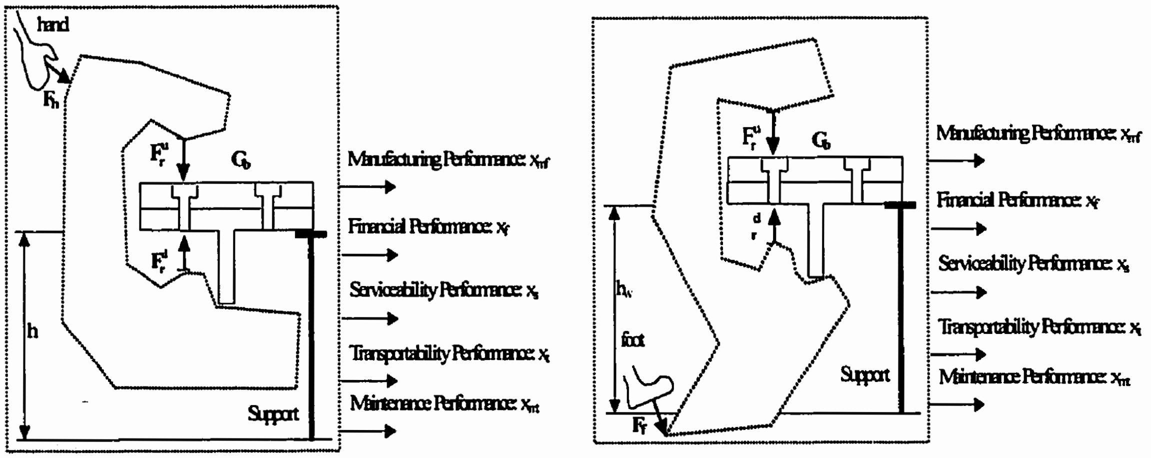


Figure 8-6 Design concept from extracting one environment element $\{\mathbf{W_b}\}$

The product and environment are updated in Equation (8-9).

$$\begin{aligned}&\mathbf{B} = \{\mathbf{F}_f \cup \mathbf{F}_h, \mathbf{F}_r^u, \mathbf{F}_r^d, \mathbf{G}_b, \mathbf{h}_w, \mathbf{x}_{mf}, \mathbf{x}_f, \mathbf{x}_s, \mathbf{x}_t, \mathbf{x}_{mt}\} \\ &\mathbf{S} = \{\mathbf{h}_w, \mathbf{Support}\}\end{aligned} \tag{8-9}$$

8.4.3 Altering the way of conflict resolution

As was described in the formal design process model **Design($\mathbf{R}^d$,S)** in Section 6.3, performance conflicts may emerge in adding a primitive product to the previously finished partial product. This was manifested in $\mathbf{X}_l \updownarrow \mathbf{X}_p$. The resolution of this conflict has been a challenging problem in design practice as well as in design research. Obviously, in each run of environment decomposition and product combination, different considerations of conflict resolution will give rise to different new design requirements for the succeeding design. Generally speaking, there could be three approaches to resolving the conflicts. 1) the newly generated primitive product can be modified so that it conforms with the partial structure; 2) an alternative primitive product may be selected from the candidate solutions; and 3) the finished partial structure may also be redesigned. Since this topic is not the concern of this thesis, more specific analysis can be found in Haroud *et al* (1995), Bahler *et al* (1995), Otto and Antonsson(1991), Oh and Sharpe(1995), Brazier, *et al*(1995), Reddy *et al*(1996).

8.5 Concluding Remarks

The paradox in studying design creativity is expressed by the following question: why do human beings create new things merely with the existing deterministic knowledge at their hands? It is similar to an argument in chaotic dynamics: chaos provides a mechanism that allows for free will within a world governed by deterministic laws. This association leads us to count on the theory of dynamical systems to investigate the design creativity.

The current research analyzed the nonlinear characteristics of design problem solving and assumed that designing is a nonlinear dynamic process. Design requirements, product environment, product descriptions, and product performances constitute the state of the dynamic system. Design governing equation is the dynamic of this process. Based on the conjecture, the randomness of design creativity is naturally inevitable, which comes from the prone-to-change nature of design requirements in design process. Design creativity is then attributed to the emergence of different attractors under the random redefinition of design requirements in each individual's design process.

Having these conclusions in mind, three potential routes to creative designs are pointed out: extending primitive design knowledge set, changing the sequence of environment decomposition, and altering the way of conflict resolution. The implementation of these three routes largely depends in designer's creativities. They are stemmed from the way to change the definition of design requirement. However, in any sense they are still just the necessary conditions for design creativity since the conditions to chaos are far more complex than what were described here.

Chapter 9 Application: Development of A Computer-Aided Mechanism Design System

9.1 Introduction

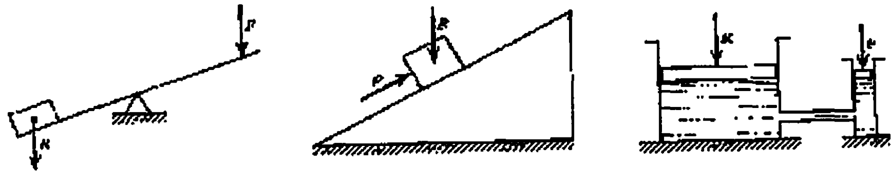
The objective of this chapter is to demonstrate how the theory established in this thesis can be used to serve the purpose of "designing the design". An example is given from the development of a computer aided mechanism design system. This example aims to design a mechanism to change the direction of straight-line motions. Although the problem is very simple, the underlying principles can be used for more complicated device design. The difficulty for the extension lies in the complexity of data and processes, not the logic of the problem solving process.

The next section will review the general machine design. It is summarized based on two textbooks (Hindhede, *et al*, 1983; Vinogradov, 2000). Section 9.3 will present a software system for solving a mechanism design problem using the models developed in this thesis. This problem is adapted from Greenwood (1961). In Section 9.4, concluding remarks are made.

9.2 Machine Design

A machine is a device consisting of a frame with various fixed and moving parts, which can transmit power, modify force or motion, and do useful work. Machines can be classified broadly as basic or simple and complex or compound. Simple machines correspond to the primitive products defined in this thesis. A simple machine is a device with at least one mechanically actuated member. A complex machine is a combination of simple machines.

There are three major categories of simple machines according to the physical principles behind them: 1) Lever action; 2) The inclined plane; 3) Pascal's principle of equal pressure distribution in fluids. These principles are illustrated in Figure 9-1.



a) Lever action b) The inclined plane c) Pascal's principle

Figure 9-1 Basic principles of simple machines (Hindhede, *et al*, 1983)

They are primitive products for machine design.

$$S^{a} = \{\text{lever\_action\_machines, inclined\_plane\_macchines, hydraulic\_machines}\} \tag{9-1}$$

Simple machines provide a mechanical advantage (MA) by increasing force at the expense of speed or augment speed at the expense of force.

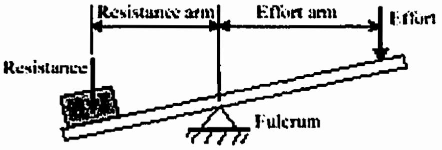
$$\mathbf{MA} = \frac{\textbf{resistance}}{\textbf{effort}} \tag{9-2}$$

where the applied force is called effort, and the load to be overcome is termed resistance. A mechanical advantage greater than one indicates an increase of force at the expense of speed.

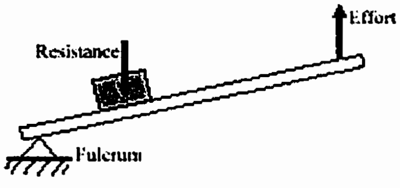
The mechanisms based on lever action generate little friction and therefore have high efficiency. In contrast, those based on the inclined plane have high friction and low efficiency. The hydraulic press lies somewhere between the two with regard to friction and efficiency. Friction plays a dual role in the three simple machines based on the inclined plane. It lowers efficiency but compensates by adding a useful self-locking feature not found in those based on the lever.

Lever action embodies four simple machines: lever, wheel and axle, pulley, and gearing.

$$\mathbf{lever\_action\_machines} = \{\mathbf{lever, wheel\ and\ axle, pulley, gearing}\} \tag{9-3}$$



a) First-class lever



b) Second-class lever

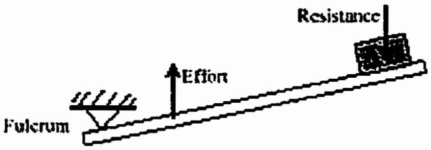


Figure 9-2 Simple machine: lever (Hindhede, *et al*, 1983)

A lever is rigid bar that is free to pivot about an axis through a point called the fulcrum. In terms of the location of the pivot point in relation to the resistance, three classes of levers are generally recognized, as is shown in Figure 9-2. In each class there is an advantage to be gained in either force or distance.

Wheel and axle is a circular lever capable of rotating instead of oscillating around its fulcrum. It consists of a circular member or crank rigidly attached to the axle, which turns with the wheel. Typically, it is shown in Figure 9-3. They may be used to magnify either the applied force or motion. When the applied force acts on the wheel, a large mechanical advantage is obtained. Acceleration takes place when the input effort is applied to the axle.

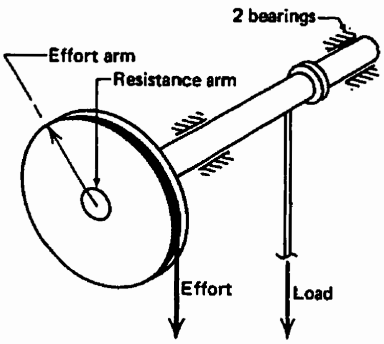


Figure 9-3 Simple machine: wheel and axle (Hindhede, *et al*, 1983)

In its simplest form the pulley consists of a single grooved wheel or sheave turned by means of a rope or chain partially confined to the groove. Pulley systems are a means of changing the effort and the speed of the load. They can also be used to change the direction of force to make the force more useful.

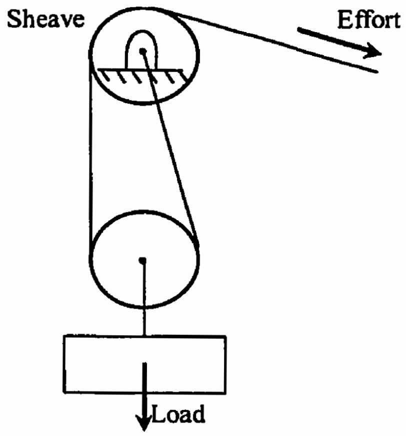
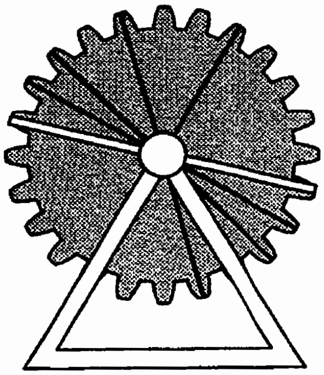
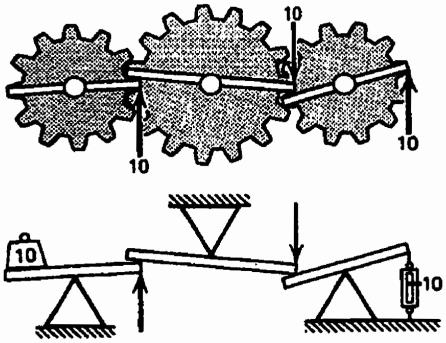


Figure 9-4 Simple machine: pulley



(a) A gear is essentially a first-class lever with arms of equal length

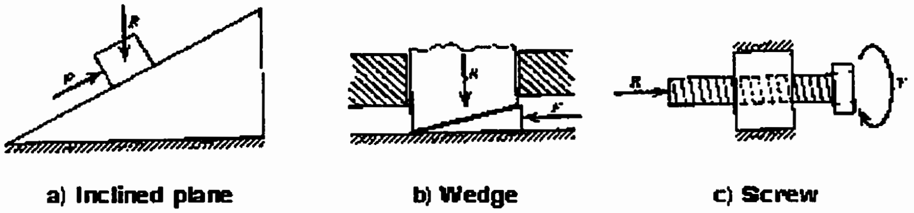


(b) A pair of gears is essentially a set of spinning levers acting in turn

Figure 9-5 Simple machine: gears (Hindhede, *et al*, 1983)

Gears are wheels with mating teeth cut in the rim or surface so that one can turn the other without slippage. The lever is basic to gearing. A pair of gears is essentially a set of spinning levers each with two equal arms acting in turn. Gears can be mounted on parallel, intersecting, or nonintersecting shafts. Consequently, they can: 1) change the plane of rotation; 2) increase or decrease the speed of applied motion; 3) magnify or reduce the applied force; and 4) provide a drive without slippage.

Based on the principle of the inclined plane, there are three simple machines: 1) inclined plane; 2) wedge; and 3) screw. They are shown in Figure 9-6.



a) Inclined plane b) Wedge c) Screw

Figure 9-6 Simple machine: the inclined plane (Hindhede, *et al*, 1983)

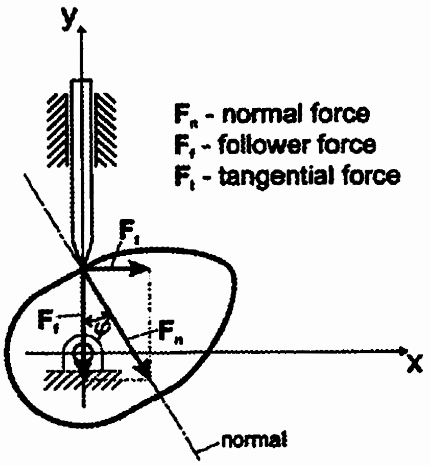


Figure 9-7 Simple machine: cam (Vinogradov, 2000)

Cam is a special form of wedge, which is shown in Figure 9-7.

Figure 9-8 shows the simple machine based on Pascal's principle.

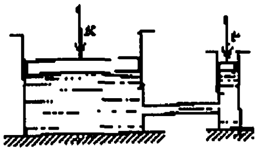


Figure 9-8 Simple machine: hydraulic machine (Hindhede, *et al*, 1983)

The concept of simple machines is useful in obtaining fundamental knowledge of machines. But not all machines can be reduced to simple machines. Spring design is based on Hooke's law, not on any of the simple machines. The theory of sliding bearings is based on hydrodynamics, not on simple machines.

The task of machine design is to make use of these simple machines to fulfill the complex functional requirements. They usually come from:

1) kinematic analysis. This establishes the necessary motion requirements;
2) force analysis. This establishes the magnitude of the acting forces and moments;
3) strength and rigidity analysis. This establishes the basic dimensions of machine components.

Kinematic analysis defines most of requirements for conceptual design. The example in the following section will only consider this part.

9.3 Environment Decomposition-Based Mechanism Design

9.3.1 Basic problem

The problem that we are going to solve is to change straight-line motion direction as shown in Figure 9-9.

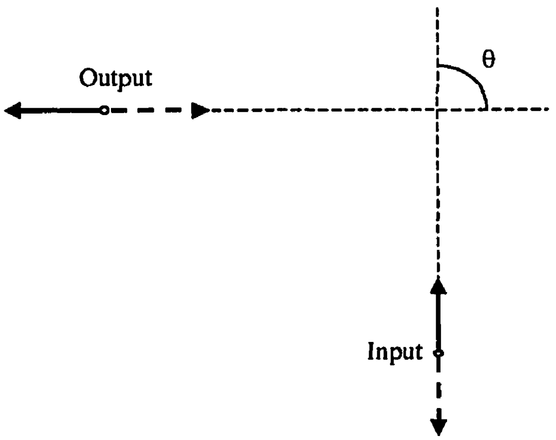


Figure 9-9 Problem of changing line motion(θ is generally close to 90°) (Greenwood, 1961)

9.3.2 Software architecture

A software architecture is adapted from Figure 6-9 to solve this problem. It is shown in Figure 9-10. The details of this architecture were discussed in Section 6-4. The following subsetions will explain each part with regard to the current problem. Figure 9-11, Figure 9-12, and Figure 9-13 are the interfaces for problem definition and solving.

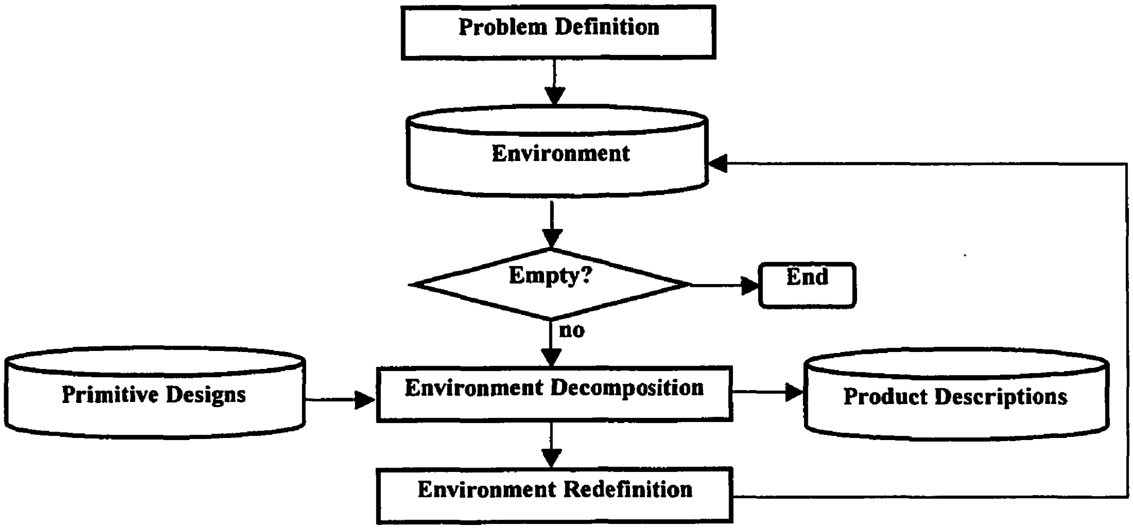


Figure 9-10 Architecture of mechanism design system

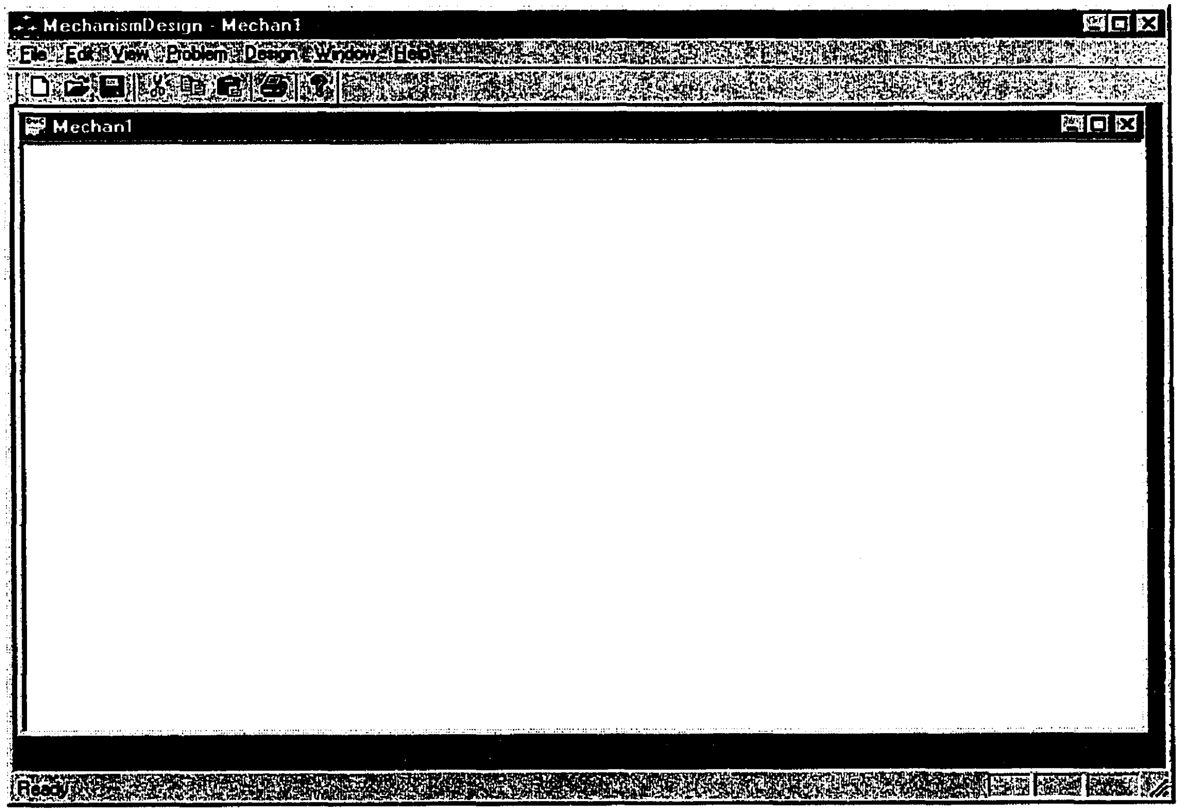


Figure 9-11 Interface of the mechanism design system

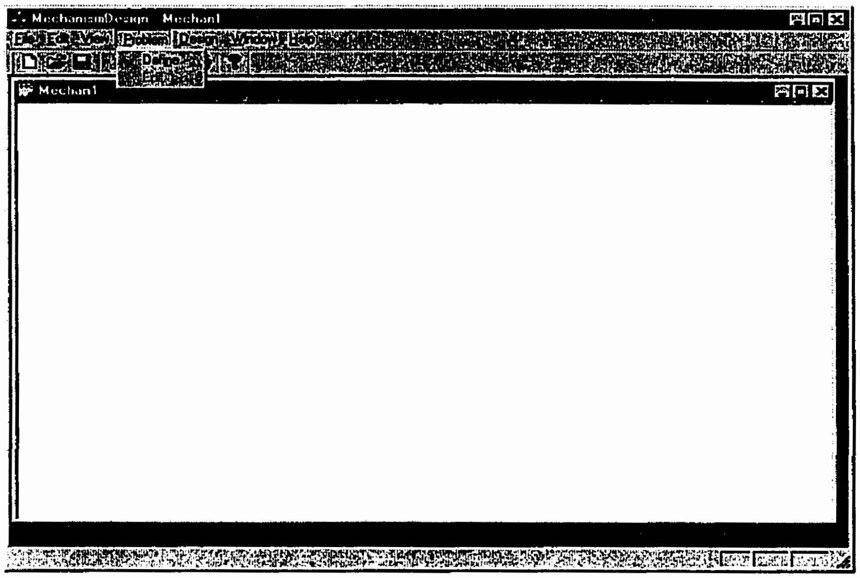


Figure 9-12 Interface for the design problem definition

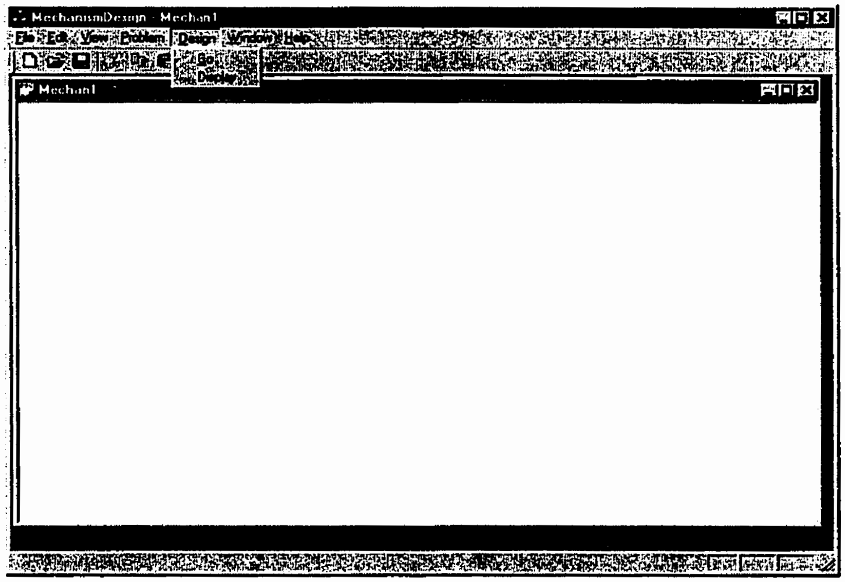


Figure 9-13 Interface for the design problem solving

9.3.3 Problem definition

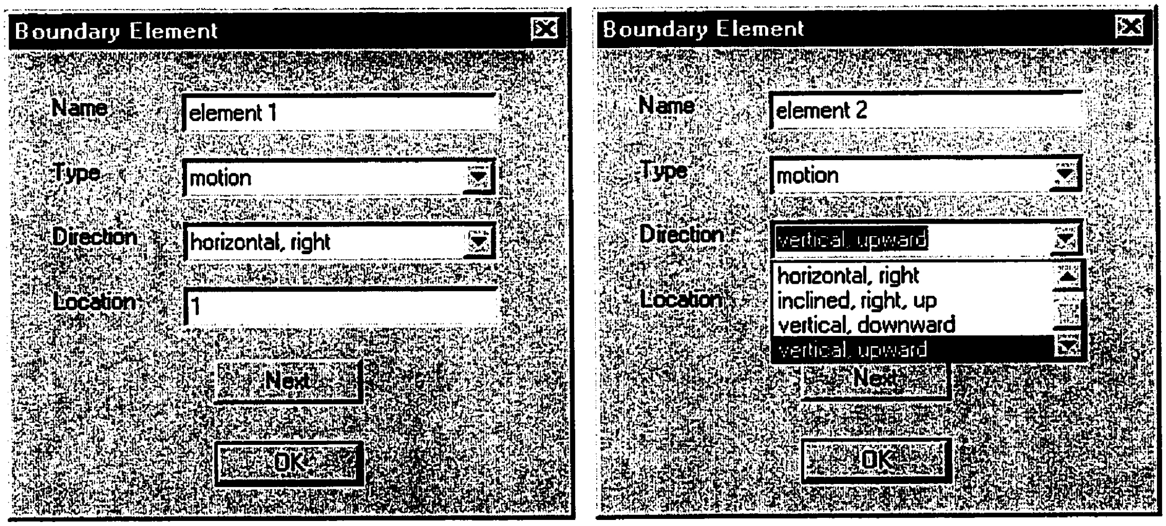
According to Equation (5-10), the design problem can be defined by environment, which consists of all actions and responses. In this case, they are input and output, i.e. vertical and horizontal sliding, for the expected mechanism. They are denoted as $\mathbf{d}_v$ and $\mathbf{d}_h$, respectively. Therefore, the environment for this problem can be represented as:

$$\begin{aligned} &\mathbf{B[0]} = \{\bar{\mathbf{d}}_v, \bar{\mathbf{d}}_h\} \\ &\bar{\mathbf{d}}_v = \{<\mathbf{direction, vertical}>, <\mathbf{value, real\_number}>\} \\ &\bar{\mathbf{d}}_h = \{<\mathbf{direction, horizontal}>, <\mathbf{value, real\_number}>\} \end{aligned} \tag{9-4}$$

Corresponding to the current problem definition, the product descriptions can be written as

$$\mathbf{S[0] = Mechanism} \tag{9-5}$$

Figure 9-14 shows the interface to input the problem. Each input and output has four members: name, type, direction and location.



Name: element 1
Type: motion
Direction: horizontal, right
Location: 1

Name: element 2
Type: motion
Direction: vertical, upward
- horizontal, right
- inclined, right, up
- vertical, downward
- vertical, upward

Figure 9-14 Dialog box for inputting the design problem

9.3.4 Primitive design

In terms of the environment decomposition based design process model presented in Section 6.4, primitive design is indispensable to solve any design problems. This section will list some useful primitive designs for the given design problem. UML modeling language

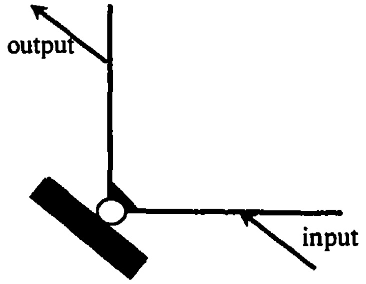
(Booch, Rumbaugh, and Jacobson, 1999) is used to illustrate each primitive design. This provides an easy transition to the software development.

Each primitive design can be defined by a class in UML, which has three parts: name, attributes, and responsibilities, as is shown in Figure 9-15.

Name
Attributes
Responsibilities

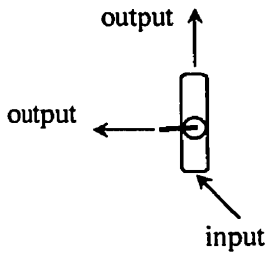
Figure 9-15 Architecture of a class in UML

The name defines the class uniquely, making it distinguishable from the others. The attributes define the class's properties. The responsibilities list its performances which include performance knowledge. The following figures will define some of the primitive designs for the example. Since responsibilities are not related to this example, they are omitted from the figure.



Name: No.1
Attributes: 1. Shape: L; 2. Input: inclined sliding; 3. Output: inclined sliding; 4. ……
Responsibilities: ……

Figure 9-16 L-shaped lever with a moving pivot



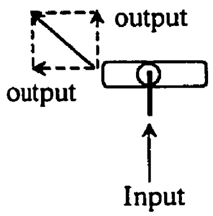
Name: No.2.1

Attributes:
1. Shape: vertical slot;
2. Input: inclined sliding;
3. output 1: vertical sliding;
4. output 2: horizontal sliding;
5. ……

Responsibilities:

……

Figure 9-17 A slot: case 1



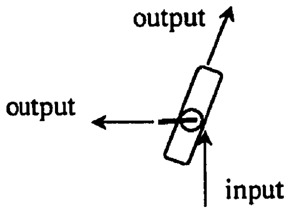
Name: No.2.2

Attributes:
1. Shape: horizontal slot;
2. Input: vertical sliding;
3. output 1: vertical sliding;
4. output 2: horizontal sliding;
5. ……

Responsibilities:

……

Figure 9-18 A slot: case 2



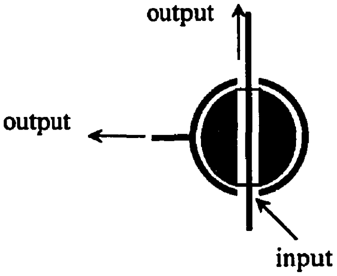
Name: No.2.1

Attributes:
1. Shape: inclined slot;
2. Input: vertical sliding;
3. output 1: inclined sliding;
4. output 2: horizontal sliding;
5. ……

Responsibilities:

……

Figure 9-19 A slot: case 3



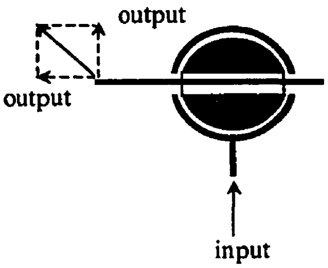
Name: No.3.1

Attributes:
1. Shape: vertical spherical bearing;
2. Input: inclined sliding;
3. output 1: vertical sliding;
4. output 2: horizontal sliding;
5. ……

Responsibilities:

……

Figure 9-20 A spherical bearing: case 1



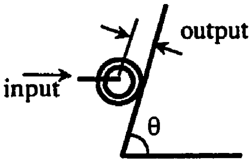
Name: No.3.2

Attributes:
1. Shape: horizontal spherical bearing;
2. Input: vertical sliding;
3. output 1: vertical sliding;
4. output 2: horizontal sliding;
5. ……

Responsibilities:

……

Figure 9-21 A spherical bearing: case 2



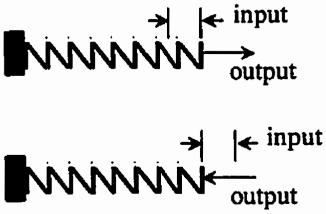
Name: No.4

Attributes:
1. Shape: roller and plane;
2. Input: horizontal force;
3. output: relative movement in the normal direction;
4. ……

Responsibilities:

……

Figure 9-22 Contact between roller and plane



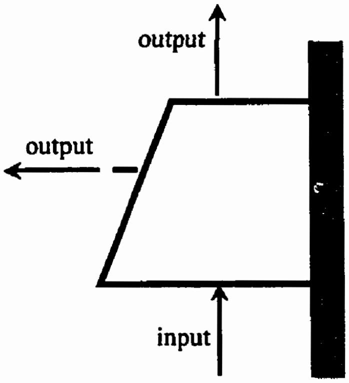
Name: No.5

Attributes:
1. Shape: spring;
2. Input: deformation;
3. Output: force;
4. ……

Responsibilities:

……

Figure 9-23 Spring



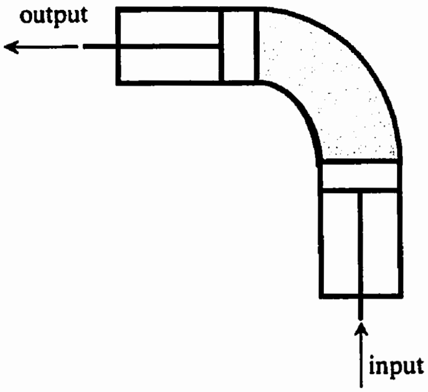
Name: No.6

Attributes:
1. Shape: wedge;
2. Input: vertical sliding;
3. Output 1: vertical sliding;
4. Output 2: horizontal drive
5. ……

Responsibilities:

……

Figure 9-24 Sliding wedge



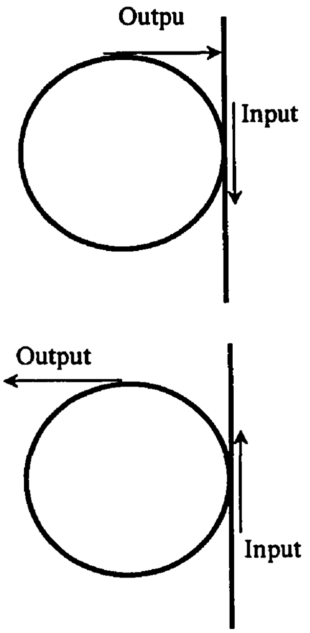
Name: No.7

Attributes:
1. Shape: pistons and fluid vessel;
2. Input: vertical sliding;
3. Output: horizontal sliding;
4. ……

Responsibilities:

……

Figure 9-25 Fluid coupling



Name: No.8.1

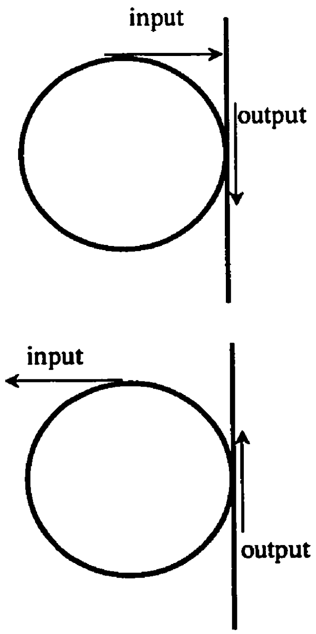
Attributes:

1. Shape: rack and pinion;
2. Input: vertical sliding of rack;
3. Output: rotation of pinion;
4. ……

Responsibilities:

……

Figure 9-26 Rack and coupled pinion



Name: No.8.2

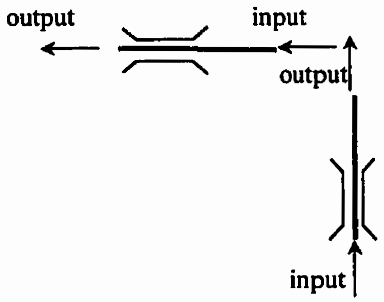
Attributes:

1. Shape: rack and pinion;
2. Input: rotation of pinion;
3. Output: vertical sliding of rack;
4. ……

Responsibilities:

……

Figure 9-27 Rack and coupled pinion



Name: No.9

Attributes:
1. Shape: straight;
2. Input: sliding;
3. Output: sliding;
4. ……

Responsibilities:

……

Figure 9-28 Rod

9.3.5 Environment decomposition process

An example will be given to show how environment decomposition process is applied to find the solution. The environment and product descriptions will be updated in this process.

1) Extracting $\vec{\mathbf{d}}_v$ from the environment set in equation (9-4), searching through the primitive designs, the following primitive designs could be found:

$$\mathbf{S}_a = \{\mathbf{No.2.2, No.2.3, No.3.2, No.6, No.7, No.8.1}\} \qquad (9\text{-}6)$$

2) If **No.2.2** is picked as a candidate, then product descriptions and environment can be updated as

$$\mathbf{S} = \{\mathbf{No.2.2, Vertical\ Rod}\} \qquad (9\text{-}7)$$
$$\mathbf{B} = \{\vec{\mathbf{d}}_h, \vec{\mathbf{d}}_l\}$$

where $\vec{\mathbf{d}}_l$ is shown in Figure 9-29.

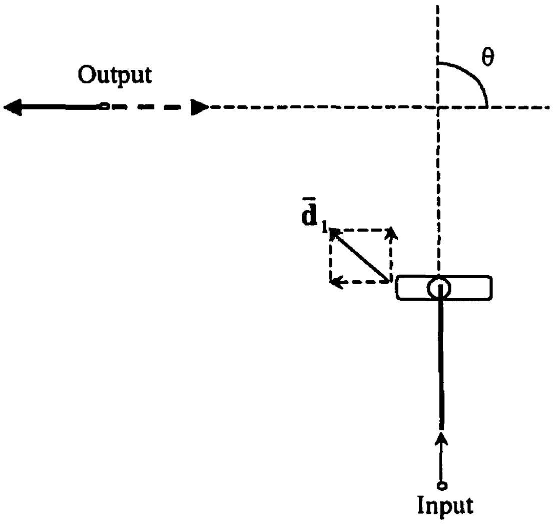


Figure 9-29 Updated product and environment

3) Since environment set is not empty yet, the process needs to continue. Extracting $\vec{\mathbf{d}}_h$ from the environment set in equation (9-7), searching through the primitive designs, the following primitive designs could be found:

$$\mathbf{S_a = \{No.2.1, No.2.3, No.3.1, No.6, No.7,\ No.8.1\}} \tag{9-8}$$

4) If **No.2.1** is picked as a candidate, then product descriptions and environment can be updated as

$$\begin{aligned}&\mathbf{S = \{No.2.1, No.2.2, Vertical\ Rod\}}\\&\mathbf{B = \{\vec{d}_2, \vec{d}_1\}}\end{aligned} \tag{9-9}$$

where $\vec{\mathbf{d}}_2$ is shown in Figure 9-29.

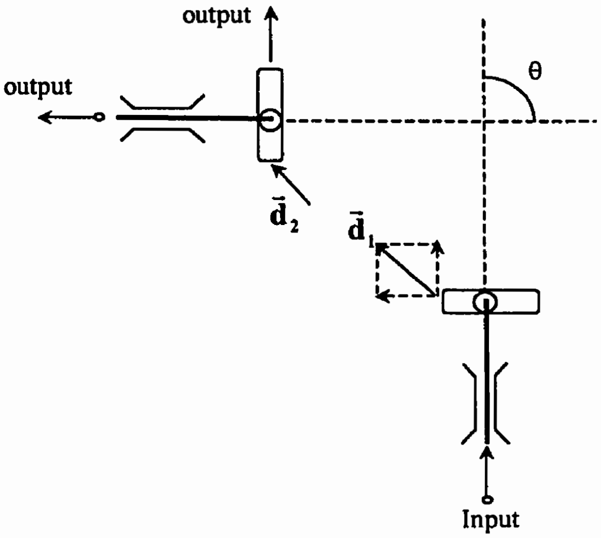


Figure 9-30 Updated product and environment

5) Again, searching through the primitive designs, the following primitive designs could be found:

$$\mathbf{S_a = \{No.1\}} \tag{9-10}$$

6) Product descriptions and environment can be updated as

$$\begin{aligned}&\mathbf{S = \{No.1, No.2.1, No.2.2, Vertical\ Rod\}} \\ &\mathbf{B = \{\}}\end{aligned} \tag{9-11}$$

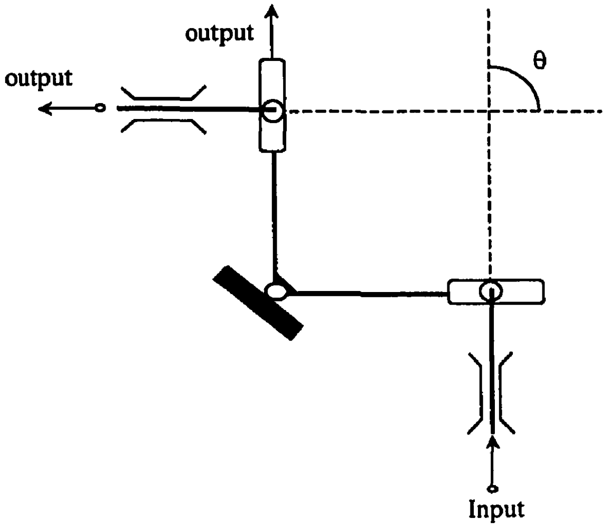
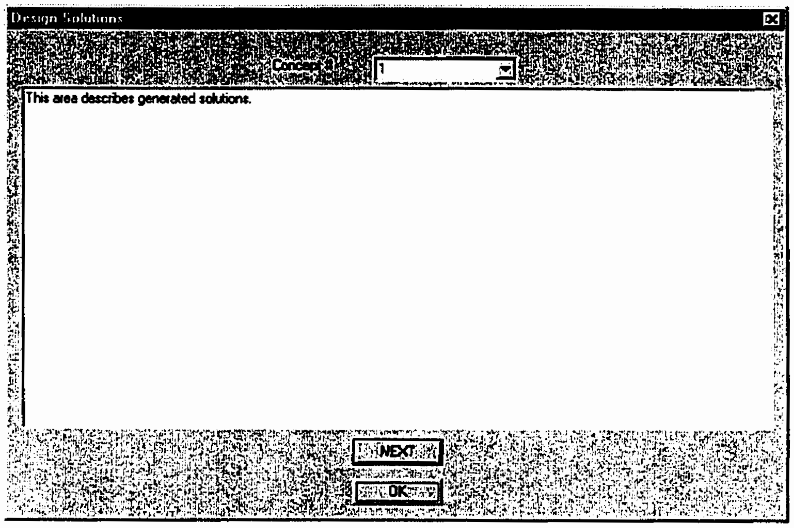


Figure 9-31 Updated product and environment

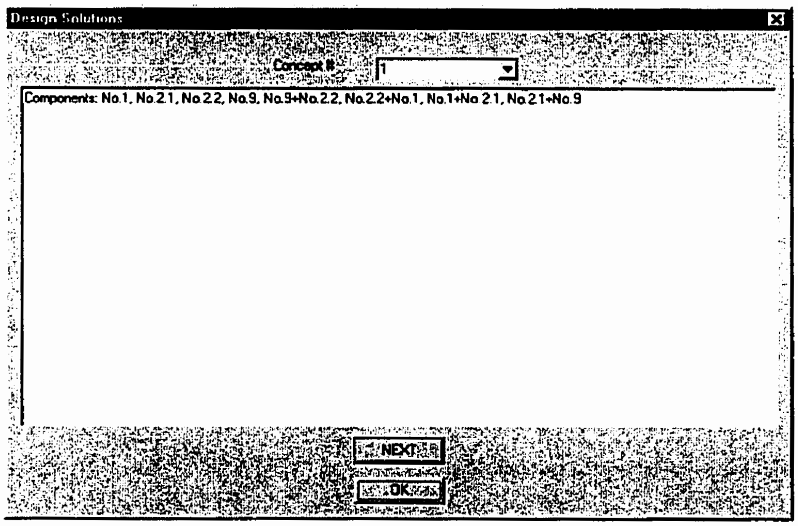
7) Since environment set is empty now, the design process ends at this point. S in Equation (9-11) is a design solution.

Figure 9-32 is the interface for showing the design solutions. Figure 9-33 is the description for the concept shown in Figure 9-31. It is constituted by a list of components. The symbol "+" represents the connection between two components. In this representation, No.9 and No.2.2, No.2.2 and No.1, No.1 and No.2.1, No.2.1 and No.9 are connected, respectively. The detailed documentation of this model is not given, since it only involves the implementation using different data models. It can vary according to the styles of different developers. The basic underlying principles can be the same.



This area describes generated solutions.

Figure 9-32 Interface for displaying design solutions

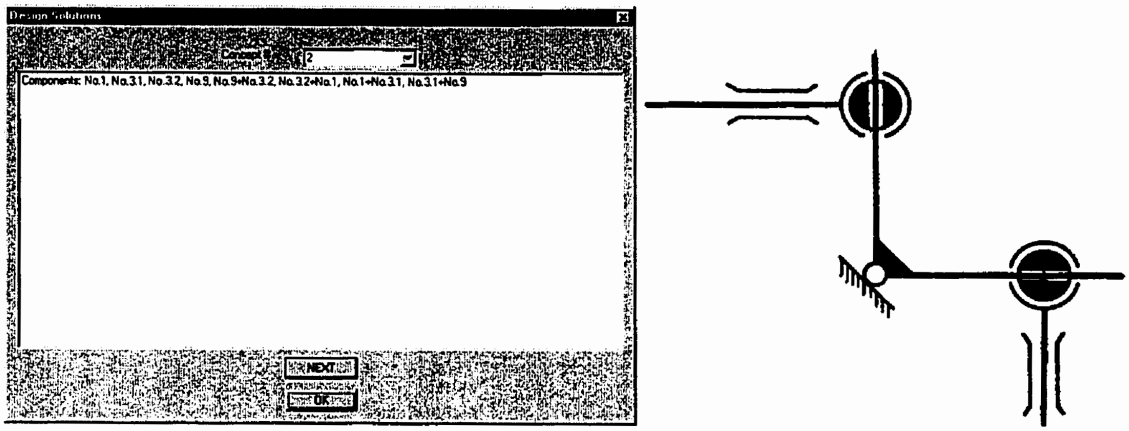


Components: No.1, No.2.1, No.2.2, No.9, No.9+No.2.2, No.2.2+No.1, No.1+No.2.1, No.2.1+No.9

Figure 9-33 Descriptions for design concept 1#

9.3.6 Examples

The following are some of other candidate mechanisms satisfying the requirements, generated from different primitive designs. The screen print-out for design concept 2# is attached. Others have the similar form, so only the graphic descriptions are given. Some comments on the candidates are also given following each figure.



Concept #: 2

Components: No.1, No.3.1, No.3.2, No.9, No.9+No.3.2, No.3.2+No.1, No.1+No.3.1, No.3.1+No.9

Figure 9-34 Concept #2: design with spherical bearings(Greenwood, 1961)

This was generated from primitive designs No.1, No.3.1, No.3.2, and No.9.

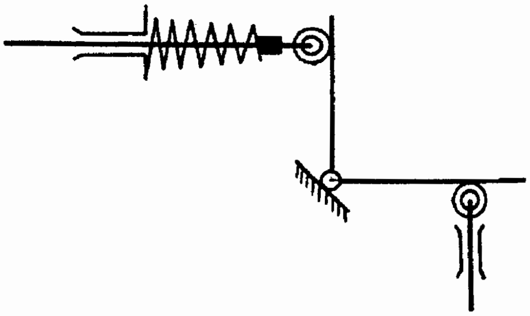


Figure 9-35 Design with spring-loaded lever (Greenwood, 1961)

This was generated from primitives No.1, No.4, No.5, and No.9. Rollers are used to protect the surfaces of lever surfaces. A maximum movement is controlled by the loading spring.

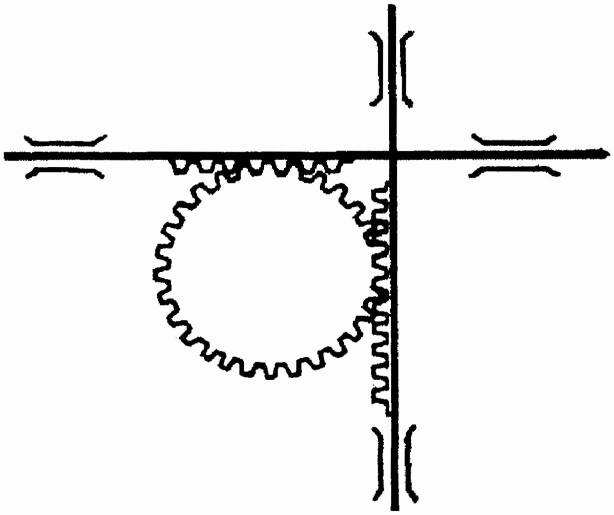


Figure 9-36 Design with racks and coupled pinions (Greenwood, 1961)

This was generated from primitive designs No.8.1 and No.8.2. They can be replaced by friction surfaces for low-cost set-up.

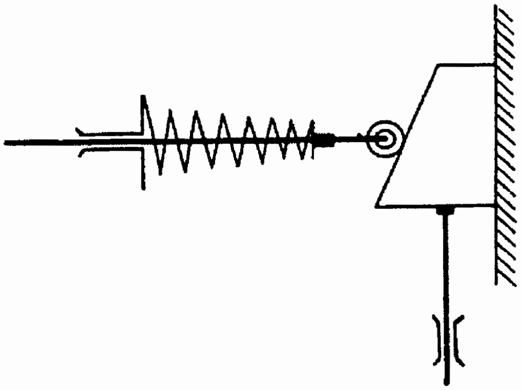


Figure 9-37 Design with sliding wedge (Greenwood, 1961)

This was generated from primitive designs No.4, No.5, No.6, and No.9. Spring-loaded follower is needed to keep the contact between the follower and the wedge. Low friction is less essential with roller follower.

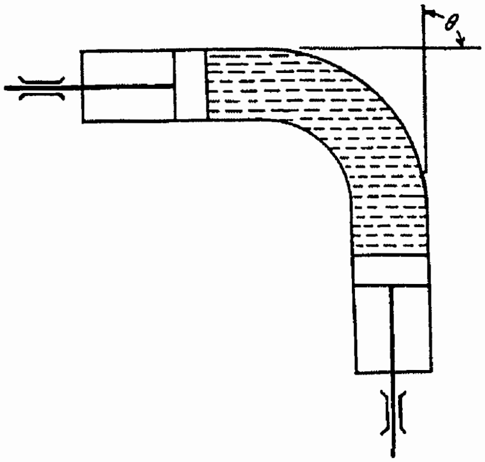


Figure 9-38 Design with fluid coupling (Greenwood, 1961)

This was generated from primitive designs No.7 and No.9. It is simple and allows motion to be transmitted through any angle. Leak problems and accurate piston-fitting can make method more expensive than it appears to be. Also, although action is reversible it must always be a compressive one for best results.

9.4 Concluding Remarks

Using the theory established in this thesis, this chapter summarized general components of machine design. A software prototype focused on mechanism design is presented to show the potential usefulness of this work, serving the research scheme proposed in Figure 1-5. Generally speaking, the present theory can support software development of product conceptual design in the following aspects:

1) Organizing design knowledge;

2) Modeling design requirements;

3) Providing product representation scheme; and

4) Constructing the framework of design process.

This simple mechanism design software can generate multiple design concepts automatically for the specific design problems. The only resource is performance knowledge. More sophisticated and potentially useful design software can be developed using the same principles. This work shows that mathematical approach to design studies can be useful not only for intellectual exploration but also for practical application.

Chapter 10 Conclusions and Future Directions

10.1 Conclusions

Figure 1-5 gives a research scheme for establishing an axiomatic system of studying design. Four specific aims and corresponding assumptions were listed in Section 1.4. This section will summarize the research results in achieving those aims and check the underlying assumptions. The following conclusions can be made about this research:

1) **Solutions to specific aim 1**. An axiomatic system of product design was established in Chapter 3. This axiomatic system includes two axioms: axiom of bounded rationality and axiom of object structuring. The first axiom states the limitation of human recognition while the second one tells what a perfect recognition should include. The contradicting nature of these two axioms forms the foundation of the design problem. Set theory was taken as the language of representing the axioms, theorems, and facts appeared in the axiomatic system. The whole discussion shows that design, as a phenomenon, can be approached scientifically and mathematical tools are powerful in studying design.

2) **Solutions to specific aim 2.** Theorems about product design were derived from the established axiomatic system following logical steps. These theorems include the definition and nature of the product-environment system, product performance, partition of product properties, design requirements, design analysis and synthesis processes, as well as design governing equation. These theorems were investigated in more details in

separate chapters. New models of product descriptions, design requirements, and design processes were established, which constitute an environment decomposition based conceptual design process model. This model captures some rational element underlying the conceptual design process, which has been viewed as a process full of intuition, randomness, and uncertainty. The research results show that well chosen axioms will lead to meaningful theorems.

3) **Solutions to specific aim 3**. Running examples were used from Chapter 4 through Chapter 8. The theory has naturally supported the representation of design problem, design objects, and design processes. Chapter 7 derived the design governing equation. This equation naturally embodies the ill-structure nature of design problem. Chapter 8 showed how the derived design process model and design governing equation imply design creativity. These two chapters tested and justified the established axiomatic system logically.

4) **Solutions to specific aim 4**. A prototype of a simple mechanism design software system was developed to implement a mechanism design example. The principles and architecture of this software are based on the derived theorems in this thesis. This justified the theory from application regard while it also demonstrated the usefulness of the theory.

Besides the above four conclusions, some lessons could be learnt from this research:

5) **Design vs. science.** If we could dig into Chapters 7 and 8, we would be able to say that design activity is different from scientific activity. But this does not exclude the fact that

design activity is a phenomenon that can subject to scientific investigation. The study of design activities and the design activity itself are in two different levels and belong to different categories of disciplines.

6) **Logic vs. randomness**. If the endeavor of scientific investigation into design activity succeeds, then a natural result would be that design activity could be performed following logical steps. This in turn leads to the conclusion that design activity would be scientific activity, and would be mathematical activity. This is indeed the conflict between logic and randomness, determinism and uncertainty. My endeavor in Chapter 8 has in fact turned the problem into the dispute between classic science and contemporary nonlinear sciences. An attempt in solving this dispute actually requires a more pragmatic position at the present time.

10.2 Future work

This research is just a start in the preliminary scientific investigation of design activities. The following work is necessary for future research:

1) This thesis tried more to make the idea understandable than to make it consistent and strict. A compromise is the use of naïve set theory rather than the axiomatic set theory. Representing levels of abstraction and complexity with naïve set theory is not theoretically beautiful, though practically reasonable. The concept of class from Russell can be adapted for this research. The axiom of object structuring could also be more naturally formulated.

2) The research of design methodology has had a reasonable history. There has been some important empirical research. The methods and results in this thesis can be used to formulate those methodologies. It will be beneficial for both sides. This axiomatic system can be tested and justified and evolved through this empirical comparison. The existing methodologies, on the other hand, can be subjected to more logical filtering. Implied problems and contradictions will be exposed for further improvement of the methodologies.

3) Though this thesis aimed to model conceptual design process, the establishment of the theory did not rely on any assumptions about conceptual design. The possibility of applying this theory to other stages of design should be investigated.

4) More general application to design education and computer aided design software development need to be explored.

5) Positive philosophical investigation into the foundation of this research (such as dispute between logic and randomness, as suggested above) will be valuable for the advance of this subject.

References

Agarwal,M. and Cagan,J.(1997), Shape grammars and their languages-a methodology for product design and product representation, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 14-17, 1997, Sacramento, California, DETC97/DTM-3867

Akin, O.(1986), *Psychology of Architectural Design*, Pion Limited, London

Akin,O.(1990), Necessary conditions for design expertise and creativity, *Design Studies*, Vol. 11, No. 2, pp. 107~113

Akin,O.(1993), Architects' reasoning with structures and functions, *Environment and Planning B: Planning and Design*, Vol. 20, pp. 273~294

Allan, J.K. and Mistree, F.(1997), "Decision-based design: Where should we go from here?, Notes of 1997 Decision-Based Design Workshop

Allen,K.R. and Skalak,S.C.(1998), Defining product architecture during conceptual design, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 13-16, 1998, Atlanta, GA, DETC98/DTM-5650

Altshuller,G.S. (1988), *Creativity as an Exact Science*, Gordon and Breach Science Publishers, New York, 1988

Amarel,S.(1966), On the mechanization of creative process, *IEEE Spectrum*, April, pp. 112~114

Amen,R.,Rask,I.,Sunnersjo,S.(1999), Matching design tasks to knowledge based software tools, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 12-15, 1999, Las Vegas, Nevada, DETC99/DAC-8688

Andreasen,M.(1991), Design methodology, *Journal of Engineering Design* **2**(4), 321-335

Bahler.D.,Dupont,C. and Bowen,J.(1995), Mixed quantitative/qualitative method for evaluating compromise solutions to conflicts in collaborative design, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **9**, 325-336

Braha,D. and Maimon,O. (1998), *A Mathematical Theory of Design*, Kluwer Academic Publishers

Brazier,F.M.T.,Langen,P.H., and Treur, J.(1995), Modeling conflict management in design: an explicit approach, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **9**, 353-366

Chittaro,L. and Kumar,A.N. (1998), Reasoning about function and its applications to engineering, *Artificial Intelligence in Engineering*, **12**, 331-336

Clausing,D.(1994), *Total Quality Development*, ASME Press, New York, USA.

Coulter,S. and Bras,B.(1999), Decision support for systematic product evolution, DETC99/DTM-8747

Cross,N (ed.) (1984), *Developments in Design Methodology*, Wiley, Chichester

Cross,N.(1992), Research in design thinking, in Cross,N., Dorst,K. and Roozenburg,N. (ed.), *Research in Design Thinking*, pp.3-10, Delft University Press.

Cross,N., Dorst,K. and Roozenburg,N.(1992), *Research in Design Thinking*, Delft University Press.

Cross, N. (1993), Science and design methodology: a review, *Research in Engineering Design*, 5:63-69.

Crutchfield,J.P. ***et al*** (1986), Chaos, Scientific American, Vol. 255, No. 6 pp. 38~49

Davis,M.D. and Weyuker,E.J.(1983), *Computability, Complexity and Languages,* Academic Press.

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

Dixon, J. and Poli, C.(1995), *Engineering Design and Design for Manufacture*, Field Stone, USA

Earl,C.F.(1998), The structure of designs, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 13-16, 1998, Atlanta, GA, DETC98/DTM-5656

Eastman,C.M., Bond,A.H., and Chase,S.C. (1991), A Formal Approach for Product Model Information, *Research in Engineering Design*, **2**, 65-80

Eastman,C.M., Bond,A.H., and Chase,S.C. (1991), Application and evaluation of an engineering data model, *Research in Engineering Design*, **2**, 185-207

Eastman,C.M. and Fereshetian,N.(1994), Information models for use in product design: a comparison, *Computer Aided Design*, **26**(7), 551-572

Eder,W.E. (ed.) 1995, *Engineering Design and Creativity*, Proceedings of the Workshop EDC, State Scientific Linrary, Pilsen, Czech Republic, 16-18 November, 1995

El-Haik,B. and Yang,K.(1999), Measures of complexity in design, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 12-15, 1999, Las Vegas, Nevada, DETC99/DAC-8569

Fauvel,O.W.(1991), Expanded use of function language in mechanical design, in N. Popplewell and A.H. Shah, editors, *Proceedings of the Thirteenth Canadian Congress of Applied Mechanics*, pp.692-693, Winnipeg

Fey,V.R., Rivin,E.R. and Vertkin,I.M.(1994), Application of the theory of inventive problem solving to design and manufacturing systems, *Annals of the CIRP*, **43**(1), 107-110

Finch,W. and Ward,A.(1997), A set-based system for eliminating infeasible designs in engineering problems dominated by uncertainty, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 14-17, 1997, Sacramento, California, DETC97/DTM-3886

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

Finger,S. and Dixon,J.R.(1989), A review of research in mechanical engineering design. Part II: representations, analysis, and design for the life cycle, *Research in Engineering Design*, **1**, 121-137

Galle,P.(1996), Design rationalization and the logic of design: a case study, *Design Studies*, **17**, 253-275

Gero,J. and Maher,M.L. (1997), A framework for research in computing, http://www.arch.usyd.edu.au/~john/publications

Gershenson,J.K. and Stauffer,L.A.(1996), The validation of a taxonomy for manufacturability design requirements, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, August 18-22, 1996, Irvine, California, 96-DETC/DTM-1514

Goel,A.K. and Chandrasekaran,B. (1992), Case-based design: a task analysis, in C.Tong and D.Sriram (eds.), *Artificial Intelligence in Engineering Design,* Vol.2, pp.65-71

Gomez-Senent,E.,Santamarina,J.L.,Chiner,M.,Aragones,P. and Capuz,S. (1997), The design dimensions: a design theory proposal, *Proceedings of ICED'97*, 467-475

Gorrti, S. R., Gupta,A., Kim,G.J., Sriram,R.D., and Wong,A. (1998), An object-oriented representation for product and design processes, *Computer Aided Design*, **30**(7), 489-501

Grabowski,H. and Benz,T. (1989), Implementing the design methodology, in *IFIP WG5.2 Workshop on Intelligent CAD*, pp.26-29, North-Holland, The Netherlands

Gu,P.(1998), Recent development in design theory and methodology research, *Proceedings of International Conference on Manufacturing Sciences*, pp.21-26, June, 1998, Wuhan, China.

Gu,P. and Sosale,S.**(2000), Product modularization for life cycle engineering, *Robotics and Computer Integrated Manufacturing*(to appear)**

Gu.,P.,Hashemian,M.,Sosale,S.(1997), An Integrated Modular Design Methodology for Life-Cycle- Engineering, *Annals of the CIRP*, **46**, 71-74

Gui,J. and Mantyla,M.(1994), Functional understanding of assembly modeling, *Computer Aided Design*, **26**(6),435-451

Harmer,Q.J, Weaver P.M. and Wallace,K.M. (1998), Design-led component selection, *Computer Aided Design*, **30**(5), 391-405

Haroud,D.,Boulanger,S.,Gelle,E. and Smith,I.(1995), Management of conflict for preliminary engineering design tasks, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **9**, 313-323

Hashemian,M. (2000), Private communications

Hazelrigg,G.A.,(1996), *System Engineering: an Approach to Information-Based Design*, Prentice Hall

Henderson,M.R. and Taylor,L.E.(1993), A meta-model for mechanical products based upon the mechanical design process, *Research in Engineering Design*, **5**, 140-160

Hew,K.P,Washio,T.,Tomiyama,T. and Umeda,Y. (1999), The mathematical theory of synthesis design: foundation, framework and method, in G.Goldschmidt and W.Porter(eds.), *Proceedings of the 4th International Design Thinking Research Symposium on Design Representation*, pp.II.205-II.217

Hew,K.P, Tomiyama,T.,Washio,T. and Umeda,Y. (1999), Towards a systematic method of modeling synthesis, in K.Ueda(ed.), Proceedings of the International Workshop on Emergent Synthesis (IWES'99), pp.189-198, December 6-7, 1999, Kobe,Japan

Horowitz, R. Maimon, Oded.(1997), Creative design methodology and the SIT method, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 14-17, 1997, Sacramento, California, DETC97/DTM-3865

Hsiao, S. and Wang,H. (1998), Applying the semantic transformation method to product form design, *Design Studies*, Vol. 19, No.3, pp.309-330

Hsu,W. and Woon,I.M.Y.(1998), Current research in the conceptual design of mechanical products, *Computer-Aided Design*, **30**(5), 377-389

Hubka,V., Andreasen,M. and Eder,W.(1988), *Practical Studies in Systematic Design*, Butterworths

Hubka V.(1980), *Principles of Engineering Design*, Butterworth Scientific, London

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

Iwasakl,Y.,Vescovl,M. and Fikes,R.(1995), Causal functional pepresentation language with behavior-based semantics, *Applied Artificial Intelligence*, **9**, 5-31

Jahangir,E. and Frey,D.(1999), Differential entropy as a measure of information content in axiomatic design, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 12-15, 1999, Las Vegas, Nevada, DETC99/DTM-8751

Johannesson,H.L.(1996), On the nature and consequences of functional couplings in axiomatic machine design, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, August 18-22, 1996, Irvine, California, 96-DETC/DTM-1528

Johannesson,H.L.(1997), Application of interaction and functional coupling rules in configuration design, *Proceedings of The ASME Design Engineering Technical Conference* *and Computers in Engineering Conference*, September 14-17, 1997, Sacramento, California, DETC97/DTM-3869

Kalyan-Seshu,U. and Bras,B. (1998), Integrating DFX tools with computer-aided design systems, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 13-16, 1998, Atlanta, GA, DETC98/DAC-5621

Kersten,T.(1996), MODESSA: a computer aided conceptual design system, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, August 18-22, 1996, Irvine, California, 96-DETC/DTM-1083

Kirschman,C.F.,Fade,G.M. and Jara-Almonte,C.C. (1996), Classifying functions for mechanical design, *Proceedings of The 1996 ASME Design Engineering Technical Conference and Computers in Engineering Conference, August 18-22, 1996, Irvine, California,* 96-DETC/DTM-1504

Knoop,W.G. (1997), The influence of information characteristics on preliminary design concepts, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 14-17, 1997, Sacramento, California, DETC97/DTM-4314

Kota,S. and Lee,C.L. (1990), A functional framework for hydraulic systems using abstraction/decomposition hierarchies, in *Proceedings of the 1991 ASME International Computers in Engineering Conference and Exhibition*, 1, pp.327-340

Kusiak,A., Szczerbicki,E. and Vujosevic,R. (1991), Intelligent design synthesis: an object-oriented approach, *International Journal of Production Research*, 1991, **29**, 1291-1308

Kumar,A.N. and Upadhyaya,S.J.(1997), Component-ontological representation of function for reasoning about devices, *Artificial Intelligence in Engineering*, **12**, 399-415

Law,W.S. and Antonsson,E.K.(1996), Multi-dimensional mapping of design imprecision, *Proceedings of DETC'96*, 96-DETC/DTM-1524.

Lee,D.J. and Thornton,A.C.(1996), The identification and use of key characteristics in the product development process, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, August 18-22, 1996, Irvine, California, 96-DETC/DTM-1506.

Lee,J.,Ward,A.C. and Hu,S.J.(1996), A set-based approach to stamping designs, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, August 18-22, 1996, Irvine, California, 96-DETC/DTM-1295

Li,C.L.,Tan,S.T. and Chan,K.W. (1996), A qualitative and heuristic approach to the conceptual design of mechanisms, *Engineering Applications of Artificial Intelligence*, 1996, **9**, 17-31

Liang,W. and O'Grady,P. (1998), Design with objects: an approach to object-oriented design, *Computer-Aided Design*, **30**, 943-956

Liu,Y. (1996), Is designing one search or two? A model of design thinking involving symbolism and connectionism, *Design Studies*, **17**, 435-449

Lossack,R.S, Umeda,Y. and Tomiyama,T.(1998), Requirement, function and physical principle modeling as the basis for a model of synthesis, in A.Brawshaw and J.Counsell(eds.), *Computer Aided Conceptual Design'98, Proceedings of the 1998 Lancaster International Workshop on Engineering Design*, pp.165-179

Maher,ML.(1990), Process models for design synthesis, *Al Magazine*, Vol. 11, No. 4, pp. 49-58

Maimon,O. and Braha,D.(1996), On the complexity of the design synthesis problem, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 26, No. 1, pp.141-150

Malmqvist,J., Axelsson,R. and Johansson,M. (1996), A comparative analysis of the theory of inventive problem solving and the systematic approach of Pahl and Beitz, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, August 18-22, 1996, Irvine, California, 96-DETC/DTM-1529

March, L.J.(1984), The logic of design, in Cross, N.(ed.), *Developments in design methodology*, Wiley, Chichester

Marston,M., Bras,B., Mistree,F.(1997), The applicability of the axiomatic and decision-based design equations in variant design, *Proceedings of DETC'97*, DETC97/DTM-3884.

Mashburn,T.A and Anderson,D.C. (1991), An extensible computer environment for modeling and analysis in mechanical design, in *Proceedings of the 1991 ASME International Computers in Engineering Conference and Exhibition*, 1, pp.127-135

Messac,A. and Chen,W. (1998), The engineering design discipline : is its confounding lexicon hindering its evolution? *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 13-16, 1998, Atlanta, GA, DETC98/DTM-5658.

Mostow,J.,Barley,M. and Weinrich,T. (1992), Automated reuse of design plans in BOGART, *Artificial Intelligence in Engineering*, **4**, 181-196

Muller,W. and Pasman,G. (1996), Typology and the organization of design knowledge, *Design Studies*, **17**, 111-130

Mullins,S. and Rinderle,J.R. (1990), Grammatical approaches to engineering design. Part I: an introduction and commentary, *Research in Engineering Design*, **2**, 121-135

Oh,V. and Sharpe,J.(1995), Conflict management in an interdisciplinary design environment, , *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **9**, 247-258

Ohki,M.,Shinjo,H.,Oohira,E. and Abe,M. (1994), Design support to determine the range of design parameters by qualitative reasoning, *IEEE Transactions on Systems, Man, and Cybernetics*, 1994, **24**, 813-819

Otto,K.N. (1996), Forming product design specifications, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, August 18-22, 1996, Irvine, California, 96-DETC/DTM-1517

Otto,K.N. and Antonsson,E. K.(1991), Trade-Off Strategies in Engineering Design, *Research in Engineering Design*, **3**, 87-103

Owen,C.L. (1998), Design research: building knowledge base, *Design Studies*, Vol. 19, No.1, pp.9-20

Oxman,R.E. (1990), Prior knowledge in design: a dynamic knowledge based model of design and creativity, *Design Studies*, Vol. 11 No. 1, pp. 17~28

Oxman,R.E. and Oxman,R.M. (1992), Refinement and adaptation in design cognition, *Design Studies*, Vol.13, No.2, pp. 117~134

Oxman,R.E.(1994), Precedents in design: a computational model for the organization of precedent knowledge, *Design Studies*, Vol. 11 No. 1, pp. 141~157

Pahl, G., Beitz, W.(1988), Engineering design: A systematic approach, Spinger-Verlag.

Poincare,H.(1929), *The Foundations of Science*, The Science Press, New York

Popper,K.(1961), *The logic of scientific discovery*, Science Editions, New York

Rane,P and Issac,J.R. (1990), Functionality: the key concept towards integration of the product cycle, in *Proceedings of the 1991 ASME International Computers in Engineering Conference and Exhibition*, 1, pp.317-326

Rao,A.V. and Prakasa,N.S.(1992), BEAS: expert system for the preliminary design of bearings, *Advances in Engineering Software*, **14**, 163-166

Reddy,S.Y. (1996), HIDER :a methodology for early-stage exploration of design space, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, August 18-22, 1996, Irvine, California, 98-DETC/DAC-1089

Reddy,S.Y, Fertig, K.W. and Swith D.(1996), Constraint management methodology for conceptual design tradeoff studies, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, August 18-22, 1996, Irvine, California, 96-DETC/DTM-1228

Rinderle,J.R. (1990), Grammatical approaches to engineering design. Part II: melding configuration and parametric design using attribute grammars, *Research in Engineering Design*, **2**, 137-146

Roozenburg,N.F.M.(1993), On the pattern of reasoning in innovative design, *Design Studies*, Vol. 14, No.1, pp4-18.

Rosenman,M.A. and Gero,J.S. (1994), The what how, and the why in design, *Applied Artificial Intelligence*, **8**, 199-218

Rudolph,S. (1995), Computer Support in Design, in Gero,J. and Sudweeks,F.(ed.), *Proceedings of the IFIP WG5.2 Workshop on Formal Design Methods for CAD*, pp.231-240, Chapman and Hall

Rudolph,S.(1996a), On a mathematical foundation of axiomatic design, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, August 18-22, 1996, Irvine, California, 96-DETC/DTM-1530

Rudolph,S. (1996b), Upper and lower limits for "The PRINCIPLES OF DESIGN", Research in Engineering Design, **8**, 207-218

Sainter,P.,Oldham,K.,Kneebone,S.(1998), The necessity for product knowledge reuse and sharing within knowledge_based engineering systems, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 13-16, 1998, Atlanta, GA, DETC98/DAC-5569

Salustri,F.A. (1998), Function modeling for an integrated framework: a progress report, in D.Cook(ed.), *Proceedings of FLAIR '98*, AAAI, 339-343

Salustri,F.A. and Venter,R.D.(1992), An axiomatic theory of engineering design information, *Engineering with Computers,* **8**(4), 197-211

Salustri,F.A. and Venter,R.D.(1994), A new programming paradigm for engineering design software, *Engineering with Computers,* **10**(2), 95-111

Schmidt,L.C. and Cagan,J.(1996), Configuration design: an integrated approach using grammars, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, August 18-22, 1996, Irvine, California, 96-DETC/DTM-1511

Segapeli,J.L. and Cavarero,A. (1996), An artificial intelligence based algorithm for object-oriented design, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, August 18-22, 1996, Irvine, California, 96-DETC/DAC-1429

Sekiya,T.,Tsumaya,A. and Tomiyama,T. (1998), Classification of knowledge for generating engineering models-a case study of model generation in finite element analysis, in S.Finger, T.Tomiyama, and M.Mantyla(eds.), Preprints of the Third IFIP Workshop Group 5.2 Workshop on Knowledge Intensive CAD, pp.52-67, December 1-4, 1998,Tokyo

Shah,J.J. (1998), Experimental investigation of progressive idea generation techniques in engineering design, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 13-16, 1998, Atlanta, GA, DETC98/DTM-5676

Shah, J. and Hazelrigg, G.(1996), *Research opportunities in engineering design*, NSF Strategic planning workshop report

Sharpe,J.E. (1995), Computer tools for integrated conceptual design, *Design Studies*, **16**, 471-488

Shashkin,Y.A. (1991), *Fixed Points*, American Mathematical Society

Shiomomura, Y., Tanigawa, S., Takeda, H., Umeda, Y. and Tomiyama, T. (1996), Functional evaluation based on function content, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, August 18-22, 1996, Irvine, California, 96-DETC/DTM-1532

Shiomomura, Y., Yoshioka, M., Takeda, H., Umeda, Y. and Tomiyama, T.(1998), Representation of design object based on the functional evolution process model, *Transactions of ASME, Journal of Mechanical Design*, 120(2), 221-229

Simon,H.A.(1969), *The Sciences of the Artificial*, MIT Press.

Simon,H.A.(1971), Style in design, in Archea, C. Eastman (eds.) *Proceedings of 2nd Annual Environment Design Research Association Conference*, pp. 1~10, Oct. 1970, Carnegie-Mellon University, Pittsburgh, PA

Simon,H.A.(1973), The structure of ill structured problems, *Artificial Intelligence*, Vil. 4, No. 2, pp.181-201

Simon,H.A.(1982), Models of Bounded Rationality, Vol.2, Cambridge, MA, MIT Press

Simpson,T.W., Rosen,D. Allen,J.K., Mistree,F.(1996), Metrics for assessing design freedom and information certainty in the early stages of design, *Proceedings of DETC'96*, 96-DETC/DTM-1521.

Sodhi,R. and Turner,J.U., Towards modeling of assemblies for product design, *Computer-Aided Design*,

Stahovich,T.F.(1999), LEARNIT: a system that can learn and reuse design strategies, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 12-15, 1999, Las Vegas, Nevada, DETC99/DTM-8779

Stone,R.B., Wood,K.L. and Crawford,R.H.(1999), Product architecture development with quantitative functional models, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 12-15, 1999, Las Vegas, Nevada, DETC99/DTM-8764

Suh, N.(1990), *The Principles of Design*, Oxford University Press.

Sun,J., Kalenchuk,D.K., Xue,D., and Gu,P.(2000), Design candidate identification using neural network-based fuzzy reasoning, *Robotics and Computer Integrated Manufacturing*, **16**, 283-396

Suwa,M.,Purcell,T. and Gero,J.(1998), Macroscopic analysis of design processes based on a scheme for coding designers' cognitive actions, *Design Studies*, **19**, 455-483

Takeda(1994), Abduction for design, in J.S.Gero and E.Tyugu(eds.), *Formal Design Methods for CAD*, pp.221-244, North_Holland

Takeda,H.,Tsumaya,A. and Tomiyama,T.(1999), Synthesis thought processes in design, in H.Kals and F.van Houton (eds.), *Integration of process knowledge into Design Support Systems*, pp. 249-258, Kluwer Academic Publishers

Taguchi, G.(1987), "System of Experimental Design", Vol. 1 and 2, UN1PUB-Kraus International Publications (White Plains) and American Supplier Institute, Inc. (Dearborn)

Tichem,M.,Storm,T., Andreasen M.M., and MacCallum,K.J.(1997), Product structuring, an overview, *Proceedings of International Conference on Engineering Design'97*, pp.45-50

Tomiyama,T.(1994), From general design theory to knowledge-intensive engineering, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **8**, 319-333

Tomiyama,T. and Yoshikawa,H.(1985), Extended general design theory, *Design Theory for CAD*, Elsevier Science Publishers B. V., Amesterdam, pp.95-130.

Tribus,M. (1969), *Rational Descriptions, Decisions and Designs*, Pergamon Press

Ullman,D.G.(1992), *The Mechanical Design Process*, New York: McGraw-Hill

Umeda,Y.,Ishii,M.,Yoshioka,M., and Shimomura,Y., Supporting Conceptual Design based on the Function-Behavior-State Modeler

Umeda,Y., Takeda,H., Tomiyama,T., Yoshikawa,H.(1990), Function, behavior, and structure, in J.S.Gero(ed.), *Applications of Artificial Intelligence in Engineering V*, Vol. 1, Berlin: Springer Verlag, pp.177-194

Vescovi,M., Iwasaki,Y.,Fikes,R. and Chandrasekaran,B. (1993), CFRL: a language for specifying the causal functionality of engineered devices, in *AAAI-93*

Wan,J., and Krishnamurty,S. (1999), Comparison-based decision making in engineering design, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 12-15, 1999, Las Vegas, Nevada, DETC99/DTM-8746

Warell,A.V. (1999), Introducing a use perspective in product design theory and methodology, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 12-15, 1999, Las Vegas, Nevada, DETC99/DTM8782

Whitney,D.E.,Dong,Q.,Judson,J. and Mascoli, G. (1999), Introducing knowledge-based engineering into an interconnected product development process, *Proceedings of The ASME*

Design Engineering Technical Conference and Computers in Engineering Conference, September 12-15, 1999, Las Vegas, Nevada, DETC99/DTM-8741

Wiegers,T.,Breemen,E.,Knoop,W. and Vergeest,J.(1996), Required and requested information during conceptual engineering design as found by categorization of empirical data, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, August 18-22, 1996, Irvine, California, 96-DETC/DTM-1309

Willem,R.A.(1990) , Design and science, *Design Studies*, **11**(3): 137-141

Xue,D., Yadav,S., Norrie,D.H. (1999), Knowledge base and database representation for intelligent concurrent design, *Computer-Aided Design*, **31**, 131-145

Yoshikawa,H.(1981), General design theory and a CAD system, in *Man-machine Communications in CAD/CAM*, Tokyo, Oct. 2-4, 1980, Proceedings of IFIP WG5.2, North_holland, Amsterdam, pp.35-38

Yoshioka,M. and Tomiyama,T. (1999), Toward a reasoning framework of design as synthesis, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 12-15, 1999, Las Vegas, Nevada, DETC99/DTM-8743

Yu,H.,Shu,L. and Venter,R. (1998), An enhanced axiomatic design process, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 13-16, 1998, Atlanta, GA, DETC98/DTM-5653

Zamirowski,E. and Otto,K.(1999), Identifying product portfolio architecture modularity using function and variety heuristics, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 12-15, 1999, Las Vegas, Nevada, DETC99/DTM-8760

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

Zeng,Y. and Gu,P.(1999a), 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. and Gu,P.(1999b), A science-based approach to product design theory. Part II: formulation of design requirements and products, *Robotics and Computer Integrated Manufacturing*, Vol.15, No.4, pp.341-352

Zhang,F. and Xue,D. (2001), Optimal concurrent design based upon distributed database and knowledge base modeling, *Proceedings of the ASME Design Engineering Technical Conferneces, September 9-12, 2001, Pittsburgh, Pennsylvania,* DETC2001/DAC

Zhu,L. and Kazmer,D. (1999), A performance-based representation for engineering design, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, September 12-15, 1999, Las Vegas, Nevada, DETC99/DTM-8748