A Systematic Study of Design Conflict:
Modeling, Representation, Resolution, and Application

Baiquan Yan

A Thesis

in

the Department of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy (Mechanical Engineering) at

Concordia University

Montreal, Quebec, Canada

October 2013

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Baiquan Yan**

Entitled: **A Systematic Study on Design Conflict: Modeling, Representation, Resolution, and Applications**

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Mechanical Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________ Chair

Dr. H. Akbari

______________________________ External Examiner

Dr. W.J. Zhang

______________________________ External to Program

Dr. Z. Chen

______________________________ Examiner

Dr. A. Akgunduz

______________________________ Examiner

Dr. Z. Chen

______________________________ Thesis Supervisor

Dr. Y. Zeng

Approved by ______________________________

Dr. A. Dolatabadi, Graduate Program Director

October 2, 2013 ______________________________

Dr. C. Trueman, Interim Dean
Faculty of Engineering & Computer Science

ABSTRACT

A Systematic Study of Design Conflict: Modeling, Representation, Resolution, and Application

Baiquan Yan, Ph.D.
Concordia University, 2013

Conflicts drive the development of technical systems and the evolution of design process. Conflict management, which mainly includes conflict identification and resolution, is a crucial part of design activity. This research conducts a systematic study and proposes a formal structure of design conflicts.

The first step of conflict management is to build up a formal model for technical system. Currently, there exist some inconsistences among different design theories because of the lack of a cohesive set of fundamental concepts about technical systems. This lack also causes misunderstanding among researchers and therefore hinders the development of design theories. This thesis presents a formal approach to representing technical systems. Both theoretical derivation and extensive example have shown that this formal representation meets the five requirements: completeness, clarity, independence, flexibility, and adaptability. A set five concepts— purpose, function, structure, behaviour and state— is identified and formally defined as the base set for technical systems.

The second step is to model conflicts based on the formalization of technical system. Current studies are based on heuristics and lack a systematic approach, and therefore fail to detect conflicts that are not predefined. This research puts forward a formal structure of design conflicts based on systematic analysis. This formal structure shows that any conflict is composed of at least three objects: two competing objects and one resource object that the former two contend for. This formal structure can be applied to different design fields and helps designers identify all conflicts existed in different design stages.

Based on the formal structure of conflicts and analysis of relation among the three objects in a conflict, this research also proposes three formal methods for detecting conflicts and presents a set of general resolution principles, which include modifying resource object, separating conflict relations in time or in space, changing the two competing objects, using optimization methods, and replacing the whole conflict.

An example demonstrates the application of the formal structures, followed with conclusion and suggestions for future research.

Thesis supervisor: Dr. Yong Zeng

Acknowledgements

It has been a long journey since I started research on design theory. This journey is filled with the pain of failures and the joy of successes. Looking back, I realize that I could not have finished my thesis if I had not gained support from so many brilliant professors, friends, and research fellows, to whom I could not adequately express my deep gratitude.

First of all, I would like to thank my supervisor Dr. Yong Zeng for everything he has done for me these years. He first introduced design theory to me about ten years ago. Having worked as an engineer, I was very attracted by the philosophical depth of general design theory, to which I have been devoting for so many years. In addition to the invaluable academic advices professor Zeng has given me, his other personalities have impressed me: his sincere appreciation of peers' work, his enduring passion for research, and his humble way of talking to his students. From him, I have not only learned the correct way of doing research, but also the way of being a person. I am grateful for having such an unforgettable experience of studying with professor Zeng.

Second, I would like to thank those who have worked with me in the same group. I thank them for their selfless help and kind encouragement. I cannot list all their names here, but I would like to say 'thank you' to Bo Chen, Zhenyu Chen, Shuren Li, Ann Nguyen, Min Wang, Xuan Sun, and Dr. Xiaoguang Deng. They have taught me software, proofread my writings, and helped me find papers. I will cherish those memories for my whole life.

Lastly, I would like to thank my family: my hard-working wife, who has always been standing behind me for more than twenty years; my self-reliant son, who helps me by always doing his school work himself without bothering me; my parents, who have never

had chance to go school, but give me the ability to do scientific research. To them I dedicate this thesis.

Table of Contents

List of Figures……ix
List of Tables……xi

Chapter 1 Introduction 1
1.1 Motivation........ 1
1.2 Research methodology........ 2
1.3 Objectives and structure of the thesis 5

Chapter 2 Literature Review........ 6
2.1 Introduction........ 6
2.2 Development of design theories 6
2.3 Studies on design conflicts 7
2.4 Representations of design 13
2.4.1 Textual representations 15
2.4.2 Mathematical representations........16
2.4.3 Physical representations........16
2.4.4 Graphic representations........16
2.5 Theoretical foundations 25
2.5.1 Axiomatic theory of design modeling (ATDM) 25
2.5.2 Recursive object model (ROM)........27

Chapter 3 Formal Representation of Technical Systems........ 28
3.1 Introduction........ 28
3.2 Analyse of current theories on technical systems........ 33
3.2.1 A review of current research 33
3.2.2 Analysis 37
3.3 Formal representation of technical systems........ 42
3.3.1 Requirements for a good representation........42
3.3.2 Graphic representation 43
3.3.3 Set-theoretic representation 46
3.3.4 Description of technical systems........47
3.3.5 Environment of a technical system........51
3.3.6 Completeness of the formal representation 51
3.4 Design example of application of formal representation........ 55
3.4.1 Introduction: A brief review of some fundamental concepts........55
3.4.2 Description of the design problem........56
3.4.3 Description of design process using Formal representation........57
3.5 A comparison between the three representations 60
3.6 Relations 61
3.6.1 Scope of technical system 61
3.6.2 Classification of technical systems........63
3.6.3 Relations in a technical system........64
3.7 Summary........ 68

Chapter 4 Structure of Design Conflict 70
4.1 Introduction........ 70

4.2 ROM analysis of definitions of conflict: conceptual structure ... 71
4.3 Formal structure of design conflict ... 76
4.3.1 Introduction ... 76
4.3.2 A formal representation ... 77
4.4 Examples of conflicts ... 79
4.5 Types of conflicts ... 85
4.6 Summary ... 88

Chapter 5 Conflict Resolution Strategies ... 95
5.1 General analysis of the structure of conflicts ... 95
5.2 Qualitative analysis of relations in a conflict ... 97
5.3 Resolution principles of conflicts ... 99
5.3.1 Modify c_r or replace c_r with a new source c_{rn} ... 100
5.3.2 Separate $(c_1 \otimes c_r)$ and $(c_2 \otimes c_r)$ in time ... 101
5.3.3 Separate $(c_1 \otimes c_r)$ and $(c_2 \otimes c_r)$ in space ... 101
5.3.4 Change c_1 or c_2 ... 101
5.3.5 Use optimization method ... 101
5.3.6 Replace the whole conflict ... 102
5.4 Principal conflict and the principal aspect of a conflict ... 102
5.5 Discussion of the criterion for selecting resolution strategy ... 105

Chapter 6 Example: Design Evolution of Brake System ... 107
6.1 A brief review of the development of vehicle brake system ... 107
6.1.1 Early mechanical brake system ... 108
6.1.2 Drum brake system ... 108
6.1.3 Hydraulic and power-assistant brake system ... 109
6.1.4 Disc brake system ... 110
6.1.5 Antilock and dynamics control brake system ... 110
6.2 Analysis of the design revolution of brake system ... 111
6.2.1 Environment analysis for designing brake ... 111
6.2.2 Conflict identification ... 116
6.2.3 Analysis of the design revolution of different brake systems ... 119
6.3 Summary ... 128

Chapter 7 Conclusions and Future Directions ... 130
7.1 Technical systems ... 130
7.2 Design conflicts ... 132
7.3 Suggestions for future research ... 133

References… ... 134

Appendix… ... 144
A1. A New Method to Solve Design Governing Equation ... 144
A2. More ROM Analysis of Definitions of Conflicts ... 152
A3. A list of publications ... 156

List of Figures

Figure 1-1 Two main strategies of design research (Zeng, 2002) 3
Figure 2-1 Three models of representation of systems (Hazelrigg, 1996) 14
Figure 2-2 Four components of data flow diagram 18
Figure 2-3 Seabed Species and Habitats (UK Marine Data Archive Centre , 2013).........19
Figure 2-4 Pixton Book Inventory and Patron Check-Out System...................................20
Figure 2-5 UML diagrams 22
Figure 2-6 A class diagram - software of mesh generation (Yan & Chen, 2004) 23
Figure 2-7 Use case diagram – mesh generation 23
Figure 2-8 SysML structure (American Systems Corporation, 2006)..............................25
Figure 3-1 A tree representation of a technical system..44
Figure 3-2 Car system in tree form 45
Figure 3-3 Shaft-coupling system..49
Figure 3-4 Technical system and its environment 51
Figure 3-5 Evolution of product in the design process (Zeng, 2002) 55
Figure 3-6 Evolution of requirements in the design process 56
Figure 3-7 Evolution of the design process: refined. 56
Figure 3-8 Schematic of a hydraulic system 57
Figure 3-9 Primitive products 58
Figure 3-10 Partial system 1 $S = S_0 \cup s_1$ 58
Figure 3-11 Partial system 2, $S = S_0 \cup s_1 \cup s_2$ 59
Figure 3-12 Partial system 3 $S = S_0 \cup s_1 \cup s_2 \cup s_3$ 59
Figure 4-1 ROM diagram of three types of Conflicts...74
Figure 4-2 Simplified ROM diagrams of three types of Conflicts 76
Figure 4-3 Example of conflict 77
Figure 4-4 Fundamental structure of atomic conflict 78
Figure 4-5 Three types of relations 79
Figure 4-6 Simplified ROM diagram for the semiconductor design example..................80
Figure 4-7 Example to show conflict..81
Figure 4-8 Graphical problem description 81
Figure 4-9 Conflict of the design problem..82
Figure 4-10 Group conflict 1 83
Figure 4-11 Group conflict 2 83
Figure 4-12 A compound group conflict 83
Figure 4-13 A compound conflict in shaft design 84
Figure 4-14 Decomposition of a compound conflict 84
Figure 4-15 A lathe cutter and rotating work..85
Figure 4-16 Description of complex conflict..85
Figure 4-17 A compound conflict..87
Figure 4-18 A complex conflict 87
Figure 5-1 Structure of a conflict...97
Figure 5-2 Complex conflict 102
Figure 6-1 ROM description of brake design problem 112
Figure 6-2 A typical mechanical brake system (Hasegawa & Uchida, 1999) 119

Figure 6-3 External band brake (Carley & Mavrigian, 1998) 121
Figure 6-4 Structure of conflict..122
Figure 6-5 Mechanism of linkage 124
Figure 6-6 New conflict in external drum brake...126
Figure 6-7 A conflict in current vehicle brake design 128

List of Tables

Table 2-1 Symbols used in ROM analysis..27
Table 3-1 Summary of concepts defined 38
Table 3-2 Definitions of function...39
Table 3-3 Definitions of behaviours 39
Table 4-1 Prisoner's Dilemma game 92
Table 6-1 Environment elements..115
Table 6-2 Conflicts identified...118
Table 6-3 Symbols used in the design 122

Chapter 1 Introduction

1.1 Motivation

Design conflict exists in all human design activities. It drives the evolution of technical system and the development of design process. Whenever human beings have some new needs that the environment cannot afford, there will be a conflict, which can only be resolved through design activities. And every time after a designer resolves a design conflict, the product will evolve to a new state or new stage. The reason that conflicts exist is that all resources we need to satisfy our needs are limited. In the past, human beings optimistically considered the resources on this earth infinite; but only in less than two centuries after industrial revolution, some resources have been used up because of our ignorant abuse. This situation reveals an important role of design conflict management: to use the resources in optimal ways. Moreover, as modern commercial product becomes more complex and multi-disciplinary, it demands more resources including human resources, natural resources, and financial resources. Though there is no single absolute answer to design a high quality product; there are some tested design principles that, if followed, may increase the likelihood of success of any final product. Among those principles, conflict management is one of the most essential that all design theory should address. Having realized the importance of conflict in design activity, many researchers have been devoting their time and energy to the strategies of conflict detection and resolution. We can classify current research works into two categories. The first category focuses on the conflict among people in the context of collaborative design, concurrent engineering or cooperative design. The second category mainly focuses on the engineering or technological aspects of the product to be designed, including conflicts in the problem specifications and conflicts generated in the design process. However, a comprehensive analysis of current research status reveals some problems. First, except for TRIZ all other studies on conflicts belong to the first category, indicating a lack of conflict studies in the engineering or technological aspects. Second, the existent models of conflict are obtained by mainly using heuristic methods and only aim at computational application. Third, there are no formal representation that can describe general conflicts independent of any design fields; even in TRIZ, "the set of contradictions (conflicts) proposed by Altshuller is not exhaustive for various problems outside engineering (Savransky, 2000).

As already pointed out in the previous part, conflict is such an important concept in design theory that it requests a systematic study, for only though systematic investigation can a formal structure of conflict be found. It is expected that such a formal form shall be complete and consistent in describing design conflicts and be able to serve as a powerful tool in conflict detection and resolution.

1.2 Research methodology

As summarized in (Zeng, 2002), the methods used in design research are classified into three categories: philosophical investigation, deductive reasoning and inductive generalization. Philosophical studies investigate design theory through retrospection and speculation, which have enriched our understanding of design research and provided us a macro-perspective to study design (Zeng & Yao, 2009). Yoshikawa (1989) has indicated that the philosophy of design represents the highest level of speculative thinking about the experience of design activities, the role and position of design in the society, the historical evolution of the design discipline. Horváth (2004) has considered philosophy of design as a meta-theoretical framework for design theories by which epistemological and ontological clarity may be brought in. In addition to the philosophical approach to design research, a design theory can also be developed by using other two systematic approaches: deductive and inductive studies, as is shown in Figure 1-1.

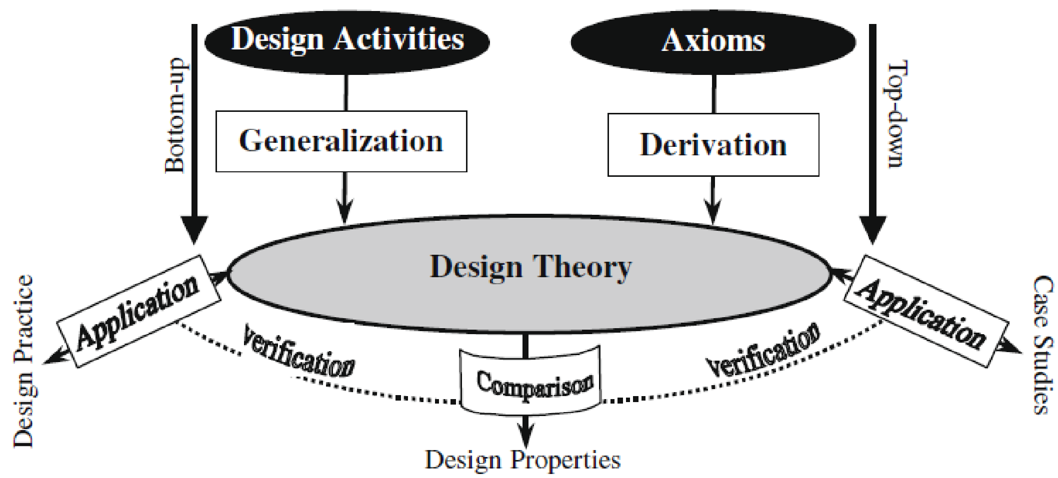


Figure 1-1 Two main strategies of design research (Zeng, 2002)

Deductive studies attempt to derive design knowledge from axioms; inductive studies aim to generalize design theory from observations on design activities. The former approach is a top–down strategy; the latter is a bottom–up strategy (Zeng, 2002). No matter what way we use to create design knowledge, a good design theory must reflect the nature and characteristics of the design process. Deductive studies aim to identify the common elements and disclose the underlying order of the design process through existing axioms. The success of understanding and modeling the design process depends on scientific exploration of design activities. Representative works include axiomatic design (Suh, 1990), General Design Theory (Yoshikawa, 1981), Formal Design Theory (Braha and Maimon, 1998) and Axiomatic Theory of Design Modeling (Zeng, 2002).

Inductive studies attempt to generalize design theories by observing designer's design activities. Commonly used methods include interviews with designers, case studies, protocol studies, simulation trials (Cross, Dorst, & Roozenburg, 1992). Having been used in investigating the process of designing and in understanding how designers design since the 1980's (Eckersley, 1988; Jin & Chusilp, 2006; Stauffer & Ullman, 1991), protocol analysis records designers' thinking and then studies their mental processes (as cited in Zeng & Yao, 2009). Case study is used by researchers to produce new design knowledge, to verify and validate an existing design theory, or to explain design phenomena. Computer simulation simulates real design situations by allowing designer to work with a computer-simulated environment. It is not only cost-effective, but also can be easily and quickly operated to present the real problem in many different perspectives in a virtual environment. Computer simulation involves designing a model of an actual or theoretical physical system, executing the model on computer, and analyzing the execution output (Fishwick, 1995). Any design theory can be verified by applying it to case studies, by comparing it to commonly accepted understanding of design properties, or by applying it to improve the design practices (Zeng, 2002).

The method used in this research is a combination of deductive and inductive strategies. In formalizing the representation of technical system, we start from the axioms proposed in Axiomatic Theory of Design Modelling (ATDM) and then derive a formal form of technical system and a set of fundamental concepts; in formalizing design conflicts, we start from different definitions and situations of conflicts and then identify the common elements using ROM analysis; and we validate the results using comprehensive cases.

1.3 Objectives and structure of the thesis

Having discussed the importance of conflict management in former sections, I set as the research goal to find a formal structure by conducting systematic investigation. This goal includes the following detailed objectives.

a) To formalize technical system representation by using a systematic approach
b) To systematically investigate the formal structure applied to all design conflicts
c) To propose a set of conflict resolution principles based on the formal structure
d) To demonstrate the applications of the formal structure using a comprehensive example.

The rest part of the thesis is composed of five chapters. In Chapter 2, we review current representative works on technical system and design conflicts; in Chapter 3, we present a formal representation of technical system(TS) and propose a set of fundamental concepts about TS; in Chapter 4, we present formal structure of design conflicts by suing ROM analysis; in Chapter 5, we investigate resolution strategies based on the formal structure; in Chapter 6, we apply our theory to a comprehensive case study; in Chapter 7, we summarize the work and propose future research.

Chapter 2 Literature Review

2.1 Introduction

This review consists of four parts. First, we present a broad picture about the current status of design theories; second, we present a details survey of design conflict and pinpoint the existing problems; third, we investigate the strategies of technical system representation; last, we introduce the theoretical foundation of this research.

2.2 Development of design theories

Comparing to the over four-thousand-year history of design practices, the study in design theory as a scientific discipline is quite young. However, with the fast development of technology and society, more and more natural resources are consumed; people begin to realize the importance of design theory. For only through good design can we use the resource in an optimized way and a good design cannot be achieved unless the designers are armed with both profound tactic design knowledge and sound design theory. Having realized this, many researchers, including scientist, professors and engineers, have put great effort to investigate the nature of design activities.

Design research as a scientific discipline started in 1960s. The early researches focused on the nature of design process and new design methods that could rejuvenate traditional engineering design practice. Though definitions about design given by different researchers are different from each other, the research results demonstrate some common characteristics about design process: design is a scientific human activity dealing with complexity and uncertainty (Jones, 1966); designers make extensive use of scientific principles, technical information and imagination to achieve a pre-specified goal (Eder, 1966; Broadbent, 1966). It also shows that design is not only pure technical but also related to the social environment as well (Simon, 1996). The second trend of design research has focused on the formulation of design problem. It is observed that design problems could not be described with well-defined structure. On the contrary, design problems are ill-structured or wicked-problem, as some researchers put it (Simon, 1996; Rittel & Webber, 1973, Cross, 1989). In addition, there is no unique solution for a design problem, which renders the modelling of design problems more complicated and uncertain. As computer is widely used by designers, some researchers have begun to think design as an information-processing process (Yoshikawa, 1985; Hubka, 1985; Hongo, 1985; Eder, 1989). In the meantime, systematic view, decision-making theory, and mathematical method are also applied by many researchers in proposing design theories, for example, systematic design theory (Hubka & Eder, 1988; Pugh, 1989; Pahl and Beitz, 1989), Theory of Inventive Problem-Solving (TRIZ) (Altshuller, 1984); decision-based design theory (Hazelrigg, 1996), computational design theory (Gero & Maher, 1997.), axiomatic design (Suh, 1990), mathematics-based General Design Theory (Yoshikawa, 1981), Extended General Design Theory (Tomiyama & Yoshikawa, 1987), Formal Design Theory (Braha & Maimon, 1998), Axiomatic Theory of Design Modeling (Zeng, 2002), science-based design theory (Zeng & Gu, 1999a, 1999b), and C-K theory (Hatchuel & Weil, 2003).

2.3 Studies on design conflicts

Conflicts are ubiquitous. They can be found between individual human beings, between human beings and their living environment, between different organizations, or between different countries, etc. Generally, any two things, if they have some common need for something else, may conflict each other. Obviously, different conflicts are resolved in different ways, depending on the conflict themselves, i.e. the nature and complexity of the conflicts, the field to which the conflict belongs, and, most importantly, the effects of and resource constraints for resolving the conflicts. While some conflicts can be easily resolved; others may be quite challenging. Research on conflict management has been conducted in many domains: in social science, in psychology, and in politics, as well as in technical fields such as equipment maintenances, diagnosis, manufacturing, artificial intelligence, and design.

Studies of conflicts in design domain are important for two reasons. First, engineering design is a very complex and multidisciplinary activity that requires for collaboration of people from different disciplines; and conflict resolution is a key factor for achieving successful designs. Secondly, the product to be designed becomes more and more complex both in structural and in functional requirements; therefore, conflicts between functions and structures may occur during the design process. Accordingly, there exist two major streams in the research of conflict. The first stream deals with the relations among different designers in the context of collaborative design and concurrent engineering. The second stream deals with the engineering or technological aspects of the product to be designed. For convenience of discussion, we name the conflicts studied in the first stream as cognitive conflicts, and those in the second stream as technological conflicts.

Cognitive conflicts have been intensively studied by many researchers in the context of concurrent engineering (Ku & DeMicheli, 1991), collaborative or cooperative design

(Klein & Lu, 1989; Lara, 1999; Lara & Nof, 2003; Sreeram, 2000; Fernandez, 2005; Yesilbas & Lombard, 2004; Wang, Shen, Xie, Neelamkavil, & Pardasani, 2002; Taratoukhine, 2002; Lottaz, Smith, Robert-Nicoud, & Faltings, 2000; Gorrti, Gupta, Kim, Sriram, & and Wong, 1998), or distributive design (José & Velasquez, 2003; Ross, Fang, & Hipel, 2002; Daniela, Damian, Jonker, Treur, & Wijngaards, 2005). A common characteristic of all those studies is that they handle conflicts occurring among people or teams of people when they have different opinions about one issue, either engineering or non-engineering. Two major approaches are used in the research: works on conflict resolutions in human setting and on computational models of conflict resolution (Klein, 1990).

Technological conflicts are conflicts between two or more functional or structural requirements of a product. Kumar and Gero (1993) have identified two types of technological conflicts: conflicts in the problem specifications and conflicts generated in the design process. In TRIZ literature (Altshuller, 1984; Savransky, 2000), conflicts are also called contradiction, which are classified into three groups: administrative conflicts, technical conflicts, and physical conflicts.

In spite of the variety of conflicts, we will ask a question: do different conflicts have a common structure? The significance of this question is that if there is a common structure, then we can find some general principles for resolving various conflicts, based on which tools to aid conflict resolution can be developed. The tools can not only improve designer's ability in designing new products but also lay a foundation to computer-aided conceptual design. In detail, a formal structure of conflicts should be able to achieve the following three objectives.

(1) To help identify conflicts,

(2) To imply the direction of conflict resolution, and

(3) To facilitate computer-aided problem solving.

First of all, a formal representation of conflict structure helps identify conflicts. Conflict detection is the first step of conflict management. The existing research largely depends on heuristic methods in detecting conflicts. Klein (1993) has introduced two detecting ways in developing DCSS (the Design Collaboration Support System), namely, least-commitment model and a range of tools by using conflict detection idioms, which is a small set of stereotyped conflict detection heuristics. Also because of the lack of formal representation, conflicts can only be resolved manually by people using tentative rules such as design reviews, change memos, and mock-ups (Klein, 2000). Other methods include identifying conflicts with the help of the classification of conflict types (Hanna & Barber, 2001) and case-based reasoning (Ross, Fang, & Hipel, 2002). A common characteristic of the existing methods of conflict detection is that pre-defined conflicts are required. A major advantage of these methods is the high efficiency of detection if a conflict similar to the problem under consideration is already defined. However, if a new type of conflicts emerges that are different from the predefined conflict types, those methods may fail to identify them. In addition, the existing methods cannot support automatic conflict identification.

Second, a formal representation of conflict structure implies the direction of conflict resolution. The aim of conflict management is to resolve the conflict. Ideally, a representation not only shows the structure of a conflict but also reveals the relations between the components of this conflict. Those relations can be used to trace the root of the conflict. Usually, a conflict can be resolved in many ways; which solution is the best depends on the relations between components.

Third, a formal representation can be used to build up computer-aided problem solving systems. As the design activities become more and more complicated and the globalization of product design prospers, web-based collaborative design begins to play important role in the development of new products. A formal structure of conflicts can be integrated with a design system to automatically detect and identify run-time conflicts. As a result, a computer-aided conflict resolution system can be implemented, which will greatly facilitate the cooperation among different designers.

As already discussed in last chapter, the objective of design activity is to create a new product or technical system; an instant question is how we describe this new system so that all participants can understand the design purpose and therefore communicate with each other. In other words, there would be no design activity without representation. Ideas must be represented if they are to be shared with others, or to be used later by oneself. Different representational modes and strategies afford distinctive opportunities for reading or for transforming design ideas. In detail, the roles of design representations include recording the design results, aiding designer's thinking, helping designers to communicate with each other, and providing a guide for follow-up activities such as manufacturing. Firstly, because most design problems are complex, they are often achieved in several steps. In each step, some intermediate results may be generated. Therefore, a proper representation is needed for each step. Secondly, a good representation can aid designers' reasoning and help them find good solutions. Research in cognitive field has shown that some representations can make a problem easy to handle while other representations for the same problem may make it difficult to understand (Zhang, 1997a). It shows that attaining insight requires effective problem representations (Kaplan & Simon, 1990). In particular, as Simon pointed out, representation of design problems holds the key to their solutions and the complexity of design can be tamed by effective representations (Simon 1996). The third purpose of using representation in design problems is the need for communication among different designers. For example, in collaborative design, some designers work together to solve a design problem. They need to share results, ideals, and knowledge, in which case a good representation that is clear to everyone can facilitate the communication and therefore improve the work efficiency. Lastly, the ultimate goal of designing a product is to manufacture it and put it into use. A clear and correct representation is a prerequisite for manufacturing. In addition, since a design may need improvement in the future, the representation serves as a record for future modification. There may be others reasons, but those factors discussed above are the major reasons why we need representations in solving design problems.

Representations are various in design field. They can be classified in terms of many criteria. From cognitive point of view, we can classify representations into two large groups: internal representations and external representations (Zhang 1997b). Internal representations are the knowledge and structure in memory, such as one-digit numbers, basic geometry forms, etc. External representations are the knowledge and structure in the environment, such as symbols, objects, words in written form etc. We can create external representations because we have internal representations in memory. External representations can be transformed into internal representations by repeat practicing and memorizing. For example, most mechanical engineering students have internal representations of a nut, a simple shaft, or a spring. But they may not have internal representations of more complex components such as a gear-box, which expert designers may have because they have practiced designing those components for many years. Another classification of representations is according to the fields they are applied to. For examples, we can identify the representations used in mechanical engineering, in electric engineering, in civil engineering, etc. We can also classify representations according to their abstract level or granularity. Representations with lower granularity are usually used in early design stages, especially in conceptual design stage. They do not include detailed information about the product to be designed. On the contrary, representations with higher granularity contain more detailed information about a product. We can also divide representations into informal and formal representations. A formal representation is more accurate and can be uniquely interpreted, while an informal representation must be understood in the context to be discussed. Human designer can use both informal and formal representations, but a machine can only understand formal representations. This research focuses on external and formal representations that may apply to different design fields.

2.4 Representations of design

Design is representation, says Simon (1996). Here, Simon not only points out the importance of representation but also implies that representation happens when design activity happens. It is safe to say that representation of design has the same history as that of design itself. As human's understanding of design evolves, the representation used in design also evolves. Since different representations are needed in different stages of product design and representations used for detailed design and manufacturing are domain dependent, this research focuses on the representations of structure and function for conceptual design stage, during which structure description and function modeling are the most important aspects of representation. In describing a device, Middendorf (1997) has classified existing models of representation into the following categories: sketches and drawings, network model, mathematical models, physical models, combinations of models, and use of dimensional analysis. Simon (1996) has grouped representation into the principal categories of analog and symbolic representation. Kossiakoff *et al* (Kossiakoff, Sweet, Seymour, & Stev, 2011) have grouped the models of representation strategies into three categories: schematic model, mathematical model, and physical model. Hazelrigg (1996) has identified three model, iconic model, analog model, and symbolic model, as shown in Figure 2-1.

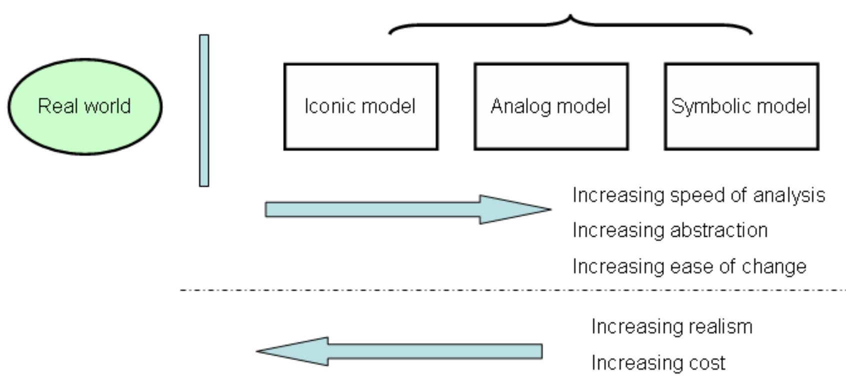


Figure 2-1 Three models of representation of systems (Hazelrigg, 1996)

Though different names are used, all these representations can be classified into four categories: text representations, graphic representations, mathematical representations, and physical representations. Text representations use natural languages. Graphic representations include diagrams or charts representing a system or process. Mathematical models use notations or symbols to represent the elements and their relationships of a system. Physical models are physical representations of a system, such as a globe model of the Earth.

2.4.1 Textual representations

Text representation represents models using natural languages, whose primitives obviously are all words. In design, text representation is indispensable in modeling design requirement and functional analysis. As the design practice has shown, most design requirements of products are written in textual format. Text representation models functions in such a way that a verb is used to describe what a product dose or is supposed to do (Freeman and Newell 1971; Jonson 1988; Lai and Wilson 1989). For example, the function of a memory card is to temporally save data. Text representation in natural language such as in English often appears ambiguous and therefore cause misunderstanding between people who use the representation. Many researchers have proposed special languages for describing functions in order to reduce the confusion resulted from using natural language. Stone and Wood (2000) develop a functional modeling language for human analysis and communication. They call this language as a functional basis, where product function is characterized in a verb-object (function-flow) format. Kuehne (2004) also investigates how to represent physical quantities using natural languages. To this end, he identifies five constituents of physical quantities: entity, quantity types, value, unit, and sign of the derivative. Text representations are also combined with other representations, for example, in mechanical design, a machine part is described in a drawing with notes written in natural language.

While textual representation can be applied to describing technical processes, it is difficult, if not impossible, to describe the structure of technical systems. Even used in technical processes, textual representations are more powerful when combining with diagrams.

2.4.2 Mathematical representations

Mathematical representations are used to express system functionality and dependencies in the languages of mathematics. They include approximate calculations, elementary relationships, differential equations, and statistical distributions. Mathematical representations also use the concept of set to describe a system; for example, Yoshikawa's General Design Theory (Yoshikawa, 1981), Braha and Maimon's Formal Design Theory (Braha & Maimon, 1998), and Salustri and Venter's Axiomatic Theory of Engineering Design Information (Salustri & Venter, 1992), Zeng's Axiomatic Design Theory (Zeng , 2001).

2.4.3 Physical representations

Physical representations directly reflect some or most of the physical characteristics of an actual system. Commonly used includes scale models, mock-ups, and prototypes. Physical representations are usually used after the detail structure feature has been decided and before manufacturing.

2.4.4 Graphic representations

A diagram is a simplified and structured visual representation of concepts, ideas, constructions, relations, statistical data, anatomy, etc. used in all aspects of human activities to visualize and clarify the topic. Graphic representations are used to convey relationships in a diagrammatical form by using commonly understood symbols. Widely used methods include system block diagrams, system context diagrams, functional flow block diagrams, data flow diagrams, UML, and sketch that are analogous to the product being designed. What graphic representation is used depends on the nature of the product.

BLOCK DIAGRAMS

A block diagram is a pictorial representation of some process or model of a complex system. The primitives of block diagrams are geometric shapes, such as squares, rectangles, circles, arrows, and lines. The squares or rectangles are connected by lines to indicate association and direction/order of traversal. When block diagram is used to describe the structure of a system, it is called system block diagram, which is often used to simplify complex systems and show blocks of functionality.

Block diagram are also often used to model a process to enable or indicate functional isolation. Typical example of usage is in software design and process flow diagrams. A block diagram is a useful tool both in designing new processes and in improving existing processes. In both cases the block diagram provides a quick, high-level view of the work and may rapidly lead to process points of interest. Because of its high-level perspective, it may not offer the level of detail required for more comprehensive planning or analysis. Team members who construct a block diagram must have a clear understanding of how the process operates.

DATA FLOW DIAGRAMS

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system (Ibrahim & Siow, 2011). A data flow diagram can also be used for the visualization of data processing. It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then "exploded" to show more detail of the system being modeled.

A typical data flow diagram is composed of four elements: process, dada store, external entity, and data flow, as shown in Figure 2-2. External entity is outside of the system being modeled, and represents where information comes from or where it goes. Processes modify the inputs in the process of generating the outputs. Data Stores represent a place in the process where data comes to rest. Data Flows are how data moves between terminators, processes, and data stores. An example of data flow diagram is shown in Figure 2-3.

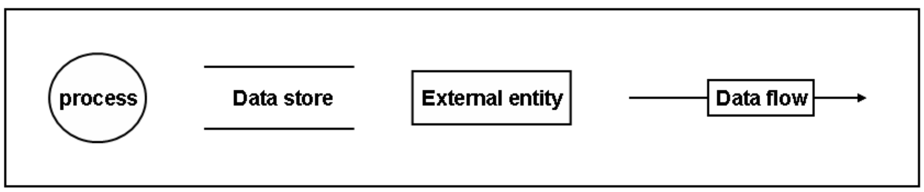


Figure 2-2 Four components of data flow diagram

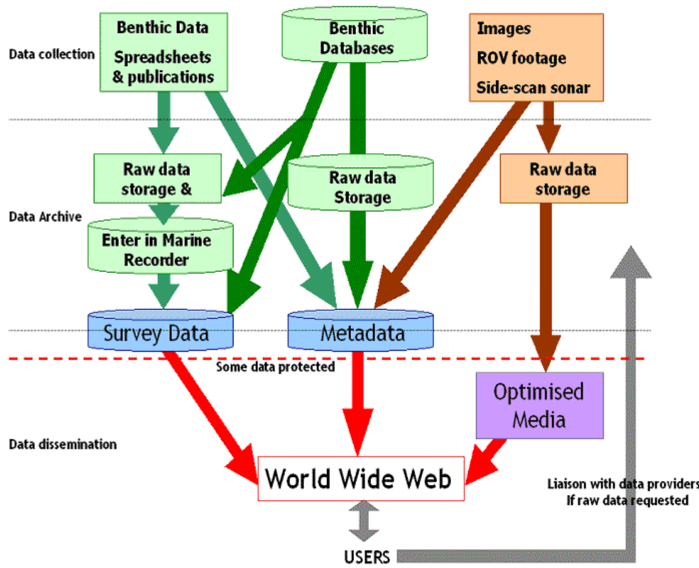


Figure 2-3 Seabed Species and Habitats (UK Marine Data Archive Centre , 2013)

Data flow diagrams (DFDs) are one of the three essential perspectives (the other two being Logical Data modeling and Entity/Event Modeling) of Structured Systems Analysis and Design Method (SSADM), a systems approach to the analysis and design of information systems. With a dataflow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented. The old system's dataflow diagrams can be drawn up and compared with the new system's dataflow diagrams to draw comparisons to implement a more efficient system.

SYSTEM CONTEXT DIAGRAMS

A System Context Diagram (SCD) is the highest level view of a system, showing a system as a whole and its relationship to the environment where the system works (Wikipedia, 2013). SCDs are a type of Data Flow Diagram, and they should always be produced as DFDs. Context Diagrams show the interactions between a system and other actors with which the system is designed to face. SCD is very helpful in understanding the context in which the system will be working. Figure 2-4 shows a context diagram of Pixton Book Inventory and Patron Check-Out System.

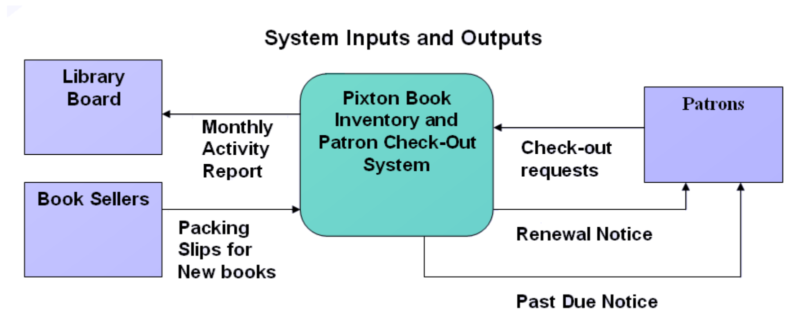


Figure 2-4 Pixton Book Inventory and Patron Check-Out System

The primitives of context diagrams are labeled boxes and lines. The system itself is described in a colored box, while the elements of the context are in different colored boxes. The relationship, labeled with a subject-verb-object format, is drawn as a line between the environment elements and the system being developed. For example, "patrons make a check-out request", “the system sends a renewal notice”, as shown in Figure 2-4. Context diagrams can also use many different drawing types to represent environment elements, such as ovals, stick figures, pictures, clip art or any other representation to convey meaning.

Context diagrams are used in the early stage of a project to identify the scope under investigation. Therefore, context diagrams are typically included in a requirements document. These diagrams must be read by all project stakeholders and thus should be written in plain language so the stakeholders can understand items within the document.

The best System Context Diagrams are used to display how system interoperates at a very high level or how systems operate and interact logically. The system context diagram is a necessary tool in developing a baseline interaction between systems and environment; environment and system or systems and systems.

BOND GRAPHS

Being a labeled and directed graph that uses vertices to represent sub-models and edges to represent an ideal energy connection between power ports, bond graph is domain-independent graphical description of dynamic behavior of physical systems (Broenink, 1999). Systems from different domains such as electrical, mechanical, hydraulic, acoustical, thermodynamic, can be described in the same way because bond graph is based on energy and energy exchange. Besides, bond-graph sub-models are reusable owing to the fact that they are non-causal.

SKETCH

Sketches serve for expressing abstract ideas, externalizing internal thoughts, and storing fleeting thoughts (Zeng, et al., 2004). Written language can do the same, but sketches have the advantage of conveying visual-spatial ideas directly. This may explain their ubiquity: maps, patterns, and architectural plans are found in diverse cultures all around the world, incised in stone, etched on leather, impressed in clay, and drawn on paper. Sketches convey abstract ideas vividly by using simple dots, lines, or shapes, etc. A vivid sketch enhances memory and makes understanding and communicating much easier than abstract medium does such as language or mathematical formula. As an old saying goes, *a picture is worth ten thousand words*. Sketches also provide a token for the contents of working memory, therefore relieving the dual burdens of holding and of operating on the content simultaneously.

UML

Unified Modeling Language (UML) is a standardized specification language for object modeling in the field of software engineering. As a general-purpose modeling language, UML includes a graphical notation used to create an abstract model of a system, referred to as a UML model.

There are 12 types of UML diagrams shown in Figure 2-5. All the diagrams are classified into two categories: structural diagrams and behavior diagrams. Structure diagrams, which are composed of class diagram, component diagram, composite structure diagram, deployment diagram, object diagram and package diagram; Behaviour diagrams, including activity diagram, state machine diagram and use case diagram, emphasize what must happen in the system being modeled. Figure 2-6 and Figure 2-7 respectively show a class diagram and a use case diagram used in the design of Automatic Mesh Generation System (Yan & Chen, 2005).

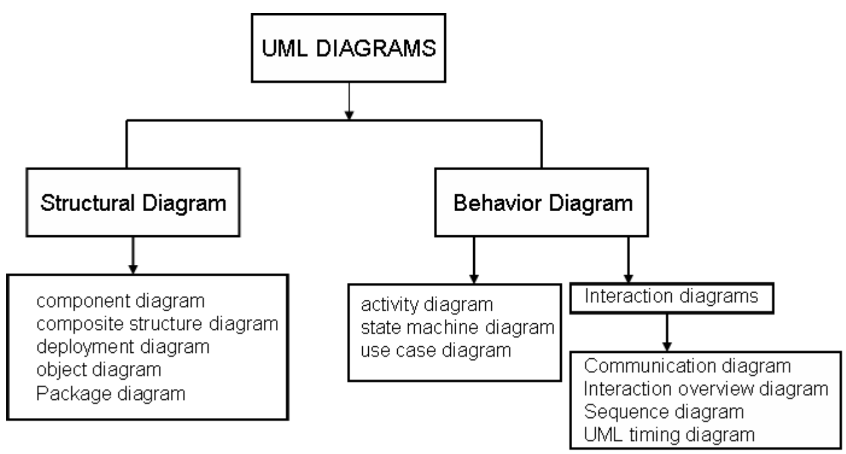


Figure 2-5 UML diagrams

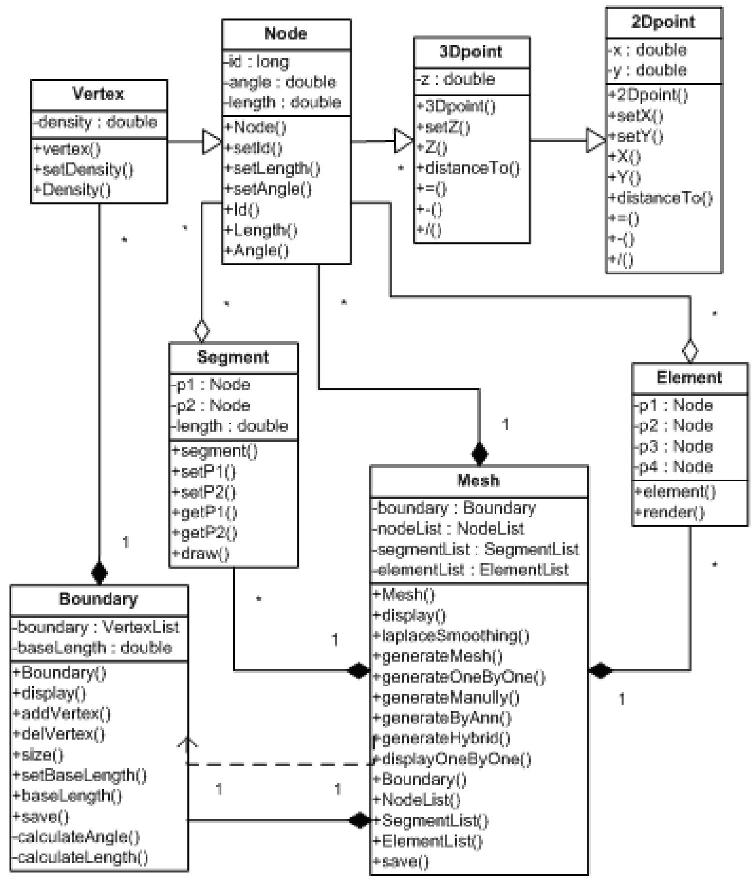


Figure 2-6 A class diagram - software of mesh generation (Yan & Chen, 2004)

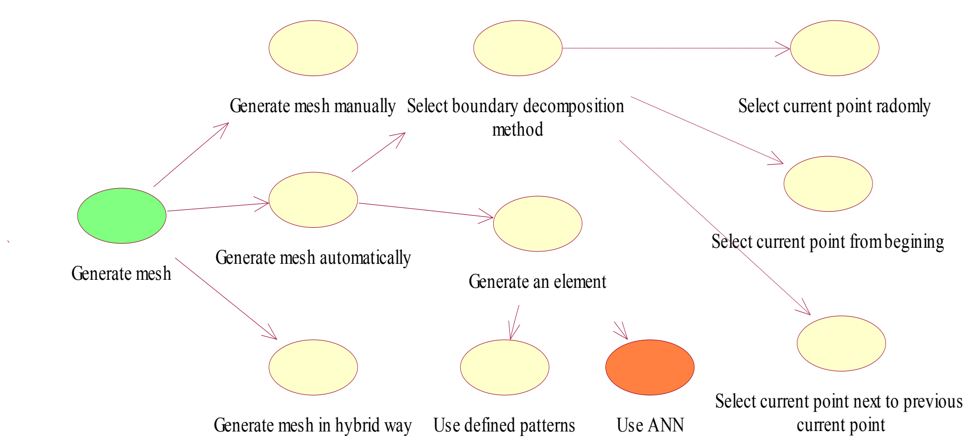


Figure 2-7 Use case diagram – mesh generation

Though aiming at software, UML can also be applied to business process modeling, systems engineering modeling and representing organizational structures.

SYSTEMS MODELING LANGUAGE (SysML)

The Systems Modeling Language is intended to unify the various modeling languages used by systems engineers. Several similarities exist between the methods applied in the area of Systems Engineering and complex embedded systems design. SysML extends the application of UML to systems which are not purely software based, and can in particular be applied to design heterogeneous embedded systems (Object Management Group, 2008). As an example, it provides support for the representation of continuous behavior and flow rates. SysML also introduces a requirement diagram to structure the requirements and link these to the system architecture and test procedures

Originated in January 2001 by the International Council on Systems Engineering (INCOSE), the Systems Modeling Language (SysML) is a Domain-Specific Modeling language for systems engineering. It supports the specification, analysis, design, verification and validation of a broad range of systems. SysML was originally developed by an open source specification project, and includes an open source license for distribution and use. SysML is defined as an extension of a subset of the Unified Modeling Language (UML) using UML's profile mechanism. Figure 2-8 shows the diagram taxonomy of SysML.

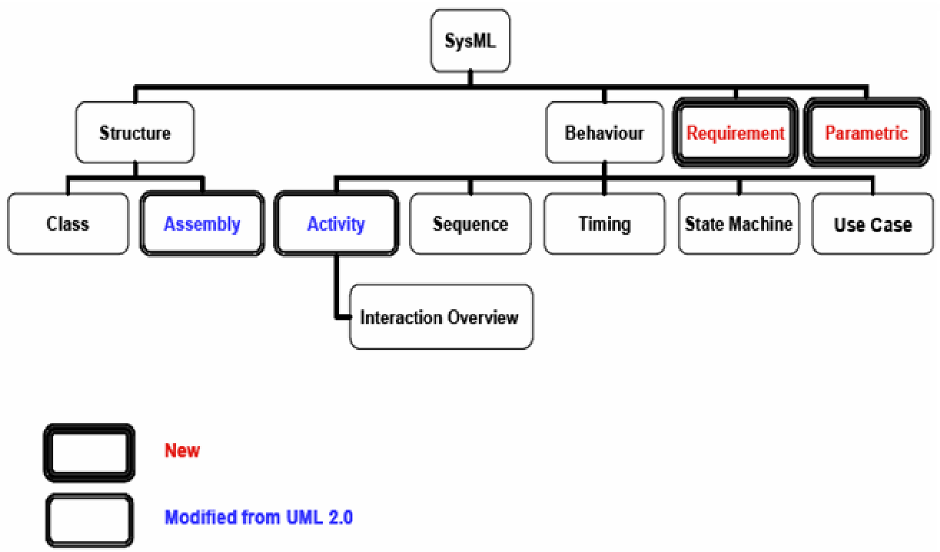


Figure 2-8 SysML structure (American Systems Corporation, 2006)

2.5 Theoretical foundations

This current research relies on Axiomatic Theory of Design Modeling (ATDM) and linguistic analysis tool, Recursive Object Model (ROM). For details, refer to (Zeng, 2002, 2004, 2008). The following presents some fundamental concepts that directly related to this thesis.

2.5.1 Axiomatic theory of design modeling (ATDM)

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

Three primitive concepts are used in the axiomatic theory of design modeling: universe, object, and relation.

[Definition of Universe]

The universe is the whole body of things and phenomena observed or postulated.

[Definition of object]

An object is anything that can be observed or postulated in the universe.

[Definition of Relation]

A relation is an aspect or quality that connects two or more objects as being or belonging or working together or as being of the same kind. Relation can also be a property that holds between an ordered pair of objects.

In addition to these three definitions, three axioms proposed in ATDM will be used as the theoretical foundation for this current research.

[Axiom of the Universal Object]

> Everything in the universe is an object.

[Axiom of Object Relation]

> There are relations between objects.

[Axiom of causality]

> The causal relation is the only plausible relation in all relations between cause and effect.

The axiom of universal object, where all objects are treated the same in that they are objects in the universe, clearly differentiates itself from set theory, where concrete and abstract objects are distinguished as element and set (Zeng, 2002). This axiom lays a foundation for formal representation in two ways. First, it is logically easy to define any mathematical operators that can be applied to objects, such as the structure operator. Second, it maintains all object operations thus defined consistent.

The axioms of object relation and causality imply that any two objects in the universe are related, and causality is the most important relation in a system. Different types of relations lead to different concrete axiom systems (Zeng, 2002).

2.5.2 Recursive object model (ROM)

ROM is a linguistic analysis tool. ROM transforms a textual representation into a diagram representation, from which relationships among objects can be easily identified. ROM defines two types of objects and three types of relations. For more details, refer to (Zeng , 2008). The symbols and their meanings are shown in Table 2-1.

Table 2-1 Symbols used in ROM analysis

Type		Graphic Representation	Description
Object	Object	O	Everything in the universe is an object.
	Compound Object	O	It is an object that includes at least two objects in it.
Relations	Constraint Relation	ξ	It is a descriptive, limiting, or particularizing relation of one object to another.
	Connection Relation	ι	It is to connect two objects that do not constrain each other.
	Predicate Relation	ρ	It describes an act of an object on another or that describes the states of an object.

Chapter 3
Formal Representation of Technical Systems

3.1 Introduction

Any design problem unavoidably involves three parties: the designer, the environmrnt, and the artifact to be designed. Designer is the person who tries to solve the design problem, environment is where the artifact is intended to work, and artifact is what a customer,also regarded as a part of environment, needs in order to achieve some specific goals. As such a complete design theory should cover all these three parties. As explained in previous chapters, environment is anything other than the artifact itself, which includes natural environment, human environment, built environment. The research on designer maily focus on desinger's creativity and cognitive process. In this chapter, I will focus on the artifact and use the term technical system (TS) instead of product as most design researchers do.

As a dispensable part of design theory, the knowledge about technical system (TS) and technical process (TP) deals with the technical system or technical process that a customer wants. Since 1960s, many researchers from different fields have contributed to the understanding of TS, though they may use different terms. Of all the knowledge about TS and TP, the representation of TS is the foundation of other aspects. An intuitive explanation about the importance of the representation is immediate. First, the representation tells us what the system is; secondly, the representation tells us how to use the system; and thirdly, the representation is a tool in aiding design activity and the development of new design methods. Hubka and Eder (1996) once asked, in which manner can technical systems be modeled and represented in all their states of existence, so that the resulting models support different functions of designing? This question can be interpreted as what should be represented and what are the requirements for a good representation? First of all, the representation should reveal all elements that constitute the technical system. Secondly, and more importantly, the representation should correspond to different abstract levels in order to support different stages of design process. Considering that a product design is divided into conceptual design, embodiment design, and detail design, we can correspondingly classify representation into three abstract levels, namely, detailed level, layout level, and topological level. In detailed level, the representation is a detailed description about a product, such as dimensions and surface finish as in mechanical engineering. In layout level, the representation of a system has the information about its subsystems and their relationships. In topological level, the representation is just a concept that is described with symbols or text. Obviously, the first two levels are domain-dependent; the third one is generic and applicable to any technical systems. The research presented in this thesis deals with the third level, aiming to provide a representation fit for all technical systems.

The concept of system has been used for a long time. A system, according to systems theory, is a set of related objects. All systems can be roughly divided into two large categories: natural systems and technical sysetms. Ecological system and human body system are two typical examples of the first group. As the name indicates, a technical system (TS), which is created by human beings, is a finite set of related technical objects. It can be a system with physical form or a system without physical form. Buildings, vehicles, computers, etc., are technical systems with physical forms; while legal system, education system, organizations, etc., are systems without physical forms. The theory about TS plays an important role in helping improve existent techinal systems or develop new design theories. The term technical system was introduced to emphasize the most important characteristic of technical objects: any technical object belongs to a system. From the designer's point of view, technical system is the object that designers intend to create. The whole knowledge about technical system forms Theory of Technical System (TTS). To understand technical systems, the following aspects should be studied (Hubka & Eder, 1988).

1) the purpose of technical systems
2) the structure of technical systems
3) the properties of technical systems
4) the states of technical systems
5) the working principles of technical systems
6) the input and output of technical systems
7) the environment of technical systems
8) the classification of technical systems
9) the complexity of the technical systems
10) The behaviour of technical systems

All technical systems are designed for some purposes: automobiles are designed for transporting people or goods, machine tools for making other products, nuclear power stations for generating electricity, etc. The structure of a technical system is the spatial or logical relationships among the elements. In order for a system to work well, all constituents of the system must be correctly integrated or assembled. Properties are the characteristics that the system holds and that differentiate this system from others. There are two types of system properties (Savransky, 2000): pragmatic and physical properties. Pragmatic properties are goal-oriented, such as suitability for manufacture and transportation, heat-insulating ability, stability, and correction resistance. Physical properties include characteristics and constants of substance and fields. A state of the system is the total set of all constants, fixed parameters, and measured parameter values of all properties of the subsystem at a given time; state decides the system's behaviors. The working principles ascribe how a system works. For a complex system, the working princile can be a set of differnent theories. Environment is where the system works. Because the number of technical systems is infinite, many classification methods have been presented, among which the widely accepted one is the classification according to the principles of action (Savransky,2000). Therefore, we have mechanical systems, electrical systems, hydraulic systems, pneumatic systems, etc. This classification may lead to confusion in some cases because many modern systems are hybrids. Nevertheless, this problem can be solved by using a formal representation.

A technical system is a complex product that may be made up of other simple products. The representation of technical system has two main goals: first, from engineering point of view, a good representation of TS is the prerequisite for manufacturing and using it; secondly, from science point of view, how to describe a technical system is the foundation for any design theory. Nobel Laureate Simon (1996) once wrote "to use it (a complex system), to achieve the simplification, we must find the right representation." By reviewing the existing representation approaches such as graphic and set-based ones, we have observed that they are able to achieve the first goal, but may fail to achieve the second one. In other words, they are not the right representation by Simon's standard. Based on this observation, this paper presents a formal representation of technical systems by using the axiomatic theory of design modelling. Such a formal representation intends to achieve the two aforementioned goals.

Parallel with technical system is the Technical Process (TP), which is composed of any artificial single action or consequences of procedures to perform an activity with assistance of a technical system or a natural object (Savransky, 2000). TP is a set of transformations from input elements into products that respect constraints, require resources, and fulfill some desired purpose (Hubka & Eder, 1988).

There are numerous taxonomies of processes. Roughly speaking, they may be divided into continuous and discrete, stable and unstable, convergent or divergent, cyclic and acyclic, linear and nonlinear, as well as they are grouped according to the name of the domain where they are analyzed. Some commonly examples of physical, technological and biological processes, to name a few, are combustion, crystallization, diffraction, dispersion, distillation, evaporation, hydrolysis, nuclear fission, nuclear fusion, oxidation, reflection, refraction, scattering, sedimentation, sublimation, cell division, fermentation, germination, growth, photosynthesis, transpiration.

Whether natural or technical, all process must follow the natural laws. The only difference between natural process and technical process is that the latter is controllable. Technical process can be accelerated, slowed down, or stopped according to different purposes. All technical processes must be achieved through corresponding technical systems. It is interesting to notice that there is no one-one relationships between technical systems and technical processes. In many cases, one TP can be achieved with different TS's; and one system may perform different processes. For example, the combustion process can be regarded as both natural and technical process. If we want to use this process for our special purpose, we can design different technical systems, such as an engine or an oven.

3.2 Analyse of current theories on technical systems

3.2.1 A review of current research

Modeling and analysis of technical system is a crucial part of any design theory because technical system is the object of design. Recently, researchers have proposed a number of diverse models to describe TS. Basic concepts identified by different researchers include structure, function, behavior, etc. In the following sections, we present major research results about TS.

Hubka and Eder's Theory of TS

In developing theory about TS, Hubka and Eder (1996) have proposed a transformation system in which TS, TP, and therefore their relations play a major role. Hubka and Eder define structure as the internal arrangement of a system, which serves a certain purpose. The behavior of a system is a set of states of a system. Function is the purposeful behavior. A system possesses a set of properties, the total measure of which defines the state of a system at a given time. The authors believe structure decides behavior, behavior contributes function, and the totality of function achieves the system's purpose. Therefore, structure and behavior are the most important properties of a system.

Suh's Axiomatic Design Theory

Suh's axiomatic theory (Suh, 1990) regards design as a continuous interplay between what we want to achieve and how we want to achieve it. His two axioms, independence axiom and information axiom, actually are two basic principles for design. He divides design process into three domains: function domain, physical domain, and process domain. In function domain designer specifies functional requirements that satisfy design objective; in physical domain designer defines physical parameters to satisfy functional requirements; and in process domain designer works with manufacturer on how to achieve design parameters. Suh's two axioms can be applied to evaluation of design solutions.

P&B Systematic Design Theory

The authors (Pahl & Beitz, 2007) treat all artifacts as technical systems (plant, equipment, machine, device, assembly or component) connected to environment by means of input and output. A system is composed of subsystems. There are three ways to divide a system: by functional relationships, by assembly operations, and by production planning. As an intended input/output relationship of a system, whose purpose is to perform a task, the function becomes an abstract formulation of the task, independent of any particular solution. A function may include some sub-functions corresponding to subtasks. The meaningful combination of sub-functions into overall function is called function structure. From syntax point of view, a function may be expressed as a noun and a verb. The authors differentiate between main function and auxiliary function. Function defines behaviour. They also define generally valid functions as the lowest level of function structure consisting exclusively of functions that cannot be subdivided further while remaining generally applicable. Working interrelationship comes into existence through physical effects, which must be compatible with each other. Conflicts may emerge some times. The combination of several working principles results in working structure.

TRIZ Design Theory

TRIZ is "human-oriented knowledge-based systematic methodology of inventive problem solving" (Savransky,2000). The authors of TRIZ define technical system(TS) as any artificial object, a set of parts linked in space; technical process(TP) is any artificial single action or sequence of procedures to perform an activity with assistance of a technical system or a natural object, a set of parts links in time. TP and TS are a unified group, which they name as technique. A technique has an input and output. The input are raw substances, fields, information, etc.; the output is a product. Any product works in a certain environment, which is defined as everything outside a technique. All techniques meet the following conditions: designed for a purpose, i.e. execute useful functions; having a set of characteristics or properties and their parameters, the values of which represent the state of technique; and organized in space or time. The authors also define behaviours as a change of properties, characteristics, and parameters between input and output of the subsystem and its environment both in time and in space. Functions is an interpretation of a subsystem behaviour, or, of the physical equations governing its behaviour. There are useful and harmful functions. Structure is the spatial relations and the temporal relations. The structure is internal arrangement of a system, including shape, hierarchy, and organization.

Umeda & Tomiyama's FBS

Umeda and Tomiyama (1995) define function as a description of behaviour abstracted by human through recognition of the behaviour in order to utilize it. Function is represented in the form "to do something" and is an image of behaviour abstracted by human, while behaviour is one or more sequential changes of states. Hierarchy of system include functional hierarchy and structural hierarchy. The authors also propose a function-behaviour-state diagram. States, as the authors believe, comprise internals state and external states. But what is external state? The authors have not given any definition; also they do not distinguish state and structure; they believe that function is subjective and define a map: B→F.

Gero's FBS

Gero(2004) believes function, behavior and structure are three foundmental concepts for a technical system. He defines function as its teleology –what the object is for. Behaviour of an object is defined as the attributes that can be derived from its structure—what the object does. Structure is defined as its components and their relationships—what the object consists of.

C-K Theory

The basic idea of C-K theory (Hatchuel & Weil, 2003) is concept space and knowledge space, and the design process is a transformation process from one space to the other or within the two space. It begins with a disjunction (k->c), ends with a conjunction (c->k). the authors indicate that disjucntion corresponds concept generation, and conjunction corresponds evaluation. C-K theory is about the design process, the authors do not explicitly discuss concepts such as function, structure, or behaviour in theory.

Maier and Fadel 'Affordance-Based Design

Quite different from other researchers, Maier and Fadel (2009) argues that function is not a fundamental concept in design theory. Instead, affordance is the fundamental one. They have identified structure, behaviour, and purpose as three fundamental concepts for any affordance relationships. They argue that "*systems afford behaviors via their structure for a purpose*".

Goel's SBF

The authors (Goel & Bhatta, 2004) define three fundamental concepts in modelling complex systems. They view SBF model as a programming language. In this model, structure is represented in terms of components, the substances contained in the components, and connections among components; function is a process that takes an input and creates an output; behaviour is a sequence of states and transitions between them.

3.2.2 Analysis

Table 3-1 presents a summary of concepts defined by different researchers. We can see that most of the theory or modelling strategies define a number of basic concepts. Only Suh's design theory and C-K theory do not define any concepts about technical system because their theories mainly focus on the modelling of design process. Most authors identify three fundamental concepts: structure, function, and behaviour. While it is generally agreed among researchers that the structure of a system is internal arrangement of system's components, there are some subtle differences in the definitions of function and behaviour, and their relations as well. A descriptive comparison of the differences is given in Table 3-2 and Table 3-3, respectively. Among all the reviewed theories, some of the models mainly aim at the development of computer aided design tool; others mainly aim to understand the nature of general design process.

Table 3-1 Summary of concepts defined

author	structure	function	behaviour	state	affordance	purpose
Hubka	✓	✓	✓	✓	×	✓
P&B	✓	✓	✓			✓
TRIZ	✓	✓	✓	✓	×	✓
Tomiyama	✓	✓	✓	✓	×	✓
Suh	N/A	✓	N/A	N/A	N/A	✓
Fadel	✓	×	✓		✓	✓
Goel	✓	✓	✓	✓		✓
Gero	✓	✓	✓			×
C-K	N/A	N/A	N/A	N/A	N/A	✓

Note: symbol ✓means that the author(s) defines the concept; symbol × means that the author(s) does not define the concept; N/A means not applicable.

Table 3-2 Definitions of function

author	Description of definitions
Hubka	… The purposeful behaviour...
P&B	…Abstract formulation of task (purpose)…
TRIZ	…An interpretation of subsystem behaviour…
Tomiyama	… A description of behaviour…, to do something…
Suh	N/A
Fadel	… The author defines affordance instead of function…
Goel	…Relation of input and output, transition process…
Gero	…System's teleology, what the system is for…

Table 3-3 Definitions of behaviours

author	Description of definitions
Hubka	… A set of states of a system…
P&B	…defined by functions…
TRIZ	…a change of properties…
Tomiyama	…One or more sequential changes of states…
Suh	N/A
Fadel	…system affords behaviour via structure for a purpose…
Goel	...a sequence of states and transitions between them…
Gero	…attributes of a system, what the system does…

From the review above, we observe that except affordance-based theory, which believes affordance is more fundamental than function, other researchers identify function, structure, and behaviour as three fundamental concepts. Structure decides behaviour; behaviour defines function. The relation between behaviour and function are widely discussed. Some researchers believe function is subjective, while others objective. Also, all technical systems have purposes.

Based on this summary and current design practice, we propose that the following five concepts constitute a canonical set for studying a technical system. The five concepts are Purpose, Structure, Function (affordance), Behaviour, and State.

Before presenting formal definitions to those concepts, we will look into the problems that current theories have. As can be seen from the above review, the models proposed by different authors mainly apply to the development of computer-aided design tools or AI. Most of the current works focus on engineering design, especially on electronic and mechanical engineering. Researches on other design fields are insufficient, such as organizational design. To summarize, current researches need to answer the following questions in order to better understand general design process and technical system.

1. What is the origin of function? Is it subjective or objective?
2. Most current functional modeling approaches aim to analyze a technical system other than to design a technical system; are those models equally apply to the design of technical system?
3. Though behavior is important to a technical system, is it possible or necessary to analyze behaviour during early design stage? Especially conceptual design. Some researchers confuse working principles with behaviour. In design stage, especially in conceptual design, it is impossible to know the real behaviour of a system; it is only principle or expected behaviour that we know.

Many theories, such as Tomiyama's and Pahl and Beitz's, believe that function is subjective. It is not. Function comes from the relation between the environment and the

TS to be designed. For instance, a car needs a brake (braking function). Is this function subjective? No, it is quite objective. It comes because the driver needs to control the car, a relation between environment element (driver) and the TS (car). This applies to the steering system to a car, or a door to a building, or a quality department to a manufacture company, etc. In other words, function naturally comes from the relation between environment and the TS. Only the selection of implementing the function is subjective. For example, there are many ways to brake a car; which way should be used is to some extent subjective. A clear understanding of the objectiveness of function of a technical system helps designers focus on the environment analysis when modeling function during conceptual design stage. Otherwise, the designer will focus on their own knowledge and therefore limit their creativity and ignore critical environment elements.

Another problem of the current theory is about the relation between function and behaviour. We observe a contradiction in the Structure-Function-Behaviour modelling. On the one hand, the authors argue that function modelling should not refer to any structure, in which case there is no knowledge about behaviour at all; on the other hand, they try to define function in terms of behaviour. This conflict comes from two confusions: the confusion between working principle and actual behaviour, and the confusion between design process and analysis process. For a general design problem, the design process may include functional analysis, concept generation, and structure implementation. While behaviour analysis for some TS is very important in evaluating and trouble shooting, it is not an indispensable subject in functional modelling stage. Behaviour analysis is conducted only in conceptual, embodiment, and detailed design stages where the working principle and material, if applicable, are selected. .

3.3 Formal representation of technical systems

Compare with other representations such as graphical or textual, a formal representation focuses on the outward form, structure, and relationships of elements rather than content. As a result, formal representation may be applied to different design fiels. In this section, we will present a formal representation of technical system based on ATDM.

3.3.1 Requirements for a good representation

In Section 3.1, we have briefly discussed all aspects of a technical system. In discussing the demands for a representation, Hubka and Edar (1996) have argued that models (representations) should serve all purpose for designing, such as communication, information, experiment, calculation, thinking aid; support methodical and systematic procedure, and the applications of computers; guarantee uniqueness of interpretation and efficient reading; and consider the efficiency and economy of modeling and representation processes. Very similarly, we observe that a representation must include all the information about the system to be studied including purpose, behavior, structure, relations between system members, properties, states, input and output, types, and environment; and moreover, a good representation approach must reflect the nature of design process and the logic of design, and therefore to help the development of design theories. As pointed in (Zeng & Cheng, 1991; Zeng & Gu, 1999), the design process is of recursive nature and the design activities follow recursive logic. Product design usually needs a series of processes or stages. Accordingly, the representation should be able to be applied to all those stages. Concretely, a ideal representation should meet the following requirements:

(1) Completeness: all information about the system must be included in the representation.

(2) Clarity: a clear and simple representations help understand the system; meanwhile clarity must imply uniqueness of interpretation.

(3) Independence: the representation does not depend on any technical fields.

(4) Flexibility: the representation should be able to describe systems in different design stages or in different states.

(5) Adaptability: the representation is open to updating the system

These five requirements show different priorities. The first three requirements suggest basic requirements, by meeting which a representation approach can achieve the first goal, i.e. to help engineers understand the system; the fouth and fifth requirements present advanced requirements, by meeting which the representation can achieve the second goal, i.e. to lay a foundation for developing new design theories.

3.3.2 Graphic representation

Graphic representation is the most intuitive and simplest method in describing a technical system. In this research, we use graphs as a synonym for diagram that represents information according to some visulization technique. Cognitive studies show that a diagram or a graph automatically support a large number of perceptual inferences, which are extreamly easy for humans (Larkin & Simon, 1987). For example, when you ask for direction, it is always easier to follow a map than to follow verbal instructions. In scientific research, graphic or diagramatic representation has been widely studied in AI for problem solving. In the practice and research of design, a commonly used graphic method is hierarchical tree representation, which can greatly help designers in conceptual design stage. Figure 3-1 shows a typical tree diagram representing a technical system.

According to Simon (Simon, 1996), most complex systems are hierarchic ones composed of interrelated subsystems, each of which in turn contains other lower level subsystems. Many researchers have applied this representation to their theories, such as Hubka and Zeng (Hubka & Eder, 1988; Zeng & Gu, 1999a, 1999b).

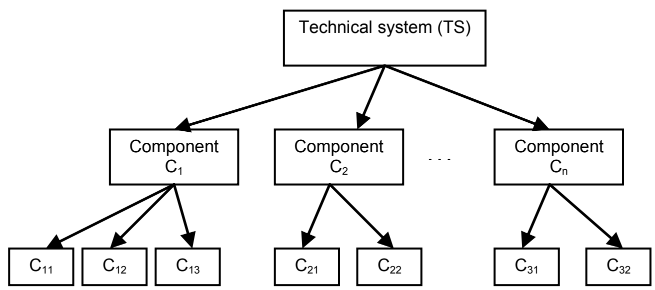


Figure 3-1 A tree representation of a technical system

Graphic representation clearly shows the hierarchic characteristic of complex system, as well as the components in the system. For example, a modern car can be described in graphic form as shown in Figure 3-2.

First of all, we introduce some symbols to denote car components.

MP: Miscellaneous auto parts

BE: Body and exterior

EE: Electrical and electronic

IN: Interior

PC: Powertrain and chassis

EN: Engine system

SS: Suspension and steering systems

TR: Transmission system

WT: Wheels and tire parts

BS: Braking system

ES: Energy source

AS: Apply system

ET: Energy transmission system

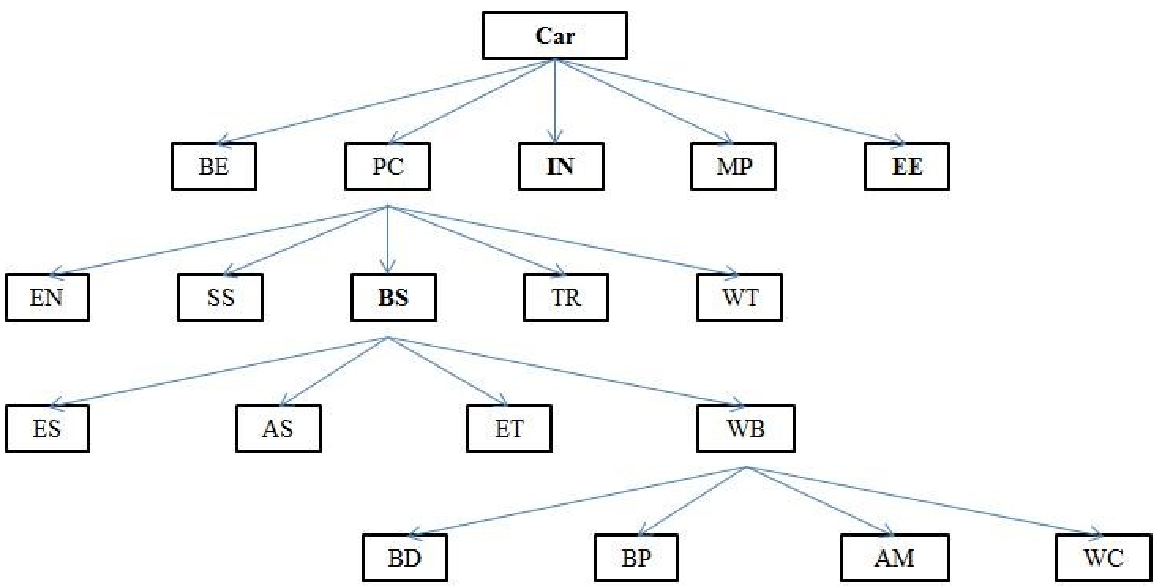
WB: Wheel or foundation brakes

WC: wheel cylinder

BD: brake disk

BP: brake pad

AM: Adjusting mechanism



MP: Miscellaneous auto parts, BE : Body and exterior, EE : Electrical and electronic, IN : Interior
PC : Powertrain and chassis, EN : Engine system, SS : Suspension and steering systems
TR: Transmission system, WT : Wheels and tire parts, BS : Braking system, ES : Energy source
AS : Apply system, ET : Energy transmission system, WB : Wheel or foundation brakes
WC : wheel cylinder, BD: brake disk, BP : brake pad, AM : Adjusting mechanism

Figure 3-2 Car system in tree form

Obviously, this car system has a hierarchic structure. A reader can know a certain amount of information about a car by just a glance. However, there will arise a serious question about this diagram: what are the relations among those componenets? For we cannot see any other relations except for the hierarchic relation. Moreover, it is difficult for the graphic representation to satisfy the flexibility and adaptability requirements, which will be discussed in detail later in this chapter.

3.3.3 Set-theoretic representation

Since a system is defined as a set of interrelated objects, it is natural to apply set theory to representing a technical system. In the literature, this method was used by Yoshikawa in the General Design Theory(Yoshikawa, 1981), Braha and Maimon in their Formal Design Theory (Braha & Maimon, 1998), and Salustri and Venter in their Axiomatic Theory of Engineering Design Information (Salustri, 1992).

Simply speaking, by using set theory, a system can be described in the form $S=\{C_1, C_2, \ldots, C_n\}$, where C_i is a component of S. C_i can be a compound object that consists of other sub-components, or is an atomic object that cannot be decomposed into any sub-objects. For example, as a technical system, an automobile may have a gear-box as one of its component, and this gearbox is also composed of several sub-components, such as input shaft assembly, output shaft assembly, intermediate shaft assembly, gearbox case, etc. All of these assemblies contain many compound objects such as bearings and synchronizers, and atomic objects such as nuts and washers. In set-theoretic representation of a technical system, we can write an vehicle system as follows.

Automobile=

{Body and exterior, Electrical and electronic, Interior, Powertrain and chassis, Miscellaneous auto parts}, and

Powertrain and chassis =

{Brake system, Engine system, Suspension and steering systems, Transmission system, Wheels and tire parts}

Hydraulic Brake system =

{Energy source, Apply system, Energy transmission system, Wheel or foundation brakes}

Wheel brake =

{Wheel cylinder, brake disk, brake pad, Adjusting mechanism, etc.}

Like graphical representation, set representation only tells the constituent components included in a system, but has difficulty in representing the aggregation process, in which a system can be built up from sub-systems and atomic objects.

Another way, borrowed from general system theory, is to use a 2-tuple, i.e. a system $S=(M,R)$, where S represents the system, M is a set of objects, and R is a set of binary relations, $R \subset M \times M$ (Villacampa & Uso-Domenech, 1999). At first glance, it seems that this representation includes both components and relations; however, those relations have to be predefined.

3.3.4 Description of technical systems

In terms of ATDM, everything in the universe is an object, and any object is related to other objects. A technical system can be regarded as a complex object that contains a number of other objects, which are named as subsystems in the conventional system language. Using the structure operator defined in ATDM (Zeng, 2002), we can write a system S as shown in Equation 3-1.

$$\oplus S = S \cup (\mathrm{S} \otimes \mathrm{S}). \qquad \textbf{(3-1)}$$

Without loss of generality, suppose that S is composed of n subsystems, some of which may be only atomic objects. We have

$$S = \bigcup_{i=1}^{n} S_i \qquad \textbf{(3-2)}$$

Therefore, the system S can be represented as

$$\oplus S = \oplus(\bigcup_{i=1}^{n} S_i) = \bigcup_{i=1}^{n} S_i \cup \bigcup_{i,j=1}^{n} (S_i \otimes S_j) = \bigcup_{i=1}^{n} (\oplus S_i) \cup \bigcup_{\substack{i,j=1 \\ i \neq j}}^{n} (S_i \otimes S_j) \qquad \textbf{(3-3)}$$

Equation 3-3 shows that the structure of a technical system can be represented with its subsystems and the relationships between them.

Also, it should be noted that both logically and physically $S_i \otimes S_j$ is not necessarily the same as $S_j \otimes S_i$. Let R be a relation that contains all possible relations encountered in all current existing technical systems, and let $r_{ij} = S_i \otimes S_j$, then we can define r_{ij} as

$$r_{ij} = \begin{cases} r \subset R \;\; if\ S_i\ is\ related\ to\ S_j, or\ i = j \\ \Phi \quad otherwise \end{cases} \qquad \textbf{(3-4)}$$

Again, we take the car system as an example. Using the formal representation, we have

$$\oplus car = car \cup (car \otimes car).$$

$$\oplus car = \oplus(MP \cup BE \cup EE \cup IN \cup PC)$$
$$= \oplus MP \cup \oplus BE \cup \oplus EE \cup \oplus IN \cup \oplus PC \cup R$$
$$where\ R = \bigcup_{\substack{i,j=1 \\ i \neq j}}^{5} (S_i \otimes S_j), \tag{3-5}$$
$$S_1 = MP, S_2 = BE, S_3 = EE, S_4 = IN, S_5 = PC$$

Similarly, Power train, one of the components, can be represented as

$$\oplus PC = \oplus(EN \cup SS \cup TR \cup WT \cup BS)$$

And Brake system BS can be represented as

$$\oplus BS = \oplus(ES \cup AS \cup ET \cup WB)$$

Wheel brake WB is represented as

$$\oplus WB = \oplus(WC \cup BD \cup BP \cup AM)$$

To help further understand, we will give a simple example to show how Equation 3-5 is used. Figure 3-3 shows a simple system. The system is made up of three objects: input shaft A, output shaft B, and the coupling C.

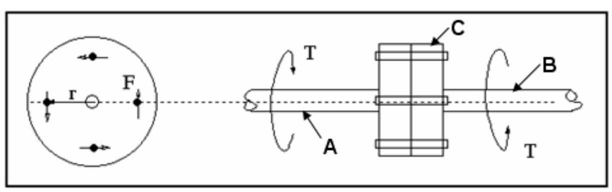


Figure 3-3 Shaft-coupling system

If shaft A rotates, shaft B rotates through the coupling C. We describe the system in formal representation.

$$S = A \cup B \cup C$$

$$\oplus S = \oplus(A \cup B \cup C)$$
$$= \oplus A \cup \oplus B \cup \oplus C \cup (A \otimes B) \cup (B \otimes A) \tag{3-6}$$
$$\cup (A \otimes C) \cup (C \otimes A) \cup (B \otimes C) \cup (C \otimes B)$$

From Figure 3-3, we can see that A is not directly related to B, therefore, $A \otimes B = B \otimes A = \Phi$. Equation 3-6 can be simplified as

$$\begin{aligned} \oplus S &= \oplus(A \cup B \cup C) \\ &= \oplus A \cup \oplus B \cup \oplus C \cup (A \otimes C) \cup (C \otimes A) \cup (B \otimes C) \cup (C \otimes B) \end{aligned} \qquad \textbf{(3-7)}$$

Even without Figure 3-3, we could easily know from equation 3-6 that there is a system composed of three sub-systems A, B, and C. A is connected with C, C with B. From Equation 3-3, we observe that if $S_i \otimes S_j = 0$ for any $i \neq j$ Equation 3-3 will be reduced to

$$\oplus S = \oplus(\bigcup_{i=1}^{n} S_i) = \bigcup_{i=1}^{n} \oplus S_i$$

This means that there are not any relationships between the components of the system. In fact, in this case S cannot be a system. In other words, to construct a system is to construct the relations between different components. Suppose now that we have two systems S_1 and S_2, we want to build up an upper system S containing S_1 and S_2, we have

$$S = S_1 \cup S_2$$

We have two ways to build up a relation between S_1 and S_2, namely direct and indirect. If we can find some means such that $r_{12} = S_1 \otimes S_2 \neq 0$ then we can construct the system S as

$$\oplus S_1 \cup \oplus S_2 \cup (S_1 \otimes S_2) \cup (S_2 \otimes S_1) = \oplus(S_1 \cup S_2) = \oplus S \qquad \textbf{(3-8)}$$

Sometimes, we cannot find direct relations S_1 and S_2 because of some constraints. In this case, a third object or subsystem C is needed such that $S_1 \otimes C \neq 0$ and $S_2 \otimes C \neq 0$. The new system is constructed as the follows.

$$\begin{aligned} &\oplus S_1 \cup \oplus S_2 \cup (S_1 \otimes C) \cup (C \otimes S_1) \cup (C \otimes S_2) \cup (S_2 \otimes C) \cup \oplus C \\ &= \oplus(S_1 \cup S_2 \cup C) \\ &= \oplus S \end{aligned} \qquad \textbf{(3-9)}$$

At this point, we have given a formal representation that can be applied to describing an existing system as well as constructing a new system.

3.3.5 Environment of a technical system

In Section 3.3.4, we focus only on the description of systems themselves. However, environment plays an important role for the system to work well. By applying ATDM, we define

[Definitionof Environment]

The environment of a system is everything except the system itself.

We can graphically describe this in Figure 3-4. Any system works under a certain environment. They interact with each other.

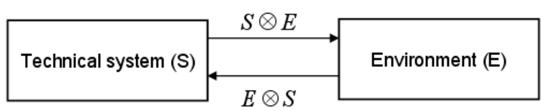


Figure 3-4 Technical system and its environment

3.3.6 Completeness of the formal representation

As discussed in the previous sections, a representation of a technical system must include the following information about the system: purpose, behavior, structure, relations between system members, properties, states, input and output, types, and environment. Completeness requires that a representation embrace all these aspects. Based on the discussion in Section 3.2, we give the definitions as follows.

[Purpose]: Purpose of a technical system is what we design for or what the system is used for. In terms of ATDM, we may write the formal form as

P: $User \rightarrow TS$

[Structure]: **S**tructure is the set of elements in a system and the set of relations that connect these elements. It can be formally written as

Structure: $\oplus TS = TS \cup (\mathrm{TS} \otimes T\mathrm{S})$.

[Function]: Function is how the TS achieve its purpose. Function corresponds with a component instead of a system. Formally,

Func: TS→*P*

[Behavior]: Behavior is a set of consecutively attained states of a system.

Behaviors include predicable and unpredictable behaviors. The predicable are desired; the unpredictable, in most cases, are not desired. Mathematically, behavior (BHVR) can be described as

$$BHVR = \cup_{i=0}^{i=\inf init} state(i)$$

[State]: State is the total of the measures of all properties of a system at a given time.

Mathematically, a state can be written as

$$State(t) = \sum_{i=1}^{n} prop_i(t)$$

Based on these five basic concepts we can define other concepts as follows.

[Environment]: the environment of a system is everything except the system.

$$E = \neg(TS)$$

[Input]: the input represents the external relationship from environment to system.

$$Input = E \otimes TS$$

[Output]: the output represents the external relationship from system to environment

$$Output = TS \otimes E$$

[Property]: a property is any characteristic or quality possessed by an arbitrary object.

We should pay attention to two points about the definitions of the five fundamental concepts. First, the definitions of Purpose, Structure, Function, Behaviour, and State are recursive and relative. The process of definition is as follows. First, we want to design a technical system having purpose P; then to reach this purpose, we need to define general structural requirements and some functions; for each of those functions, if there exist some structure that possesses the behaviour to achieve the function, then we obtain a solution for this particular function; otherwise, we need to decompose the function, which we can regard as a purpose at this point, into sub-functions. This process is recursive and the definitions of purpose and function are relative. They stop when we are able to define properties of an atomic product.

Second, it is complicated to describe the state of a large system. We must first define some parameters to describe the state. Taking car as an example, to describe its state, we need to know the states of all its components and the states of all sub-components of those components, whether we are interested in running state or others, even for a simple substance like water, there are three states: liquid, gas, solid. But, even for liquid water, the state at 20°C is different from that at 21°C. Also, state can be described either continuously or discretely. For most technical system, the state for the whole system is best described in discrete states. One of the challenges in describing complicated states is to define a proper view of properties.

Now, we exam the formal representation against all these aspects. A typical technical-environment system is shown in Figure 3-4. Let Ω stand for the whole product-environment system. T stands for technical system; E, environment. We have

$$\oplus\Omega = \oplus(T \cup E) = \oplus E \cup \oplus T \cup (T \otimes E) \cup (E \otimes T). \qquad \textbf{(3-10)}$$

E is the environment that can be categorized into three parts (Zeng, 2004): natural environment, built environment, and human environment. The relation $E \otimes T$ is from environment to the system, so it stands for the input of the system. The relation $T \otimes E$ is from the system to environment, so it stands for the output of the system. Now, we consider $\oplus T$. Suppose that T is composed of n elements $T_1, T_2, \ldots, T_n$, we have

$$\oplus T = \oplus(\bigcup_{i=1}^{n} T_i) = \bigcup_{i=1}^{n}(\oplus T_i) \cup \bigcup_{\substack{i,j=1 \\ i \neq j}}^{n}(T_i \otimes T_j). \qquad \textbf{(3-11)}$$

Obviously, $\bigcup_{i=1}^{n}(\oplus T_i)$ is the elements of the system, and $\bigcup_{\substack{i,j=1 \\ i \neq j}}^{n}(T_i \otimes T_j)$ stands for all relations between any two elements; therefore, $\bigcup_{i=1}^{n}(\oplus T_i)$ and $\bigcup_{\substack{i,j=1 \\ i \neq j}}^{n}(T_i \otimes T_j)$ form the structure of the system. As Hubka (1984) has pointed out that structure decides the behaviors. Therefore, Equation 3-11 includes behavior information. In a similar way, we can prove that Equation 3-11also includes information about properties and therefore states of a system.

The discussion above shows that the formal representation satisfies the completeness requirement. In addition, the form and logic underlying this representation are simple and independent of any specific system. Therefore the formal representation satisfy the requirements for clarity and independence. In the following section, we will demonstrate the flexibility and adaptability using a case study.

3.4 Design example of application of formal representation

It is shown in the previous section that the formal representation can be applied to describing technical systems as well as constructing systems. It also shows that this representation is complete, clear, and independent of any specific domain. In this section, we will demonstrate that this representation is not only able to clearly describe the structure of a technical system in different design stages but also able to represent any adaptations of a system.

3.4.1 Introduction: A brief review of some fundamental concepts

Structure of Design Problem:

> A design problem is implied in a product system and composed of three parts: the environment in which the designed product is expected to work, the requirements on product structure, and the requirements on performances of the designed product.

Design Process (Zeng & Cheng, 1991; Zeng & Gu, 1999a):

> The design of a product is a recursive and evolution process, during which the structure and relationships are continuously changing.

Figure 3-5 graphically explains this process.

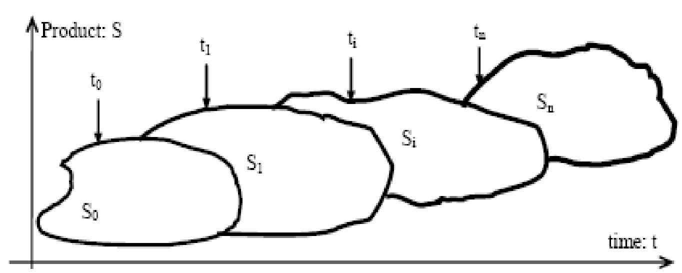


Figure 3-5 Evolution of product in the design process (Zeng, 2002)

In the mean time, the requirement also is changing, as shown in Figure 3-6.

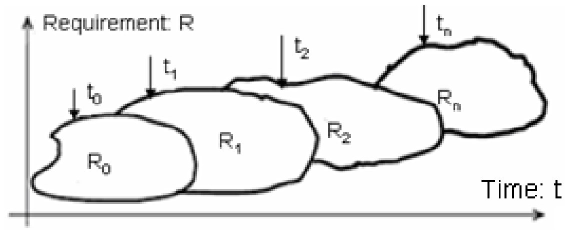


Figure 3-6 Evolution of requirements in the design process

As a result, the state of design is also changing with time. Similarly, we have

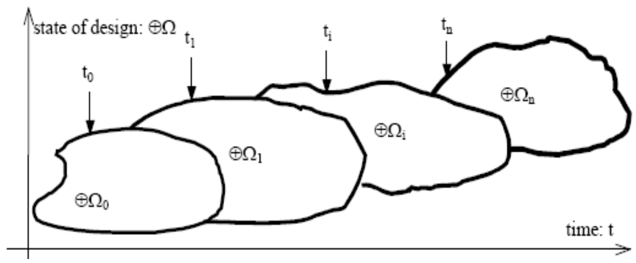


Figure 3-7 Evolution of the design process: refined.

Any product design, especially complex technical system designs, undergo this evolution process indicated in Figure 3-5. During the design process, the product description must be evaluated against the prescribed design requirements to determine if the designed product satisfies the requirements. And, as product descriptions include all the results generated in the dynamic design process, the representation scheme must imply different levels of product descriptions (Zeng & Gu, 1999b).

3.4.2 Description of the design problem

Design a hydraulic system used for moving a worktable from left to right, and return after reaching at desired position. To make the design problem clearly understood, one possible final system to be designed is illustrated as in Figure 3-8.

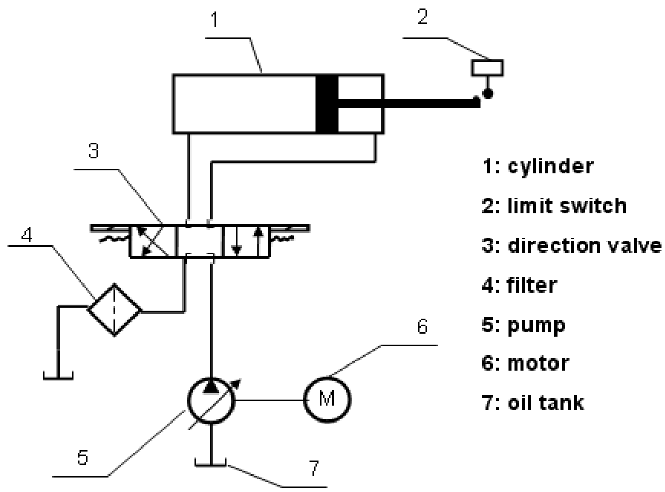


Figure 3-8 Schematic of a hydraulic system

3.4.3 Description of design process using Formal representation

[Definition of primitive product]:

Primitive products are those products that cannot be decomposed into other products, or that need not to be decomposed with respect to the system.

For example, when we design a hydraulic system, we do not need to know the detail structure of a valve, so in this case, this valve is a primitive product. Figure 3-9 lists a set of possible primitive products. For the generation of atomic concepts, please refer to Appendix A, or other papers. This example demonstrates how to represent system structure in different design stages.

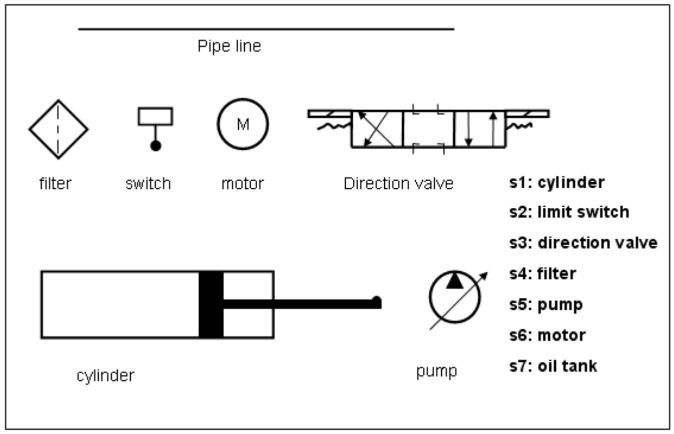


Figure 3-9 Primitive products

At the very beginning, the system $S = S_0 = \Phi$. Φ is an empty set.

Step 1: $S = S_0 \cup s_1$, we obtain a partial system as shown in Figure 3-10. and,

$$\oplus S = \oplus(S_0 \cup s_1) = \oplus s_1 \quad \textbf{(3-12)}$$

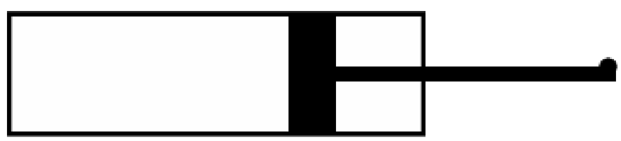


Figure 3-10 Partial system 1 $S = S_0 \cup s_1$

Step 2: $S = S_0 \cup s_1 \cup s_2$ we obtain the second partial system as shown in Figure 3-11 and

$$\oplus S = \oplus(S_0 \cup s_1 \cup s_2) = \oplus s_1 \cup \oplus s_2 \cup (s_1 \otimes s_2) \cup (s_2 \otimes s_1) \quad \textbf{(3-13)}$$

In this special case, we can write $(s_1 \otimes s_2) = (s_2 \otimes s_1) = r_{12}$, so, Equation 3-13 can be written as

$$\oplus S = \oplus(S_0 \cup s_1 \cup s_2) = \oplus s_1 \cup \oplus s_2 \cup r_{21} \quad \textbf{(3-14)}$$

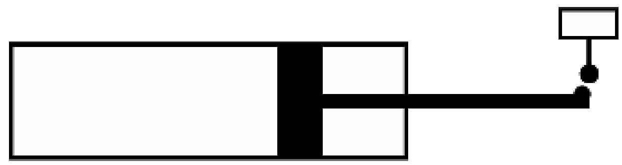


Figure 3-11 Partial system 2, $S = S_0 \cup s_1 \cup s_2$

Step 3: $S = S_0 \cup s_1 \cup s_2 \cup s_3$, we obtain the third partial system shown in Figure 3-12. And the representation is

$$\oplus S = \oplus(S_0 \cup s_1 \cup s_2 \cup s_3) = \oplus s_1 \cup \oplus s_2 \cup \oplus s_3 \cup r_{21} \cup r_{13} \quad \textbf{(3-15)}$$

where $(s_3 \otimes s_1) = (s_1 \otimes s_3) = r_{13}$ and $(s_3 \otimes s_2) = (s_2 \otimes s_3) = 0$

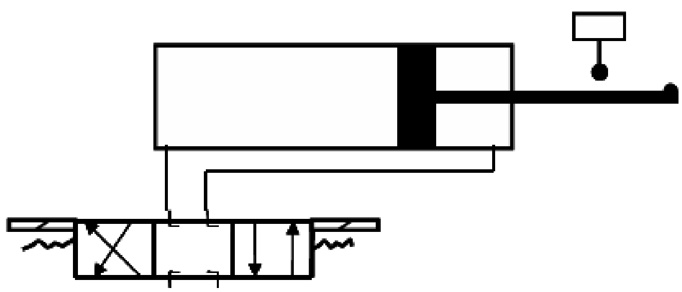


Figure 3-12 Partial system 3 $S = S_0 \cup s_1 \cup s_2 \cup s_3$

In a similar way, we can step by step complete the design of the whole system, which is shown in Figure 3-8. And the formal representation is as in Equation 3-16.

$$\begin{aligned} \oplus S &= \oplus(S_0 \cup s_1 \cup s_2 \cup s_3 \cup s_4 \cup s_5 \cup s_6 \cup s_7) \\ &= \oplus s_1 \cup \oplus s_2 \cup \oplus s_3 \cup \oplus s_4 \cup \oplus s_5 \cup \oplus s_6 \cup \oplus s_7 \\ &\quad \cup r_{12} \cup r_{13} \cup r_{34} \cup r_{35} \cup r_{56} \cup r_{57} \end{aligned} \quad \textbf{(3-16)}$$

Where, $(s_3 \otimes s_4) = (s_4 \otimes s_3) = r_{34}$, $(s_3 \otimes s_5) = (s_5 \otimes s_3) = r_{35}$, $(s_5 \otimes s_6) = (s_6 \otimes s_5) = r_{56}$,

$(s_5 \otimes s_7) = (s_7 \otimes s_5) = r_{57}$

This design example clearly shows that the formal representation explained in Section 3.3 can describe technical systems in different design stages. From Figure 3-8 to Figure 3-12, we can see this representation indeed reveals the recursive and evolution nature of design process.

3.5 A comparison between the three representations

In this section, we will briefly compare these three representation schemes according to the five requirements, which are rewritten as follows.

- 1) Completeness
- 2) Clarity
- 3) Independence
- 4) Flexibility
- 5) Adaptability

It is easy to see that all of the three representations, i.e. graphic representation, set-theoretic representation, and formal representation, satisfy the first three requirements, and therefore the first goal of representing a system, to help people understand, manufacture, and use the system, can be achieved with all of them.

For the last two requirements, the example given in Section 3.4 shows that the formal representation scheme imply different levels of product descriptions, and therefore satisfy the requirements for flexibility and updatability. But, the graphic and set-theoretic representation cannot achieve these because both of them cannot clearly express relations between different elements. For example, the representation $S=(M,R)$ does not have the updating mechanism. Suppose that, at a certain design stage, we have obtained a partial system $S_1=(M_1,R_1)$, and another partial system $S_2=(M_2,R_2)$, needs to be added to form a new system $S=(M,R)$. In this case, the element set M of the new system can be written as

$M=M_1 \cup M_2$, but what about relation set R? It is obviously that $R \neq R_1 \cup R_2$. According to the definition given by Villacampa and Uso-Domenech (1999),

$R \subseteq M \times M = (M_1 \cup M_2) \times (M_1 \cup M_2)$.

But no next step follows to relate R with $R_1$and R_2. This implies that this representation method is not fit for updating the represenation of systems, therefore is not able to describe systems in different design stages.

3.6 Relations

This section will look into the relations in a technical system. Since a technical system is made up of many related components, the number of components and the nature of relations among those components decide the complexcity of a technical system. To understand a system, we must analyse the relations among the system's components.

3.6.1 Scope of technical system

The many different definitions of deign given by different authors indicates a wide range of design fields. It is difficult to cover all design fields with one definition. This presents a question about design research: what is the scope of design research, or in other words, what is the application scope of a design theory? In order to define the scope of design theory, we propose the following assumption.

Any man-made object needs a design before it is made.

For complex technical system, this assumption obviously holds. For example, if we want to manufacture a submarine or build a building, we must follow a well-defined design. For very simple objects, this assumption is not obvious, but holds too. For example, when we want to make a pencil now, we may not need a drawing, but the first man who made pencils must have had a clear design that described the shape and material for the pencils.

Base on the above assumption, design theory can be applied to the following three fields: engineering design, organizational design, and software system design. Engineering design is the commonest field that heavily depends on design practice, such as civil engineering, mechanical engineering, chemical engineering, computer engineering, electronic and electrical engineering, etc. For instance, when we need a new type of vehicle, a new bridge, or a new dishwasher, etc., we must design them first and then manufacture. Most of the current research in design theory is based on and applied to engineering field. Another field that needs design theory is software system design, such as a ticket booking system or a bank account management system. Though hardware is an indispensable part of such kind of systems the design focus is on software relations. The major difference between an engineering system and a software system are in the form of the systems and the laws that the designs follow.

In addition to these two fields mentioned above, organizational design also needs design theory, though people often do not realize they are using design theory when they try to create a new organization. The organizational design practice includes designs of a manufacturing company, of a social organization such as education system, transportation system, legal system, or government organization, etc. The objective of organizational design is to create a concise and effective organization that can achieve a certain purpose. Obviously, organizational design mainly follows social laws and rules. For example, we want to build up a new company that manufacture some new machines, in addition to satisfying the business requirements for the product, a great attention should be given to the organization structure design, such as how many departments we need, how many people for each department are needed, so that the new company can use their limited resources optimistically.

3.6.2 Classification of technical systems

We use the term *Technical System* in a very broad way in this current research. It refers to all man-made objects, either simple or complicated. Base on the analysis given above, we classify all technical systems into three groups: engineering objects, organizational objects, and software. To make this classification easy to understand, we consider that all objects made mainly according to natural science or laws are engineering system; that all objects created mainly according to social science are organizational systems; and that all computer programs, which mostly rely on logic and mathematics, are called software objects.

All engineering objects can be further divided into two large groups: movable and immovable systems. Movable systems include all kind vehicles that have a purpose of moving man or things. Immovable systems include such objects as buildings, bridge, machine tools, devices, etc. We may be able to enumerate all types of man-made movable systems (objects); but it is hard to list all immovable objects.

Software objects include computer-aided software, real-time control software, and information management software. Computer-aided software is a type of programs that aid human beings as a tool, such as word processors, CAD tools, or FEA software, etc. Real-time control software is used to control technical systems, such as missile control system. Real-time control software may include three parts: data acquisition, data processing, and control signal sending. Information management software is mainly to create and maintain data; typical applications include bank account management system, ticket booking system, etc. This type of software usually comprises three major parts: database, interface, and business logic.

Organizational objects include all levels of government, business organization, community organization, and research institute. It is obvious that organizational objects are all about how to arrange personnel so that an organization may have high efficiency and output. All organizations have a clear hierarchical structure, personnel in lower rank report to those in higher rank.

It is straightforward that first two categories definitely need design theory, but some people may have doubt about the third category. It seems in practice that when an organization is established, there is no design involved. This is because those people who want to create an organization have an internal picture already or they just simply follow the precedent. We may regard creating something by following precedent as an emulation design, and creating something according to the designer's internal image as an experience-based design. Both methods are intensively used in today's deign fields.

3.6.3 Relations in a technical system

Any scientific research is about finding relations; there is no exception in design research. Unlike other scientific research such as physics, chemistry, whose major objectives are to investigate causal relations between events or phenomena, design studies deal with far more categories of relations. As already defined before, a technical system is a set of related objects; a relation is the way how two objects are related. This relation ascribes how two objects are structurally and functionally connected and affected; therefore, a major part of design practice is to build up a set of causal relations. The system exerts a requirement on the type of relation, and in turn the nature of the relation influences the performance of a system. Because of the large variety of relation types, it is a challenging task to identify all of them, and yet it is crucial for describing and constructing a system. We can classify them into two general categories: causal relations and constructional relations. Strictly speaking, causal relations only exist between an object and its components, and between a system and its input. The former is named as internal causal relation; the latter, external causal relation. A constructional relation is about how two or more sub-systems are cooperated together so some system's function can be achieved. For example, an engine and a transmission in a car are constructional related; there are no internal causal relations between them. In other words, the function of a transmission is not related to the function of engine, though when an engine fails, the transmission cannot rotate; so we say engine and transmission under a vehicle system have an external causal relation. Since relation is an indispensable part of any systems, we need to further discuss the relations. In the following section, we will look into causal relations and constructional relations that exist in design fields.

1. Causal relations

Causal relation is such a relation between two objects that if one changes the other will change correspondingly. Mathematically,

$$\forall A, B \quad if\ \partial A \rightarrow \partial B\ then\ A\ causes\ B \qquad \textbf{(3-17)}$$

Note: in this thesis we use symbol →to stand for to cause something to happen, the symbol ∂ means change.

Statement 1: There is a causal relationship between a technical system and any of its sub-system. Mathematically, this can be represented as follows.

$$if\ a \subseteq S, then\ \partial a \rightarrow \partial S \qquad \textbf{(3-18)}$$

Statement 2: Causal relations are transitive, i.e. for three objects A, B, and C in a system

$$if\ \partial A \rightarrow \partial B\ and\ \partial B \rightarrow \partial C\ then\ \partial A \rightarrow \partial C \qquad \textbf{(3-19)}$$

Statement 3: For a complex technical system, there exist some causal orders among causally related components of the system.

2. Constructional relations

A constructional relation between two objects is such a relation that the two objects both belong to an upper system, they are not causal related but are needed to work together so that a system can function properly. For any constructional related objects, on the one hand, they cooperate to achieve some goal; on the other hand, they compete with each other for resource allocated to the system. For example, a manual transmission and a brake in a car have a constructional relation. Though not affecting each other, they are both needed for a car to run; in the meantime, they compete with each other for some resources of the car, such as weight and space.

Any two objects under the same system can be said strongly related, weakly related, or non-related. Two strongly related objects have a causal relation or have such a constructional relation that some system goal can be achieved only when these two objects both function properly. An engine and a transmission in a car, for example, have a strong relation. If two objects are weakly related, the system will still function even if one of them does not function properly, but the performance of the system will deteriorate. For example, the front brakes and rear brakes of a car can be said weakly related. But the two front brakes are strongly related. Non-related objects in a system are called functionally independent of each other. For instance, we can say that the department of psychology and the department of mathematics of a university are independent of each other.

It should be noted that two constructional related objects may have a causal relation too. To build up a relation between two objects, we can use the following ways.

- Force relations: two objects are related by force. For example, two mating gears are related by force relations.
- Geometrical relations: two objects are related mainly from geometry point of view, for example an O-ring and a shaft in mechanical engineering.
- Movement relations: one object moves with respect to another object.
- Information relations: two components have information exchange. These relations are most often seen in the field of software and organizational objects.
- Energy or field relations: two objects are not physically contacted, but related through a field. For example, in a motor, the rotor and the magnets.
- Fixation relation: two objects are permanently fixed together.

In the process of designing a system, the identification of relations and the design of structures are performed recursively. It is noted that this list is far from exhaustive. In fact, the classification of relations, together with their formal representation in an abstract level, is a crucial part for creating design concepts, and is a future research topic in design science.

3.7 Summary

In this chapter, we first review the current theories on techincal system, sumarize the concepts applied to TS, and then we propose a set of fundamental concepts that any theories should discuss. We also identify two major problems existing in current theories.

After extracting a set of basic concepts used in technical system, we present a formal representation for describing technical systems by using axiomatic theory of design modeling. Based on Hubka's observations (Hubka & Eder, 1996) and the two goals of representing a system, five requirements for a good representation are also proposed. Both theoretical derivation and examples show that this formal representation satisfies the five requirements. It encompasses all necessary information about a technical system, reflects the recursive nature of design process, and presents a platform of representing systems in different design stages. As a result, this representation not only applies to describing existing technical systems, but also provides a foundation for developing theories in the field of design science.

A unique characteristic of this representation lies in the application of the two axioms about objects and of structure operator $\oplus$. Based on Axiom 1, relations also are objects. A natural conclusion follows: the elements of a system and the relations between them are mathematically in the same abstract level. This enables the structure operation in different design stages, and therefore the formal representation thus derived can be applied to the description of technical systems during the design process. The application of this formal structure is demonstrated with a detailed design example.

Scope and classification of technical systems are also discussed. It is shown that all technical systems belongs to one of the following categories: engineering objects, organizational objects, and software. For any technical systems, two major relations, causal and constructional, are defined. Causal relation is such a relation between two objects that if one changes the other will change correspondingly. A constructional relation between two objects is such a relation that the two objects both belong to an upper system, they are not causal related but are needed to work together so that a system can function properly. To sum up, To build up a technical system is to generate a set of relations among a set of objects.

Chapter 4
Structure of Design Conflict

4.1 Introduction

The subject of conflict has been intensively studied, not because so much is known in a scientific way but because it is so important a topic (Coombs & Avrunin, 1988). Obviously, it is not an easy task to give a general definition that can cover all possible types of conflicts. In Merriam-Webster dictionary (2007), conflict is defined as competitive or opposing action of incompatibles, or mental struggle resulting from incompatible or opposing needs, drives, wishes, or external or internal demands. In Britannica dictionary (2007), conflict is defined as the arousal of two or more strong motives that cannot be solved together in psychology sense. In the research field, different researchers define conflict in different ways, depending on their background and research interests. Some researchers define conflict in a very general way; while other researchers, instead of giving a general definition, classify conflicts first, and then define each type of conflict. For example, according to English and English (1958), conflict is the opposition of response tendencies. In terms of Harrinton *et al. (*1995), conflict is disagreement between two or more viewpoints on some decision or values proposed in design. Altshuller (1984) has defined three types of conflicts: administrative conflict, technical conflict, and physical conflict. Hanna and Barber (2001) have classified conflicts in the context of design into three categories: goal conflict, plan conflict, and belief conflict. Coombs and Avrunin (1988) have identified three types of conflicts and just simply named them as Type I, Type II, and Type III conflicts. Because of the tremendous diversity of conflict types and their importance in solving problems, some questions may naturally arise. Firstly, in spite of this variety of conflicts, do all those conflicts have a common model or formal structure independent of the contexts? Secondly, if the answer is yes, how do we model and represent it? Thirdly, how can we develop resolution strategies based on the model? In this chapter, we will first present a common model (structure) by analyzing typical definitions given by different researchers and descriptions of different conflict situations; and then validate the model through comprehensive examples. For the third question we will answer in the following chapter.

4.2 ROM analysis of definitions of conflict: conceptual structure

ROM is a computer-aided tool applied to linguistic analysis. For more details please refer to (Zeng , 2008). In the following we will list some definitions of conflict from current literature and analyze them using ROM. This list is not complete, but covers a considerable range of conflict research.

[D1] Physical conflict is a situation that mutually opposing demands are place upon one and the same system.

[D2] Goal conflict is a condition that two or more design goals cannot be achieved together.

[D3] Plan conflict is a condition that there are harmful interactions between agents' plans.

[D4] Belief conflict is a condition that involves inconsistent beliefs.

[D5] Type I conflict is within an individual because he or she has to choose between options, each of which may be better than any other in some respect.

[D6] Type II conflict is between individuals because they want different things but must settle for the same thing.

[D7] Type III conflict is between individuals because they want the same thing but must settle for different things.

[D8] Conflict is disagreement between two or more viewpoints on some decision or values proposed in design.

[D9] Conflict is the opposition of response tendencies.

Definition 1 clearly tells us that a conflict occurs if we want a system to do two mutually opposite things at the same time. The famous Uncertainty Principle (Hawking & Mlodinow, 2008) in quantum physics reveals a typical physical conflict situation: to accurately locate the position and measure the speed of a particle at the same time create a conflict requirement for the energy of a quantum.

Definition 2 indicates that two or more goals cannot simultaneously be satisfied because the goals are against each other or competing something else. In conventional mechanical design, for example, we want a shaft to have a larger diameter in order to obtain higher strength on the one hand, but we also want it have a smaller diameter to save material on the other hand. Obviously, these two design goals, to design a stronger shaft and to make use of less material, cannot be achieved at the same time because these two goals contend for one and the same resource: the diameter of the shaft to be designed.

Definition 3 tells that the plan conflict happens when the sequential relationship is violated in a plan. Such type of conflicts can be easily found in planning manufacturing process of mechanical parts.

The conflicts defined in definition 4 often happen in collaborative designs. For example, in software design, one team may believe the data processing speed is most important while another team may reasonably believe the security is most important. To achieve a higher speed may need a simple algorithm, but to reduce the security risk may need a complicated algorithm. A conflict will emerge because of their different beliefs in algorithm design.

Type I conflict as defined in Definition 5 happens when one has to choose between two or more things. For instance, a student may face a choice of whether going to movie or dancing on a Friday evening. In this situation, two activities, go to movie and go dancing, compete for one resource, the student.

Type II conflict explained in definition 6 happens when individuals want different things but must settle for the same thing. For example, in design new model of racing car, two chief designers may have different preferences for the color of the car body. One may want the car painted in red color while the other may strongly believe green color would be more appealing. Obviously, the conflict arises because the two designers contend for the only resource, the car body.

Type III conflicts happen because two or more individuals compete for one and the same thing. For example, a conflict emerges between two super-power countries when both of them want to be the leader of the world affair.

All the definitions have one common characteristic, that is, a conflict happens when there are competing relationships between two things over another (third) issue or object. In the following section, we will analyze some definitions using ROM.

ROM analysis of definitions

For the sake of simplicity and similarity of all the definition presented above, we will analyze only three definitions given by Altshuller (1984) using ROM analysis. For more definition analysis, please refer to Appendix B. By applying ROM to the analysis of the definitions given in TRIZ, we can obtain three diagrams shown in Figure 4-1.

1. Definition of Administrative Conflict (AC): AC is such a situation that something has to be done but how to do it is unknown.
2. Definition of Technical Conflict (TC): TC is such a situation that if one part is improved another part is deteriorated in the process.
3. Definition of Physical Conflict (PC): PC is such a situation that opposite requirements is placed on same system.

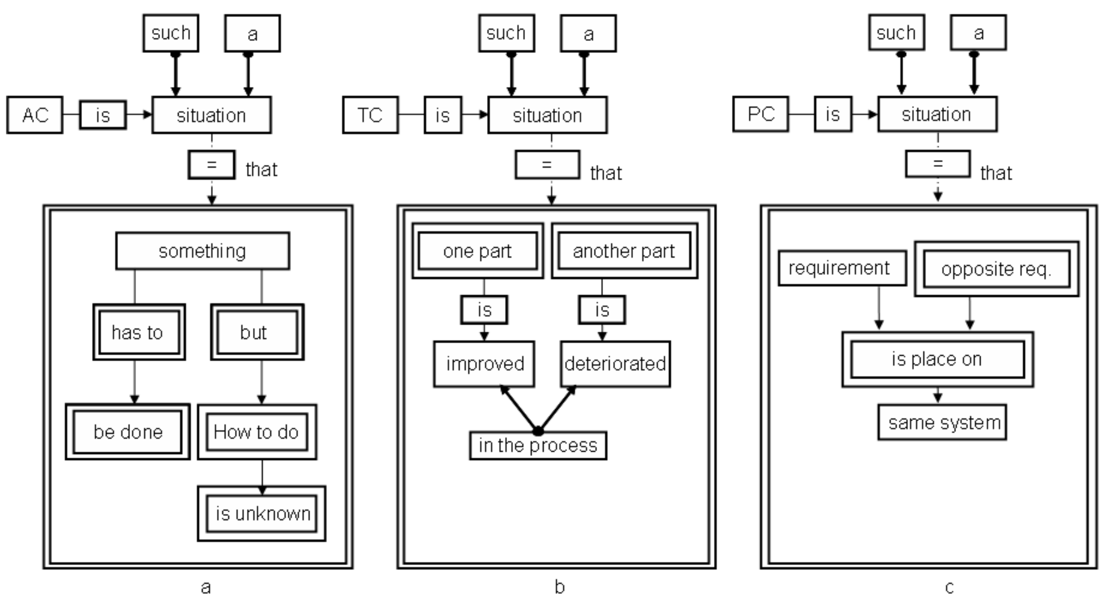


Figure 4-1 ROM diagram of three types of Conflicts

Figure 4-1 has already shown the common characteristic of the three conflicts, but it is not very clear. We can further simplify Figure 4-1 into Figure 4-2.

In Figure 4-2a), b), and c) are the simplified ROM graphical representations of administrative, technical and physical conflicts, respectively. Looking into their structures and recalling that everything in the universe is an object according to ATDM, we observe that all three types of conflicts involve a three-object block shown on the right side of each diagram.

In diagram a), the three objects are object *"something"*, object *"need doing"* and object *"unknown how to do"*. In diagram b), the three objects are compound object *"one part and another part"*, object *"improve"*, and object *"deteriorate"*. In diagram c), the three objects are object *"one part"*, object *"requirement"*, and object *"opposite requirement"*. The relations among the three objects, regardless of the conflict type, are in the same pattern, that is, two objects oppositely relate to the third object. For instance, in a), we need to do something, but do not know how to do it; in b) to improve one part of the compound object leads to the deterioration of the other part of the compound object; in c), we require the object have one certain property and we also do not want this property.

Based on the above analysis, we propose the following conjecture.

[Conjecture of Conflict Structure]:

A conflict is composed of three basic elements: two competing objects and one resource object that the former two objects contend for.

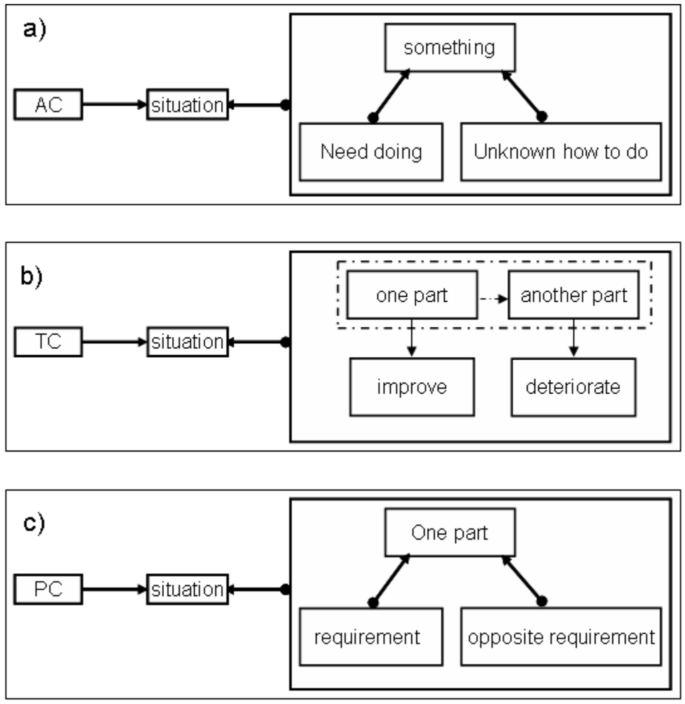


Figure 4-2 Simplified ROM diagrams of three types of Conflicts

4.3 Formal structure of design conflict

4.3.1 Introduction

To understand the conjecture proposed in the former section, we should keep in mind two things: 1) a conflict must have three parts; 2) the resource object is not always obvious, under which circumstance we may feel something wrong yet we cannot see what cause the conflict.

There are many ways to represent the structure of an object, each of which has certain advantages and disadvantages. As we already know, all atomic conflicts are composed of three elements: two competing elements and one resource element. A direct and intuitive way is the graphical representation. Considering we want to design a window for a room, on one hand we want a large window so that more sunlight can come in the room; on the other hand a large window means more energy loss in winter time. This creates a design conflict situation: window should be large and small at the same time. We can represent this conflict situation using a triangle diagram as shown in Figure 4-3.

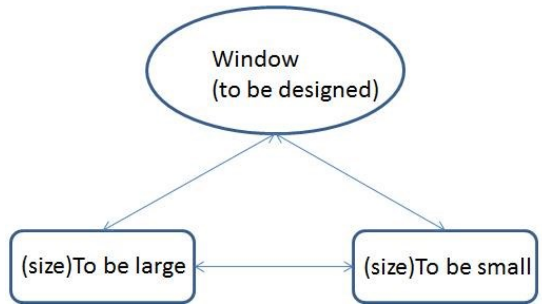


Figure 4-3 Example of conflict

This triangle clearly shows the design conflict: we have conflict requirements for the parameter of a window. In the meantime, we can also represent this conflict using a set as follows.

Design conflict = {window to be designed, a large size, a small size}

Comparing with graphical representation, this set-based method is more formal and simple, but less clear and intuitive. We cannot see the relations among those elements.

4.3.2 A formal representation

Though both graphical and set-based representations are intuitive and direct, they cannot provide logic reasoning aid to designers when they try to resolve a design conflict. In the following part we will propose a formal representation, which focuses on the outward form, structure, and relationships of elements rather than content. Different from graphical and set-based representations, formal representation may provide a logic tool that facilitates conflict identification, conflict resolution, and therefore enhance designer's ability in designing new products.

Based on the concept of object introduced in the axiomatic theory of design modelling, the above conjecture on conflict can be mathematically represented as

$$C = c_+ \cup c_- \cup c_r \qquad \textbf{(4-1)}$$

Where C stands for the conflict, c_+ and c_- are two competing objects, and c_r is the resource object. Equation 4-1 can also be described as in the following figure.

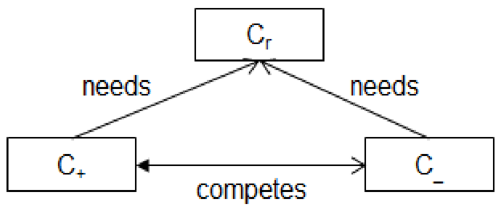


Figure 4-4 Fundamental structure of atomic conflict

Using the structure operator on equation 4-1, we have

$$\begin{aligned} \oplus C &= \oplus(c_+ \cup c_- \cup c_r) \\ &= \oplus c_+ \cup \oplus c_- \cup \oplus c_r \cup (c_+ \otimes c_-) \cup \\ & (c_- \otimes c_+) \cup (c_+ \otimes c_r) \cup (c_- \otimes c_r) \cup (c_r \otimes c_-) \cup (c_r \otimes c_+) \end{aligned} \qquad \textbf{(4-2)}$$

Equation 4-2 tells us that a conflict is also an object composed of three sub-objects. It is the special relationships among the three sub-objects that distinguish a conflict object from other types of objects. Simply speaking, the meaning of Equation 4-2 is that C_+ and C_- both need C_r, but cannot be satisfied at the same time; as a result a conflict occurs between c_+ and c_-. It should be noted that c_+ and c_- can be any objects. They can be either physical or non-physical. The symbol c_r represents any resource in the universe, such as space, time, weight, information, or properties of some objects.

Recalling the triangle structure of an atomic conflict, we observed that three types of relations, shown in Figure 4-5, that may incur conflict situations based on ROM and linguistic analysis (Zeng & Yan: personal communication, Oct. 9,2010).

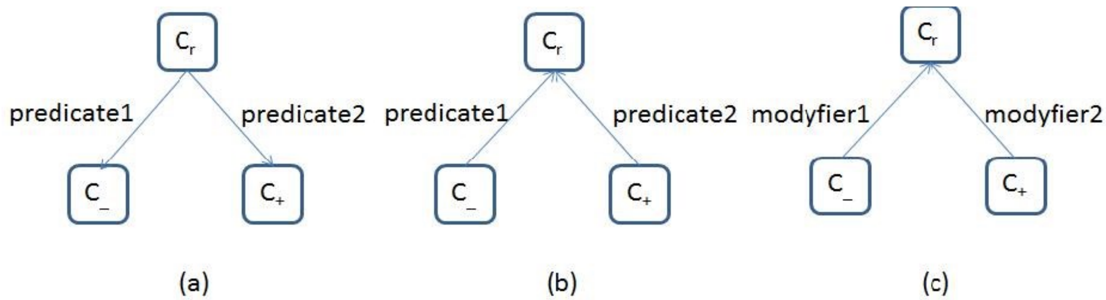


Figure 4-5 Three types of relations

Figure 4-5a indicates that the resource object wants to do two different things at the same time, such as the case that a person wants to go fishing and to go golfing on a particular Sunday. Figure 4-5b shows that two objects want to exert different influences on the resource object at the same time, as in the situation that two countries want to claim the same land. Figure 4-5c represents a situation that an object is required to have two conflict properties, such as in the following semiconductor design that the dielectric constant K needs to be both low and high in the same time for different reasons.

4.4 Examples of conflicts

Although we cannot prove this conjecture by deduction, we can examine it using comprehensive examples that represent different design problems.

Example 1: Semiconductor design

In semiconductor chip design, the insulator should have low dielectric constant K in order to reduce parasitic capacity, but meanwhile the insulator should also have high K in order to store information better (Savransky, 2000). To analyze this example using ROM, we can obtain a simplified ROM diagram for it shown in Figure 4-6.

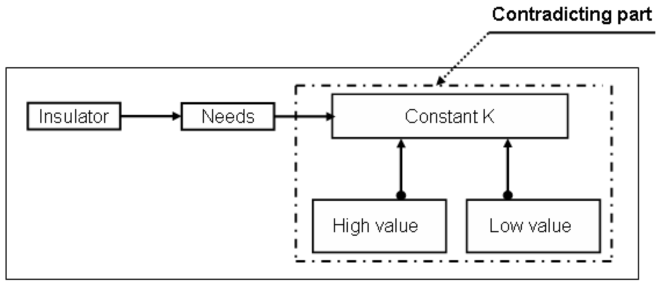


Figure 4-6 Simplified ROM diagram for the semiconductor design example

In this case, a conflict appears because two conflicted requirements are placed on parameter K. By using Equation 4-1, the components of this design conflict can be written as

$$C = c_+ \cup c_- \cup c_r$$

c_+= need low K to reduce parasitic capacity,

c_-= need high K to store information better, and

c_r= dielectric constant K

Example 2: Implant design

In bone implant design, if the material used is too rigid, i.e. with a very high young's module, stress shielding may occur; as a result the bone may become weaker than before. If the material is too soft, i.e. with very low young's module, the patient may suffer more pain in the process of recover.

Here, we identify a design conflict. What are the three objects in this particular conflict? In this case,

c_+=: goal 1, to avoid stress shielding

c_-=: goal 2, to reduce the move amount of the bone during the recover

c_r=: Young's module

$$C = c_+ \cup c_- \cup c_r \text{ ,}$$

In graphical form, this can be described as in Figure 4-7.

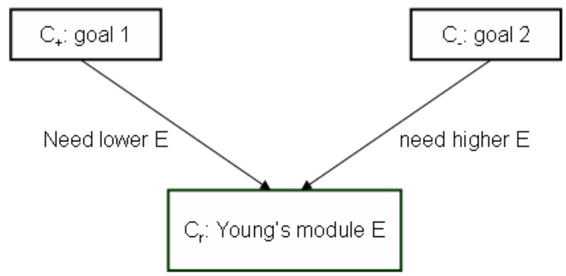


Figure 4-7 Example to show conflict

This figure clearly shows that the two design goals conflict each other. On the one hand, goal 1 requires a lower Young's module; on the other hand, goal 2 requires a higher module.

Example 3: Conflict in engineering design

A metal cylinder is being polished from inside using an abrasive disk. In the process of the work the disk wears away. How can one measure the diameter of the disk without stopping the polishing process and removing the disk from the bowels of the cylinder (Altshuller, 1984)?

The situation explained in this problem can be described in Figure 4-8.

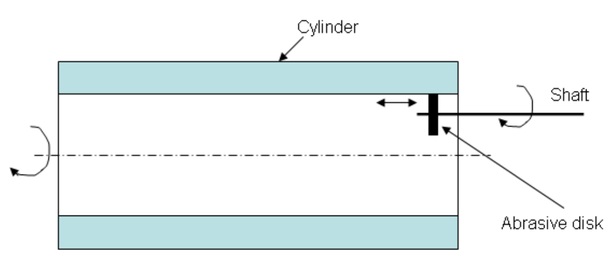


Figure 4-8 Graphical problem description

A conflict happens. On the one hand, we want to stop the process in order to measure the diameter of the disk; on the other hand, we cannot stop the process for some reason such as keeping the high productivity. So the conflict is shown in Figure 4-9 and the three elements are identified as follows.

c_+: Measuring the diameter

c_-: Keeping high productivity

c_r: Polishing process

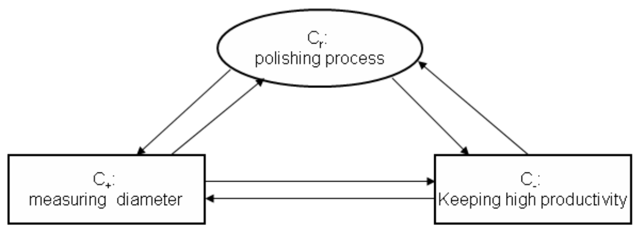


Figure 4-9 Conflict of the design problem

Example 4: A group conflict

In the Project Apollo (Wikipidia, 2011), there were various communities within NASA that differed over priorities and competed for limited resources. The two most identifiable groups were the engineers and scientists. As ideal types, the engineers worked together to build up vehicles that would function reliably within fiscal budget; the scientists engaged in pure research and expand knowledge about the Moon. Nevertheless, they contended with each other over a great variety issues associated with Apollo. For instance, the scientists disliked having to configure payloads so that they could meet time, money, or launch vehicle constraints. The engineers, likewise, resented changes to scientific packages added after project definition because "these threw their hardware efforts out of kilter". Both groups had valid complaints and had to maintain an uneasy cooperation to accomplish Project Apollo.

The above situation embodies two conflicts between the scientist group and engineer group, which can be represented by Figure 4-10 and Figure 4-11.

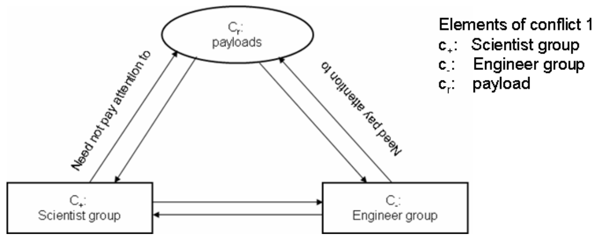


Figure 4-10 Group conflict 1

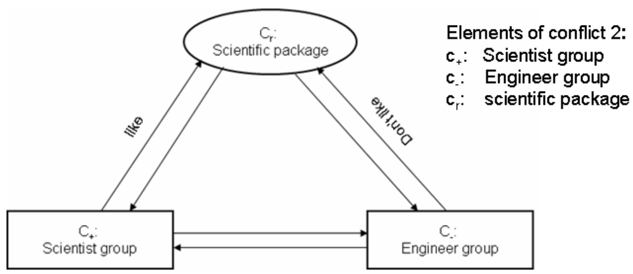


Figure 4-11 Group conflict 2

A compound conflict can be formed by combining these two conflicts, as shown in Figure 4-12.

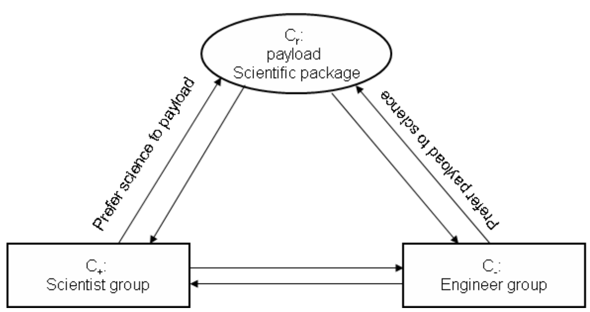


Figure 4-12 A compound group conflict

Example 5: Mechanical design

When we design a shaft, on the one hand, we want the shaft to be very strong by using a larger diameter; on the other hand, we want the diameter to be small in order to reduce the weight and cost. This creates a design conflict: we need larger diameter and a smaller diameter at the same time. This conflict can be expressed in Figure 4-13.

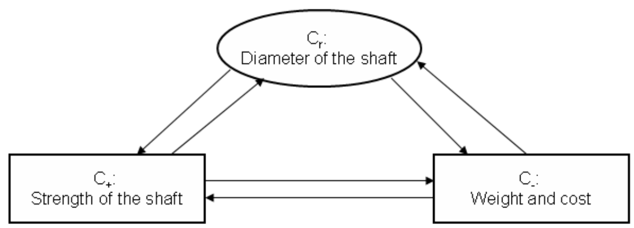


Figure 4-13 A compound conflict in shaft design

This is a compound conflict; one of the competing objects includes two sub-objects, that is, C_- includes weight and cost. This compound conflict can be decomposed into two atomic conflicts as shown in Figure 4-14.

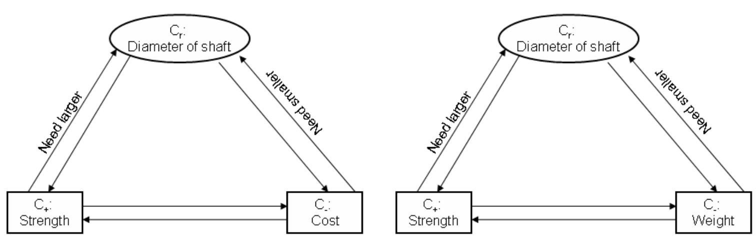


Figure 4-14 Decomposition of a compound conflict

Example 6: Lathe cutter design

There are two important angles on a lathe cutter bit: the top rake and the front clearance. Refer to Figure 4-15. The values of these two angles affect the strength of the cutter, the surface quality of the work piece, and the stability of the cutting action. Some conflicts exist in deciding the values of these two angles. Simply put, the conflict can be described as shown in Figure 4-16.

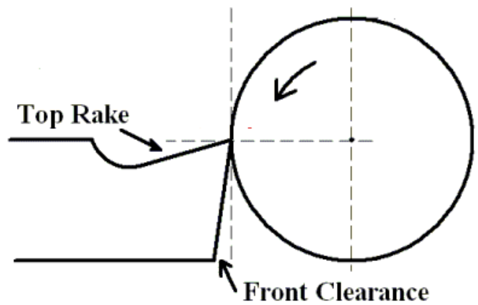


Figure 4-15 A lathe cutter and rotating work

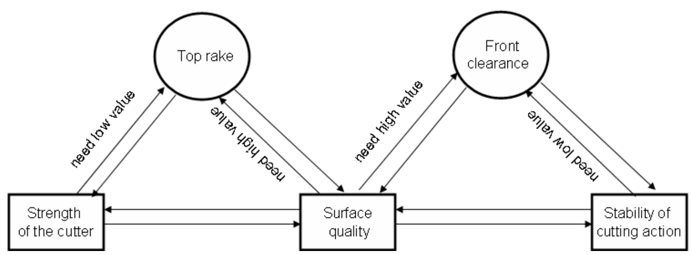


Figure 4-16 Description of complex conflict

4.5 Types of conflicts

The conjecture of conflict structure states that any conflict is composed of three elements. Here the conflict is meant an atomic conflict, the simplest conflict. In reality, however, most conflicts are complicated and a number of objects are involved. If all the objects are linearly related, i.e. the complicated conflict can be decomposed into simple ones, we name this type of conflict compound conflicts; if all or some of the objects are interrelated and the complicated conflict cannot be decomposed into simple ones, we name it complex conflict. In the following we present formal definitions of those different types of conflicts based on the conjecture.

Atomic conflict

An atomic conflict is such a conflict that the three elements are all primitive objects.

$$C_{atomic} = \sum_{i=1}^{3} c_i = c_1 \cup c_2 \cup c_3 \qquad \textbf{(4-3)}$$

Here c_1 and c_2 are competing objects; c_3 is resource object. All of them are atomic objects or can be treated as atomic under discussion. For example, two programmers working on a same project may have different preferences for the programming languages used for the new project. One prefers C^{++}, while the other prefers JAVA.

Compound conflict

If the resource object is a compound object, then the conflict is called compound conflict.

$$C_{cmpd} = \sum_{i=1}^{n>3} c_i = c_1 \cup c_2 \cup c_3 \cup c_4 \cup ... \cup c_n = Cf_1 \cup Cf_2 \cup ... \cup Cf_k \qquad \textbf{(4-4)}$$

C_{cmpd} is composed of more than three objects, but we can group every three objects to form an atomic conflict. In other words, C_{cmpd} can be decomposed into some atomic conflicts. We can easily find examples in international relations. For instance, country A and country B have disputes over some boundary and trade regulations. In this case, the resource object is a compound object that is composed of two sub-objects: *boundary* and *trade regulations*

Simple conflict

Both compound conflict and atomic conflict are simple conflict.

Complex conflict

A complex conflict includes at least two simple conflicts that are coupled.

$$C_{cmpl} = \sum_{i=1}^{n \rhd 3} c_i = c_1 \cup c_2 \cup c_3 \cup c_4 \cup \ldots \cup c_n \neq Cf_1 \cup Cf_2 \cup \ldots \cup Cf_k \qquad \textbf{(4-5)}$$

If some competing objects are involved in more than one conflict that must be resolved at the same time, such as in the case of lathe cutter design, then the conflict is a complex conflict.

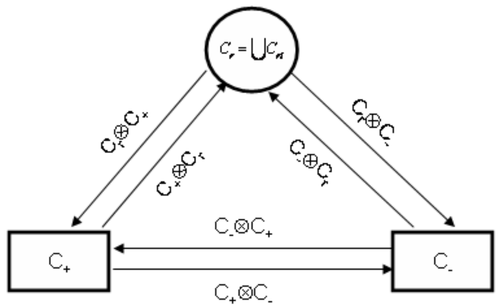


Figure 4-17 A compound conflict

Figure 4-17 shows the structure of a compound conflict with compound resource object. Figure 4-18 shows a complex conflict that includes two simple conflicts.

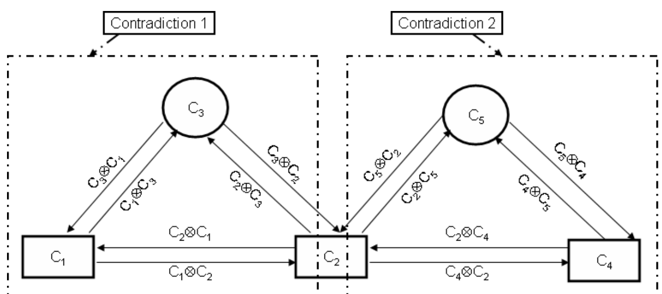


Figure 4-18 A complex conflict

The difference between a compound conflict and a complex conflict is that the former can be divided into two or more conflicts that can be resolved separately, but the later cannot because a complex conflict is composed of several coupled conflicts, which must be considered simultaneously. To resolve a compound conflict, we may first decompose it into several atomic conflicts. Suppose $c_r = c_{r1} \cup c_{r2}$ then the conflict $C = c_+ \cup c_- \cup c_r$, can be decomposed as two atomic conflicts C_1 and C_2, where $C_1 = c_+ \cup c_- \cup c_{r1}$, and $C_2 = c_+ \cup c_- \cup c_{r2}$.

In summary, design conflicts can be classified into complex conflicts and simple conflicts, which include atomic conflicts and compound conflicts.

4.6 Summary

In this chapter, we have proposed a conjecture that an atomic conflict is composed of three elements. Two elements compete for the third one. First, we select a number of typical definitions of conflict from the literature, and then we analyze those definitions suing ROM. The analytical result shows that any conflict regardless of the context always consists at least three parts and therefore can be represented in a formal form. A variety of examples from different design fields agrees with this analytical result.

It should be noted that the formal structure of a conflict proposed in above section is a general form of all types of conflicts; it is not an isolated idea but a generalization of the different definitions of conflict in different contexts, based on extensive analysis and observation using ATDM and ROM. There is no contradiction between this structure and any other recognized definitions or results. Though we aim to apply this structure to design science, the conjecture proposed in the previous section also applies to conflicts existed in the field of social science. As noted in previous sections, conflicts exist everywhere: we can see conflicts among family members, among employees in a company, among different ethnic groups, and among countries. Wherever there are competitions, conflicts will exist. If we look deep into those conflicts, all of them consist of at least three part, or three objects. For example, if there exists a conflict between country A and country B, there must be some issue(s) on which the two countries disagree. Denoting this issue with symbol Cr, we can write the structure of this conflict as

$$\oplus C = \oplus(A \cup B \cup C_r)$$

Before we end this chapter, we will briefly compare the formal structure with another two conflict modeling strategies. One is TRIZ, which is the only literature on engineering conflicts. The other is game theory, which is called the mathematical model of conflicts.

<u>Comparison with TRIZ</u>

TRIZ is the only theory that discusses conflicts happed in engineering design. Three types of conflicts are defined in TRIZ, namely administrative contradiction (AC), technical contradiction (TC), and physical contradiction (PC).

An AC is defined as such a situation that something is required to do but it is unknown how to do it. A TC is defined as a situation that an action is simultaneously useful and harmful; it represents a conflict between two subsystems. A PC is defined as a situation that a given subsystem should have property A to execute a necessary function and property non-A or anti-A to satisfy some constraints. We will demonstrate the difference and similarity using an example.

Problem description:

Detect the number of small (<0.3 micrometers) particles in a liquid with very high optical purity (Savransky, 2000).

1. **Description Using TRIZ**

AC: it is necessary to detect the number of small particles in a liquid with very high optical purity, but what to do?

TC: if the particles are very small the liquid stays optically pure, but the particles are invisible. OR if the particles are big they are detectable, but the liquid is not optically pure.

PC: the particle size must increase to be viewed AND NOT increase in the meantime to keep the optical purity of the liquid.

2. **Description using formal structure:**

First, we will exam if there are some existing method that can be applied to achieving the goal, i.e. to detect the number of small particles. If we can find some satisfied methods, the problem is solved; otherwise, we find a conflict, which can be written in a formal form.

$$\oplus C = \oplus(C_+ \cup C_- \cup C_r)$$

Where C_+ stands for the task of detecting the number of particles, C_-stands for the current environment, and C_r stands for the device we are looking for. In this situation, the meaning of the above equation can be stated as that C_+ needs C_r but C_- cannot provide. Because there is no resource in the current environment that meets our need, a conflict arises. It should be noted that this structure corresponds with the AC defined in TRIZ. Both of them indicate nothing but a problem, which is to design a method to detect the number of particles in a pure liquid.

Second, if we let C_+ stands for *able to be viewed*, C_-stands for *optical purity*, and C_r stands for *the size of the particles*, object C in the above equation also represents the PC corresponding to TRIZ's definition. According to the formal structure, the PC in this example can be interpreted as a situation that two properties of the liquid (able to be viewed and optical purity) put forward a conflict requirement for the size of the particles, both big and small in the same time. Similarly, object C can also represent TC.

Comparison with Game Theory

Having been extensively studied since 1940's, game theory can be viewed as a mathematical modeling of conflicts. It deals with a complicated decision making process in which two or more decision makers, or players in game theory language, are involved, and each player having multiple choices. There are two basic assumptions in game theory made about players (Myerson, 1991): they are rational and they are intelligent. Moreover, since the logical root of game theory is Bayesian decision theory, game theory is also regarded as an extension of decision making theory. In game theory, each player's choices give him different payoff depending on the strategy that he and his opponent would use, and every player's objective is assumed to maximize the expected value of his own payoff. The results of game theory are largely used in social science to study the behavior of decision makers. Following is an example widely used in game theory books.

Prisoners' dilemma (Myerson, 1991): Two players are accused of committing two crimes together, one minor crime that can be proved without any confession and one major crime that can only be proved with at least one player's confession. The prosecutor promises that, if only one confesses, the confessor will go free while the other one will be put in jail for 6 years; if both confess, they will go to jail for 5 year; if neither confesses, they will go jail for 1 year. Denoting play 1 with P_1, player 2 with P_2, confess with f, not confess with g. The player's payoff is measured in the number of years of freedom over the next 6 years. The game can be represented in strategic form shown in the following table.

Table 4-1 Prisoner's Dilemma game

Prisoner's Dilemma game in strategic form

P_1	P_2	
	g_2	f_2
g_1	5,5	0,6
f_1	6,0	1,1

Though there are other representations in game theory for a game, such as extensive form, which is in a tree form, the objective of game theory is to investigate which strategy a player should adopt in order to maximize his payoff. If the conflict in this game is represented using formal structure, it has the same form as $C = c_+ \cup c_- \cup c_r$.

Where c_+ stands for player1, c_- stands for player2, c_r stands for payoff. It is noted that in game theory the representation emphasize the dependency of player's payoff on the strategies that he would use, therefore it basically is a decision process because all choices are already available' while the formal structure represents a conflict situation that a player faces, and each player has the same conflict in mind, i.e., players must choose this or that strategy, but cannot be more than one.

Summary of the Comparisons

Game theory has been a mature field and applied to economics and politics. If we divide conflict management into four stages: conflict identification, conflict resolution, decision making, and decision evaluation, game theory belongs to the third and fourth stage. Though called conflict modeling, game theory is more about decision making than conflict resolution. It can be used to study human being's behaviour during decision process, but cannot be applied to design, especially not applicable to engineering design. Its two assumptions about player's rationality and intelligence are also arguable in reality.

On the other hand, the result presented in this paper belongs to the first two stage of conflict management: conflict identification and resolution. The formal structure can serve as guidance to identifying conflict, especially in engineering design. As we know, in a complex engineering design, requirements analysis constitutes a large part of designers' work. Conflicts among different requirements are unavoidable. With a formal structure of conflicts in mind and aided by proper tools such as ROM, designers can easily spot a conflict. Moreover, a complete analyse of the relations between the three objects in a conflict always indicates the direction to conflict resolution.

Among all the researches on conflicts, TRIZ is the only one aiming at design conflicts, or conflicts in engineering. Grouping conflicts into three types, TRIZ's researchers have proposed some principle and heuristic methods to resolve different conflicts. While related to some concepts used in TRIZ, the formal structure proposed in this current research has a total different theoretical foundation: the conflict defined in TRIZ is based on philosophical dialectic, while the formal representation is based on two axioms from ATDM. The two axioms are 1) Everything in the universe is an object; 2) There are relations between objects. And any atomic conflict is composed of three objects, two competing objects and one resource object. Whenever two objects compete for another object, there will be a conflict. Another difference between our definition of conflict and TRIZ is that we do not differentiate TC from PC as TRIZ does. This is a reasonable treatment of design conflict, for the concept of sub-system is relative. Besides, from TRIZ's definitions, PCs and TCs can be resolved with same principles. The third difference is that this formal structure applies to any types of conflicts, such as conflicts in social science and politics science, while TRIZ does not.

Chapter 5
Conflict Resolution Strategies

5.1 General analysis of the structure of conflicts

Conflict resolution consists of two objectives: to find the strategies for resolving the conflict and to select the best or proper strategy among the possible strategies. Of the two objectives, to find the strategies is a searching process; to select the best or proper solution is a decision-making process. In the following sections, we will investigate the resolution principles based on the formal structure and analysis of relations.

For any conflict C= $c_1 \cup_2 \cup c_3$, by using structure operator defined in Axiomatic Theory of Design Modeling (ATDM), we have

$$\begin{aligned} \oplus C &= \oplus(c_1 \cup c_2 \cup c_3) \\ &= \oplus(c_1 \cup c_2) \cup \oplus c_3 \cup (c_1 \otimes c_3) \cup (c_2 \otimes c_3) \cup (c_3 \otimes c_2) \cup (c_3 \otimes c_1) \end{aligned} \quad \textbf{(5-1)}$$

Let C_3 be the resource object. Equation 5-1 can be divided into three groups: $\oplus(c_1 \cup c_2) \cup \oplus c_3$, $(c_1 \otimes c_3) \cup (c_2 \otimes c_3)$, and $(c_3 \otimes c_1) \cup (c_3 \otimes c_2)$. Term $\oplus(c_1 \cup_2) \cup \oplus c_3$ represents the description of current situation of the system to be considered; term $(c_1 \otimes c_3) \cup (c_2 \otimes c_3)$ tells that both objects c_1 and c_2 require resource c_3; the third term $(c_3 \otimes c_1) \cup (c_3 \otimes c_2)$ represents the effects of c_3 on c_1 and c_2. The logic underlying these three terms is as follows.

Under the present situation represented by $\oplus(c_1 \cup c_2) \cup \oplus c_3$, because c_3 has effects on c_1 and c_2, both c_1 and c_2 need to exert certain requirements on c_3, i.e. $(c_1 \otimes c_3) \cup (c_2 \otimes c_3)$, which leads to conflict if $(c_1 \otimes c_3)$ conflicts with $(c_2 \otimes c_3)$. From equation 5.1, we observe the following. 1) A conflict exists between two related objects; 2) the two objects must compete for a same third object. There may be conflicts that involve more than three objects, but the basic form of a conflict can be represented as in Equation 5-1. More importantly Equation 5-1 indicates the routes to solving conflicts. Because any atomic conflict is composed of three objects, it falls into one of the following 3 cases according to the relations between its three objects.

Case 1: c3 does not exist.

This is the case where c_2 demands something from c_1, but c_1 does not have. This type conflict is comparable to administrative conflict (AC) in TRIZ. In design, it happens when human beings, represented by c_2, have new needs for some existing system, represented by c_1. For example, if $c_1$is the system of all existing transportation tools and c_2 stands for human needs, we can write c_1 = {bicycle, car, train, airplane…}. Suppose that we want to fly across the Pacific Ocean, we can select airplane as c_3, in this case, there are no conflicts. Now if we need a tool c_3 that can carry human beings through the Earth. By checking the system c_1, we learn that no such tool c_3 available, so a conflict arises.

Case 2: c_3 exist, c_1 and $c_2$are two different subsystems

This is the typical situation where two objects or subsystems compete for one and the same limited resource. For example, two people compete for one position; two people and the position form a conflict. In technical design, this situation occurs when we try to improve one subsystem, another subsystem is adversely affected. For example, when we design algorithm to automatically generate mesh in a given area, if the quality of one element is improved, the quality of its neighbour elements may be deteriorated. This means we cannot improve both. This is a conflict between two elements (two subsystems).

Case 3: c_3 exist, c_1 and c_2 are from the same subsystem

This is the case when two conflicted requirements are placed on one parameter, which is comparable to physical conflict as defined in TRIZ.

It should be noted that in a conflict, the relationship between c_1 and c_2 is not always important. Instead, the relation between c_1 and c_3 and the relation between c_2 and c_3 are crucial. Objects c_1 and c_2 can be in the same system, in which case C is a physical conflict, or in two systems, in which case C is a technical conflict.

5.2 Qualitative analysis of relations in a conflict

Considering a conflict $C = c_+ \cup c_- \cup c_r$, from equation 5-1, we have

$$\begin{aligned}\oplus C &= \oplus(c_+ \cup c_- \cup c_r)\\ &= \oplus(c_+ \cup c_-) \cup \oplus c_r \cup (c_+ \otimes c_r) \cup (c_- \otimes c_r) \cup (c_r \otimes c_-) \cup (c_r \otimes c_+)\\ &= \oplus c_+ \cup \oplus c_- \cup \oplus c_r \cup (c_+ \otimes c_-) \cup (c_- \otimes c_+) \cup (c_+ \otimes c_r) \cup (c_- \otimes c_r) \cup (c_r \otimes c_-) \cup (c_r \otimes c_+)\end{aligned} \quad \textbf{(5-2)}$$

This equation can be graphically represented in Figure 5-1.

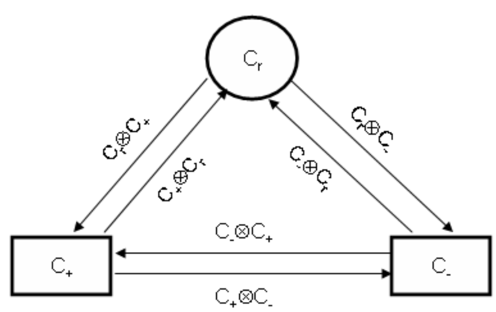


Figure 5-1 Structure of a conflict

According to ATDM, a conflict C can be viewed as a compound object composed of three sub-objects. It is only the special relations between these three sub-objects of object C that makes C a conflict. Among the three elements of a conflict C, there are total 6 relationships, which can be explicitly defined according to the particularity of a conflict. There are also some commonalities that undergo all conflicts independent of the type of the conflict concerning the relations. In the following, we discuss the qualitative relationships among those 6 relations of a conflict. In order to ease the analysis, let us first denote the 6 relations with r_1, r_2, r_3, r_4, r_5, and r_6, respectively.

$r_1 =: c_+ \otimes c_-$

$r_2 =: c_- \otimes c_+$

$r_3 =: c_+ \otimes c_r$

$r_4 =: c_r \otimes c_+$

$r_5 =: c_- \otimes c_r$

$r_6 =: c_r \otimes c_-$

Independent of the type of conflicts, these 6 relations shown in Figure 5-1can be classified into three groups as follows.

Relations $r_1 =: c_+ \otimes c_-$ and $r_2 =: c_- \otimes c_+$

From representation point of view, object c_+ and c_- must be in the same abstract level. In this conflict, they are contending limited resource c_r, so there exists a competitive relationship between c_+ and c_-. Qualitatively, if $\partial c_+ > 0$, then $\partial c_- < 0$, and vice versa.

Relations r_3***=:*** $c_+ \otimes c_r$ ***and*** r_5 ***=:*** $c_- \otimes c_r$

Relations r_3 and r_5 can be defined as that c_+ and c_- require c_r to do something, but c_r cannot satisfy their requirements at the same time. There exists a causal relation between r_3 and r_5. By using qualitative analysis, we have if r_3 = true, then r_5 = false, and vice versa.

Relations r_4***=:*** $c_r \otimes c_+$ ***and*** r_6***=:*** $c_r \otimes c_-$

These two relations can be defined as that the resource object affects both of the competitive objects. In many cases, the influences of resource object to the two competitive objects are not balanced.

In most cases, the relationships in a conflict are not equally important; therefore, we should first identify the principal relation, i.e. the principal aspect of conflict, before we apply resolution methods. In a more challenging manner, there may exist many conflicts for some complex problems and those conflicts may be interconnected. In the conflict shown in Figure 4-18, object c_2 exists in two conflicts. In this case, if we try to solve conflict 1, we need to consider conflict 2, and vice versa. In such a case, it is very important to find the principal conflict to start with.

5.3 Resolution principles of conflicts

For conflicts of case 1, i.e. administrative conflicts, since they lie at the surface of the problem and are just a goal we want to achieve to improve some undesirable situation, there is no need to discuss the resolution. As we know, a conflict happens only when objects c_+ and c_- require limited resource c_r in the same place and at the same time. To resolve a conflict existed in a technical system, a transformation action is needed. The methods used for the resolution depend on the relationships between the three objects in a conflict. Starting from Equation 5-2 and Figure 5-1, we can define the following relations independent of the type of conflicts.

(a) ($c_+ \otimes c_-$) and ($c_- \otimes c_+$): object c_+ and c_- are two subsystems of the same system; they are competing for limited resource c_r. So there exists a competitive relation between c_+ and c_-.

(b) ($c_+ \otimes c_r$) and ($c_- \otimes c_r$): both object c_+ and c_- have requirement for c_r.

(c) ($c_r \otimes c_+$) and ($c_r \otimes c_-$): resource object c_r affects both object c_+ and c_-.

In principle, a conflict can be solved by changing c_r, by removing ($c_+ \otimes c_r$) or ($c_- \otimes c_r$), or compromising between ($c_+ \otimes c_r$) and ($c_- \otimes c_r$), corresponding to different relations, we propose the following six principles for resolving conflicts.

5.3.1 Modify c_r or replace c_r with a new source c_{rn}

Considering the two relations ($c_r \otimes c_i$) and ($c_i \otimes c_r$) (i= -, +) in equation 5-1, we know that the former relation is about how c_r affects c_i, the later about why c_i needs c_r. If we can find a c_{rn}, so that ($c_{rn} \otimes c_+$) = ($c_r \otimes c_+$) or ($c_{rn} \otimes c_-$) = ($c_r \otimes c_-$), the conflict is resolved. For example, in airplane design, some metal components are designed to be heavy for high strength, but the heavier the plane, the more energy is needed. This is a conflict, to solve which, we can consider using advanced composite materials instead of metal, because composite materials can be as strong as metal but much lighter. In this case, c_+ can be regarded as the requirement for strength; c_-, the requirement for energy saving; c_r, the metal component. C_+ requires the metal components heavier in order to be strong; c_- requires the same component light in order to save energy. Obviously, these two requirements contradict each other, and the application of new material (i.e. replacing old source C_r with new source C_{rn}) solves this conflict.

5.3.2 Separate ($c_+ \otimes c_r$) and ($c_- \otimes c_r$) in time

We know that both c_+ and c_- require resource c_r, a conflict emerge only when they require c_r in the same time for different purposes. Otherwise, there will be no conflict. So if we can make the two relations ($c_+ \otimes c_r$) and ($c_- \otimes c_r$) happen in different time, we solve the conflict. The design of flywheel in automobile is an example that solves this type of conflict: it stores energy when accelerating and releases energy when decreasing.

5.3.3 Separate ($c_+ \otimes c_r$) and ($c_- \otimes c_r$) in space

For a technical conflict, c_+ and c_- are from different subsystem, so naturally, ($c_+ \otimes c_r$) and ($c_- \otimes c_r$) has already been separated in space. For a physical conflict, c_+ and c_- initially are from the same subsystem; therefore, if we cannot separate the conflict in time, we need to separate it in space, such as the case of hammer design.

5.3.4 Change c_+ or c_-

Changing c_+ or c_- will lead to the change of relations ($c_+ \otimes c_r$), ($c_- \otimes c_r$), ($c_r \otimes c_+$), or ($c_r \otimes c_-$), and therefore solve the conflict. In the engine power example, there is a conflict between engine power and energy consumption. However, if some means can be applied to increasing the efficiency of the engine, the conflict is solved.

5.3.5 Use optimization method

Another way to resolve a conflict is the trade-off between the two objects that require the same resource. Generally, a conflict represents a subsystem whose function is required by its super-system. We can obtain the optimal effect by balancing the requirements of the two objects c_+ and c_- for resource c_r. when a conflict situation can be quantified, an mathematical optimization technique can be used; for example in beam design, we can formulate the conflict between strength and weight as

$$\begin{aligned}&\min(W(d))\\&subject\ \ to\ \ \sigma_{\max}\leq\sigma_p\end{aligned}$$

5.3.6 Replace the whole conflict

The last method is to replace the system having a conflict with a new sub-system. This may create an inventive design. A new material replaces existing one; new principles or effects are applied to a new structure; both methods can be used to remove a conflict.

5.4 Principal conflict and the principal aspect of a conflict

In a complex design, many conflicts may exist, and furthermore those conflicts may be interrelated. Figure 5-2 shows a complex situation with two interdependent conflicts.

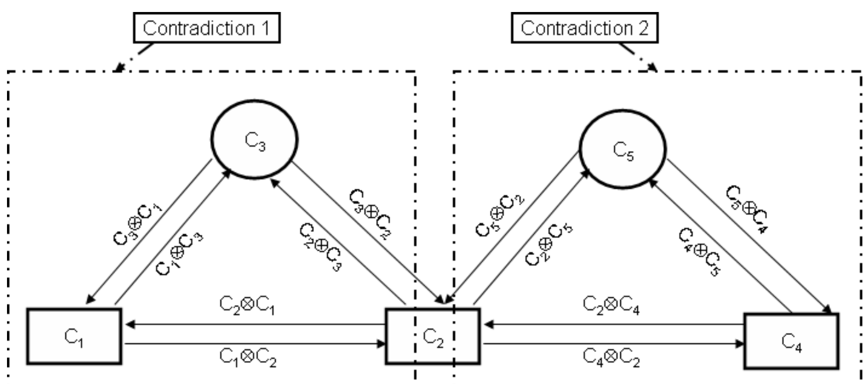


Figure 5-2 Complex conflict

It is noted that object C_2 exists in two conflicts. In this case, if we try to solve conflict 1, we need to consider conflict 2, and vice versa. It is very important to find the principal conflict to start with. Also, the three objects contained in a conflict may be compound objects. In this case, the relations, ($c_+ \otimes c_r$), ($c_- \otimes c_r$), ($c_r \otimes c_+$), or ($c_r \otimes c_-$), need to be decomposed further in order to find the root of the conflict, or the principal aspect of the conflict. In designing a complex technical system, it is unavoidable that a number of conflicts would exist. Since we are limited with resources, time or budget, we must identify the most important conflict among all the conflicts, and find solutions for this conflict at first. In the following, we propose some fundamental rules for identifying principal conflicts and the principal aspect of a conflict.

Rule 1: Laws

Laws include natural laws and social laws. In order to make a technical system to work, first we must obey the natural laws, some of which we have already known very well while still some we have not known yet. Natural laws prescribe both what we can do and what we cannot. When a designer masters a law, it becomes his design knowledge, which will aid him improve design. For instance, before we knew Archimedes' Principle about buoyant force we were not capable of designing complex boat. Meanwhile, we must also follow the man-made laws prescribed for some products. For example, food industry must obey a country's Food Laws.

Rule 2: Regulations

Regulations are similar to social laws but less demanding. Almost all industrial fields have their owner regulations about the product they are making. For example, safety regulations for vehicle brake system, construction codes for building design, etc. Regulations come from the feedback of applications of a technical system category. Regulations prescribe for a certain category of technical systems some fundamental requirements that the system must be met before going to market. Regulations aim at protecting public interest. Therefore, a conflict that involves regulations should be considered as a principal conflict.

Rule 3: Safety (environmental and personal)

For some inventive product design, there is no regulation to follow. In resolving design conflicts, in addition to obeying natural laws, we should analyse the potential effects of the product on environmental and human safety. From product life cycle point of view, this may include the effect of manufacture, use, maintenance, and recycling. For example, some toxic by-product may be produced in manufacturing paper; some material used in vehicle brake pads may cause cancer when using brake; some technical systems such as nuclear power generators may pose a tremendous challenge when they need recycling. All conflicts involving those aspects should be addressed during product design.

Rule 4: Market

All products are made for sale. A product must be better than or different from others to have a good share of market. In early stage of family car production, drivers found they could not drink coffee in their car because there was no place to put the cup. The resolution of this conflict made cars with cup holder very popular and therefore earned extra market for those cars.

Rule 5: Cost

The last but not the least (from business point of view) rule in identifying principal conflict is the cost. This rule is closely related to the Rule 4 because cost is also a market feature. Difference between cost and market lies in that market is more comprehensive and complicated. For any technical system, cost includes manufacturing cost, use and maintenance cost, and recycling cost. Cost is one of the principal facts that cause design conflict. For example, when we choose to use a stronger but more expensive material, there will be a design conflict between quality and cost.

5.5 Discussion of the criterion for selecting resolution strategy

In most cases, there are a number of resolution strategies for a conflict, which have both advantages and disadvantages. This poses a challenge for selecting the best or proper resolution strategy. In design practice, designers usually use matrix method. They first list all solutions with the pros and cons, assign a numeric value to each pro and con, and select a solution with the largest value. TRIZ suggests ideality as a criterion for good design. An ideal product has no mass, no volume, and no energy needed (Savransky, 2000). Obviously, this is an ideal state, and it cannot exist in reality. For practical use, TRIZ authors define ideality as the ratio of useful functions to harmful functions. A solution is better than others if it creates more useful functions and less harmful functions than any other solutions do. In axiomatic design, Suh (1990) presents two axioms to guide design practice. The first axiom states that "maintain the independence of the functional requirements", and the second axiom states that "Minimize the information content of the design". Suh define information as the probability of satisfying a given functional requirement.

Considering we already have a formal structure and six resolution principles, we will propose some criteria by combining the analysis of current literature and this formal structure. Suppose we have a number of resolution strategies, we will ask the following questions for each solution.

1. Has this solution fully resolved the conflict?

2. What is the effect of this solution on the environments, especially on the system that the conflict comes from?
3. Is the solution easy to implement?
4. What are the harmful functions that the solution may incur?

Based on the answers to those questions, we put forward the following rules for selecting resolution method.

Rule 1: The extent to which the solution satisfies the conflict

Rule 2: Cause the least change to the system under discussion

Rule 3: Least complexity

Rule 4: Largest ratios of useful functions to harmful functions

Rule 5: A potential leading to inventive design

Rule 1 is based on the fact that there may be no solutions that can fully resolve the conflict, so we select the solution that can resolve the conflict to the greatest extent. Rule 2 requires that the solution should not cause significant change to the existent system. Rule 3 requires the solution should be as simple as possible. Rule 4 requires the solution cause the least harmful effects. Rule 5 states that the solution that may lead to an inventive design has the top priority.

It should be noted that all rules are only guidance. The best solution can only be achieved by good understanding of the concrete problem itself and sound judgement of the situation with the aid of rules.

Chapter 6
Example: Design Evolution of Brake System

In this chapter, a comprehensive example–the evolution of vehicle brake system design–is introduced for understanding design activities and demonstrating the applications of formal structure of conflict proposed in previous chapters. We first present a brief review of the development of vehicle brake system, and then investigate the driving force for those developments using formal conflict analysis methodology. It shows that design conflict is the driving force of evolution of technical system. The result of resolving a conflict not only improves existing design but also may create brand-new design concepts as well, and those conflicts that are not resolved in a product design may indicate the trend for product evolution.

6.1 A brief review of the development of vehicle brake system

Motor vehicles have served people as the primary means of transportation for several decades. Like power train system and suspension system in a vehicle, brake system is one of the most important and indispensable systems for a motor vehicle. With the development of automotive technology, brake system has undergone great changes in structure, performance, and reliability. In terms of the structure of a brake, modern vehicle brake systems are classified into two types: disk brake system and drum brake system, the development of which has experienced several stages (Wikipedia, 2006).

6.1.1 Early mechanical brake system

The first generation of brake system was pure mechanical, designed for early motor vehicles with steel rimmed wheels. When the driver press the foot pedal or pull the handle, the force is transfer to a curved wood, friction is created between the wood and steel wheel, and therefore the vehicle is slowed down or stopped. Such a brake system consists of curved wooden blocks that act as brake shoes, levers, rods, pivots, and cables. As the earliest vehicle brake system, this kind of mechanical brake system was born together with the first self-propelled road vehicle. It was the ancestor of modern vehicle brake systems and had basic function to stop a vehicle.

6.1.2 Drum brake system

When the steel wheel was replaced with pneumatic rubber ones in 1895, the brake used for steel wheel was not fit for rubber tires because of the poor wear property of rubber tires. A new brake system, called external band brake system, was developed to solve this problem. Instead of exerting friction force directly upon the rubber tire, new devices were designed to apply friction force to the axle or transmission shaft, or to a drum attached to them. A typical external band brake system was made up of drums, metal bands with lining, and a mechanical operation system composed of levers, rods, pivots, and cables etc. With this type brake, more friction could be achieved.

Since the structure of external band brake was open, rain or snow was easy to enter the space between the brake bands and drums. This lowered the brake efficiency. Meanwhile, road dirt easily caused rapid wear of brake band linings. Also, unexpected braking occasionally took place due to drum expansion. Those problems were later solved by introducing internal expanding shoe brakes because the closed structure of the new brakes protected brake shoes from weather and dust.

6.1.3 Hydraulic and power-assistant brake system

For a long time in the early stage of automotive industry, all brakes had been operated manually through a mechanical systems consisting of levers, rods, pivots and cables. Such a brake system had some obvious disadvantages. First, the linkage system was not reliable enough; second, the brake pressure could not be equalized on the wheels; third, the brake system required much human efforts, which would easily fatigue the driver. Those problems were well solved by the introduction of the hydraulic system, where fluid power was used to transfer the force applied to the brake pedal. Hydraulic brake systems are based on two properties of liquid: liquid is incompressible, and pressure is exerted equally in all directions in a closed system. Also, a hydraulic system can be used to increase or decrease force or motion. All the cylinders and pipe lines form one closed hydraulic system filled with brake fluid. The hydraulic brake system also greatly simplified the job of providing balanced four-wheel braking, since the pressure is transmitted from the master cylinder to the pistons of all wheel cylinders equally without pressure loss and time delay. As a result, pressures are applied to all brake shoes identically and simultaneously. By changing the area of master cylinder piston and wheel cylinders, hydraulics also reduced the required effort amount applied to the pedal for braking, which resulted in easier braking and safer driving.

To make full use of the advantage of hydraulic, another new innovation, power-assistance, was employed in hydraulic braking system. Power brakes use a booster, usually engine vacuum or hydraulic pressure, to assist brake pedal application, which greatly reduce the effort amount required for braking, and therefore made braking easier and driving safer. Today, power brakes have already been standard equipment on nearly all motor vehicles.

6.1.4 Disc brake system

Although drum brake is better than the open structure brake, its defects showed more and more apparent as motor vehicles became faster and heavier. First, its ability of dissipating heat is low. When applying brake, a tremendous amount of mechanical energy is transformed into heat, but the enclosed structure leads to slow dissipation of heat, which causes fast fade of brake shoes. Another defect of drums is that it traps water when the vehicle passes through a puddle. The water trapped in the drums lower braking efficiency. A solution to those problems is to replace drums with discs. First, disc brakes have high cooling ability. Second, the open structure does not trap water any more. Thirdly, self-adjusting ability of disc brakes simplifies the design and eliminates periodic adjustment required by drum brakes. Though having better performance, disc brakes are more expensive than drum brakes. To balance the cost and performance, some modern cars use disks for front brakes and drums for rear brakes, because the front brakes contribute 60-70% braking force for the vehicle while the rear ones only 30-40%.

6.1.5 Antilock and dynamics control brake system

When using excessive braking force, the wheel will be locked. This lock-up greatly lower the brake efficiency and make the steering system loss function. This is a life-threatening situation. The invention of Antilock Brake System (ABS) and Vehicle Dynamics Control (VDC) solves these problems. Speed sensors constantly monitor the speed of all wheels and send information to the Electronic Control Unit (ECU), and the ABS will rapidly lower the brake pressure whenever the system detects a lock-up. By minimizing wheel lockup and skidding, ABS allows drivers to maintain steering control when braking hard, and keep the vehicle move straight and stable to avoid the chance of getting sideways during a panic stop. Antilock brake system can greatly shorten the brake distance and improve the vehicle stability.

Vehicle Dynamics Control (VDC) system combines the basic functions of ABS with the steering control to improve vehicle stability. A steering angle sensor detects the driver's steering inputs; and yaw rate sensor, the core of the VDC system, monitors any change in motion of the vehicle's vertical axis to detect under- steering, over-steering, or fishtailing. With information provided, the VDC control module can judge if the vehicle is responding normally. Since VDC deals more steering than braking, we will not discuss it in this research.

6.2 Analysis of the design revolution of brake system

The brief review of brake history indicates that modern brake system has been evolving over the years to meet different requirements. In this section, we will analyze the different design based on conflict analysis.

6.2.1 Environment analysis for designing brake

To drive safely, a driver must be able to fully control the direction and the speed of a vehicle. While the direction is controlled through steering system, the speed is controlled with brake system. By simply analyzing the status of a vehicle on road, we can easily define three basic functions for a modern vehicle brake system: 1) to decelerate vehicle, including stopping, 2) to maintain vehicle speed during downhill operation, and 3) to hold vehicle stationary on a grade. So the brake design problem for a vehicle can be written in textual form as follows.

> *To design a brake that can fulfill the three basic functions mentioned above and meet current regulations so a driver can maneuver the vehicle safly on any roads and under any wheher conditions.*

Using ROM analysis, we can generate the corresponding ROM diagram as shown in Figure 6-1.

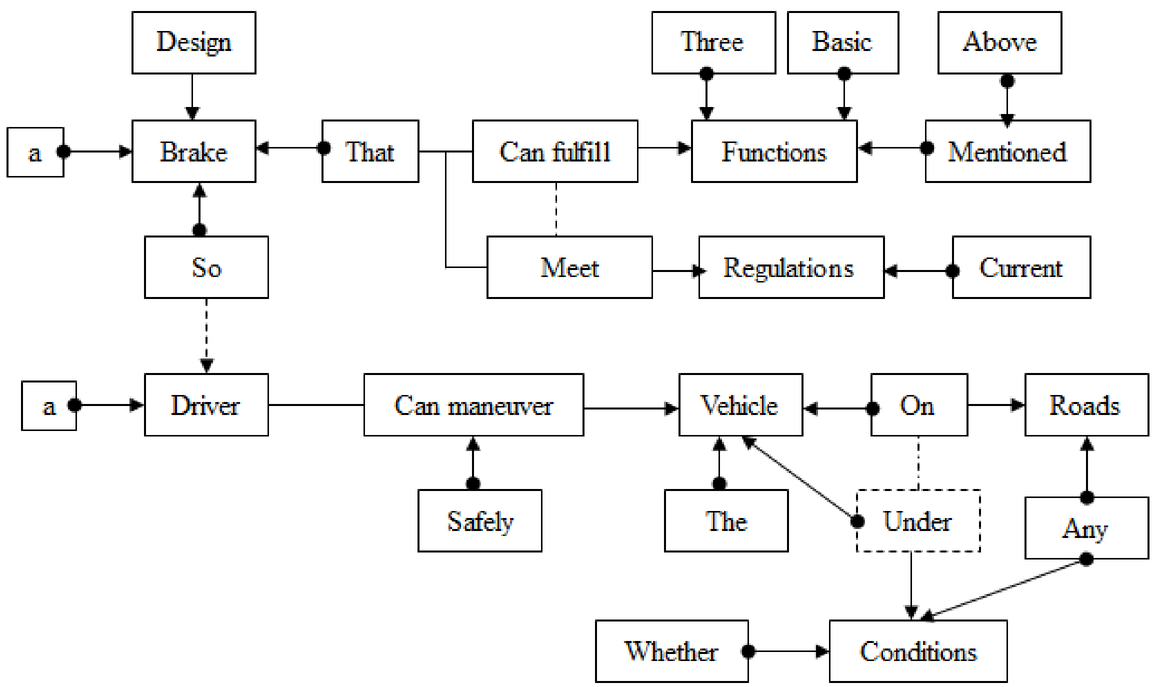


Figure 6-1 ROM description of brake design problem

From this diagram, we identify some important objects: *brake*, *functions*, *regulations*, *driver*, *vehicle*, *roads*, and *whether*. By applying the question-asking technique proposed by Wang and Zeng (2009), we may ask the following questions concerning the brake working environment and therefore requirements.

1. What is the life cycle of the brake?

2. What are the environment elements?
3. How are those elements related?
4. How does the brake affect its environment and vice versa?

For a vehicle brake, the whole lifecycle mainly includes design, testing, manufacture, installation, use, Maintenance, and recycling. Like any other engineering products, a brake's working environment includes natural environment, built environment, and human environment. All of the three types of environment are composed of different elements in different stage of the brake lifecycle. Generally speaking, anything that affects the brake system or is affected by the system is considered as a relevant environment element. The ROM diagram has revealed some obvious environment elements: regulations, driver, vehicle, roads, and whether. Since the brake is a part of a vehicle, which is composed of body, exterior, electrical and electronic, interior, powertrain and chassis, and miscellaneous parts, the first environment element to be considered in brake design shall be the vehicle. Moreover, of all the vehicle components, powertrain and chassis, of which brake is a part, has a direct influence on brake design. Regulations, which belong to Built Environment, may include national regulations, industrial standards, and company standards (Yu, 1990), such as Federal Motor Vehicle Safety Standards (FMVSS), European Economic Community (EEC) regulations, Japan Brake Regulations (JASO), etc. All brake systems must respect those regulations and meet certain standards. Another environment element is driver, who controls the vehicle speed by using the brake. Road is another environment shown in the ROM diagram Figure 6-1. It relates brake maily via wheel and tire system. Whether, or climate, is also an important environment that must be taken into consideration in designing brake system. All the environment elements mentioned above are obtained from ROM analysis, but there are some other elements that can be better identified through lifecycle analysis. For the sake of simplicity, we list all the environment elements in Table 6-1.

Table 6-1 Environment elements

	Natural environments	Built environments	Human environment
Design stage	Natural laws, such as Newtonian laws, thermal laws, etc.	All means or tools used in designing, other components	Designer User
Testing stage	Test conditions, whether, temperature, etc.	Testing equipment, testing methods and procedures, etc.	Testing engineer, mechanic, etc.
Manufacture		Manufacture technique,	Mechanic, machinist, process engineer, etc
Installation		Equipment, other vehicle components, the structure and dimention of the vehicle	mechanic
Use			Driver
Maintenance		Relevent regulations,	Mechanic, engineer
Recycle	Natural environment related to recycle	Recycle, technique, policies, regulations,	People…

6.2.2 Conflict identification

After identifying all relevant environments, we need to analyze the relations between product and its environments, and the relations between environment elements. By doing so, we can identify design conflicts. Of the three categories of product environments, natural environment should be considered first by analyzing relations between brake system and its natural environment element. Natural environment can be divided into two groups: the physical nature and the laws that govern or describe this nature. While the variety of physical nature is infinity, the laws known to human being are very limited. For any design, the product must respect all relevant natural laws in the first place; otherwise, it cannot exist. Second, the product must be fit for the natural environment and cannot have harmful effects on the physical nature during its lifecycle. In the meantime, some product design may incite a need to discover new natural laws and effects.

Any new product must be compatible with its built environment. Being one of the safety components in a vehicle, brake system must respect all relevant regulations and standards before going to market. Those regulations and standards usually prescribe the minimum requirements on brake's performance and environmental influence. For example, FMVSS 135 prohibits locking of the rear brakes before the front brakes. Other vehicle components that affect brake design may include structural requirements, the weight of vehicle, the center and distribution of the weight, etc.

Road conditions play an important role in brake design because the traction between road and tires determines the maximum brake force. Besides, water or dust on the road affects the performance of a brake. Manufacture is another important factor that needs to be considered during design process. Generally speaking, we must design something that can be manufactured using current technology though there are some cases that a new design may drive research on new manufacture technology. Any product needs to be tested before going to market; therefore how to test also must be considered. The test may include what functions or parameters need to be tested and the procedures to perform the test. The last two factors needed considering are maintenance and recycling from product life cycle point of view. All those factors put together, some conflict will emerge. Table 6-2 lists some conflicts identified.

Table 6-2 Conflicts identified

Categories of requirements		Description of Conflicts to be solved	
Natural	Thermal analysis	Heat deteriorate the performance of brake	Safety requires constant performance
Built	regulation	Fast vehicle speed	Small stopping distance
		High performance materials	Tendency of environment pollution
	Other components	The size of a vehicle is predefined	The brake needs certain room to be installed
		Steering	Lock of front brakes disable steering
		Stability	Lock of rear brakes lower the stability
		Requiring vehicle move straight while braking	Unequal brake force between left and right brake leads to brake deviation
human		Driver's force is limited	Need large brake force

6.2.3 Analysis of the design revolution of different brake systems

The environment and conflict analyzed above are based on current environment, though some of them should have been considered in the very beginning stage of brake development. If we compare today's brake system and the first-generation brake, we will find a significant difference between them. They differ in the numbers of components, in materials, in technologies used on brakes, etc. The major driving forces for this revolution are the conflicts between brake system and environment or between different environments, and the way we resolve those conflicts. This section demonstrates how those conflicts were resolved at each stage of brake development history.

6.2.3.1 Early mechanical brake systems

The early mechanical brake system was designed for motor vehicles that was powered by steam engine and used steel-rimmed wheels. It worked almost in the same way as those on the horse-drawn carriages. A typical early mechanical brake is shown in Figure 6-2.

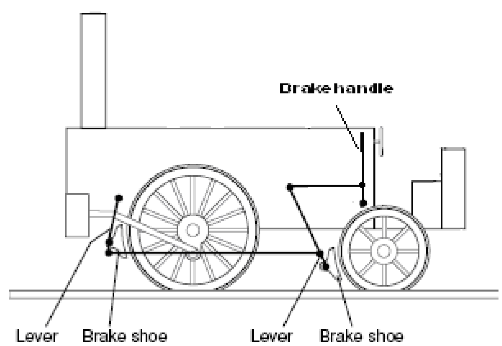


Figure 6-2 A typical mechanical brake system (Hasegawa & Uchida, 1999)

The first environment element is always the structure of the vehicle itself for any brake must be designed to fit the vehicle structure first.

The second environment that affects the brake most is the speed of the vehicle. From the history of automotive, we know that the speed of a steam-powered vehicle is usually less than 20km/h. For example, in 1865 the *Locomotives Act* of Great Britain limited the speed to 4 mph (6.4 km/h) in the country and just 2 mph (3.2 km/h) in towns and cities. If the initial velocity is V, the deceleration is a, the brake distance is $V^2/(2a)$ when the vehicle comes to full stop. Suppose we require the deceleration is about $2m/s^2$, the brake distance would be around 8 meters when the initial speed is 20km/h.

The third environment considered in designing brake is the wheels, which were made with steel. This fact made it possible to exert force directly onto the steel wheels. A simplified dynamic analysis for a wheel with angle velocity $w > 0$ is as follows.

$$T_f - F_b r_e = 0,$$

$$F_f = \frac{T_f}{r_e} = N\mu$$

$$a = \frac{F_b}{m} \qquad \textbf{(6-1)}$$

Where, F_b is the traction between road and wheel; N represents the normal force needed on the brake; m represents the equivalent mass of the vehicle; and μ is the friction coefficient between brake shoe and wheel. If we are given the values m, r_e, and μ, and require deceleration a, we can obtain the force N needed on the brake shoe. This force can be easily achieved by using a lever system, which, together with brake shoes and vehicle steel wheels, constitutes the very early mechanical brake system.

In summary, the environment elements considered in this design stage included only steel wheels, vehicle structure, and speed.

6.2.3.2 Drum brake system

When rubber tire replaced steel wheel, a new conflict occurred: the tire could not directly bear against brake shoe (a piece of wood in the mechanical brake). This conflict was resolved by braking on axle or shaft instead on rubber tires, or attaching an extra component to tires. This resolution of the conflict between the rubber tire and brake shoe led to a new brake system, called external band brake system.

Figure 6-3 shows a typical structure of external band system. Compared with the mechanical brake used for steel-rimmed wheel, the fundamental change of the external band system is that the brake itself became a fully independent component of a vehicle. Moreover, the structure became more complex, and more components were used in this new system. It can be seen that the major driving force of this development from mechanical brake to this external drum brake is the new conflict between rubber tire and brake, which was caused by the change of environment element.

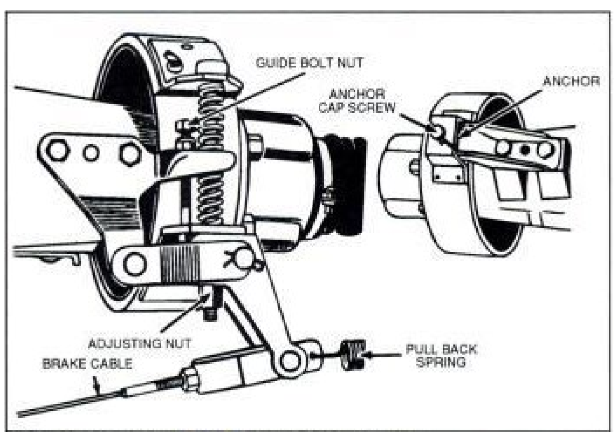


Figure 6-3 External band brake (Carley & Mavrigian, 1998)

This new drum brake was all external, open to its entire natural environment components; such as dust, air, water, a feature which soon turned into a problem. As the vehicle became faster, divers used brake more often when driving. Two serious problems emerged because of dust, water, and air etc. The first is the braking efficiency was lowered, which increase the safety risk; the second is that the distance between two overhauls of brake became shorter, which increased the cost for maintenance. This posed a new design conflict: the brake requires a high efficiency and long distance between two overhauls but the environment element water, dust and air reduce them.

Table 6-3 lists all symbols used in this sub-design problem. Figure 6-4 demonstrates the relations among all those elements.

Table 6-3 Symbols used in the design

Name	symbol
Shoe and drum	b
air	a
dust	d
water	w
efficiency	e
Distance between overhauls	m

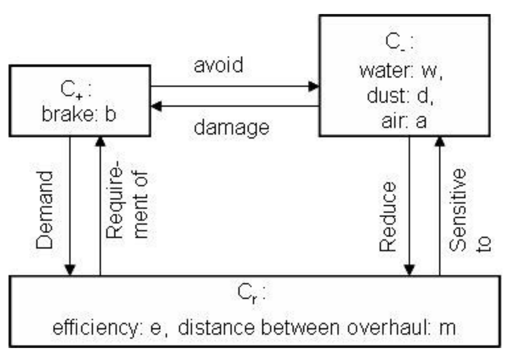


Figure 6-4 Structure of conflict

Using the EBD and conflict analysis method, we identified that this design conflict, donated by *C*, is composed of components *b*, *a*, *d*, *w*, *e*, and *m*. And mathematically, the structure can be written as

$$\begin{aligned} \oplus C &= \oplus(b \cup a \cup d \cup w \cup e \cup m) = \oplus(c_+ \cup c_- \cup c_r) \\ &= \oplus c_+ \cup \oplus c_- \cup \oplus c_r \cup R \\ where,\quad & c_+ = b; \quad c_- = a \cup d \cup w; \quad c_r = e \cup m \\ R &= \bigcup_{i,j=:+,-,r;i\neq j} c_i \otimes c_j \end{aligned} \qquad \textbf{(6-2)}$$

In previous chapters, we have pointed out that a conflict can be resolved by changing any of its components and therefore changing the relations between components. Based on this principle, to solve the current brake design conflict, we can change C_+, C_-, or C_r theoretically. In practice, however, we cannot change environment but we want high efficiency and long distance between overhauls. This leaves us with only two choices: providing a means such that water and dust cannot enter the brake, or changing the structure of brake so that water and dust have least effect on the brake. And indeed, in the history of vehicle development the first choice led to the design of internal drum brake, while the second choice led to a brand-new brake design, disc brake, which will be discussed in later sections.

The internal drum brake is an intuitive choice for improving the efficiency and lengthening the distance between two overhauls. Since the dust and water could easily get into the external drum brake, we could use a cover to stop them from entering the brake. Compared to external drum brake design, the internal drum brake take into consideration more environment elements, including dust, air, water and the speed increase of the vehicle.

6.2.3.3 Hydraulic and power-assistant brake system

For any vehicle, it is always the driver, also one of the environment components, who provides the input of brake application process. Given that the structure of brake is determined and the road traction is enough, the deceleration and brake distance greatly depend on the force that the driver can exert on the brake. In the early mechanical and drum brake systems, the force was transmitted through a pure linkage system, whose mechanism can basically be simplified as in Figure 6-5.

Figure 6-5 Mechanism of linkage

Assume the lever ration is *r*, we have

$$F_{out} = F_{in} \times r \qquad D_{out} = \frac{D_{in}}{r} \qquad \textbf{(6-3)}$$

For any given vehicle, the value *r* is constrained by its structure. Since the force provided by driver is very limited, if we want the vehicle to decelerate fast the pure linkage system would fail the purpose. Besides, constant use of brake can easily fatigue drivers. Therefore a conflict between limited human capability of providing force and requirement for a large output force emerged.

Another problem, as already explained in section 6.1.3, is that linkage system was neither reliable enough nor capable of distributing equal pressure on the four wheels. When trying to decelerate a vehicle at very high speed, the vehicle will deviate if the brake forces on four wheels are not equal. This is a serious safety risk. These problems could be solved using a hydraulic system, where fluid is used to transfer the force applied to the brake pedal. Hydraulic system bases on three properties of liquid: 1) liquid is incompressible, 2) pressures in all direction are equal, and 3) liquid can increase or decrease force and motion, like a lever. Property 2 makes it easier to distribute equal pressure on the four wheels, while property 3 provides a larger force amplification ratio *r* compared with linkage system, and therefore reduce the driver's effort when applying brake. Also, hydraulic system is simpler in structure than linkage system, which renders hydraulic system more reliable. However, when the vehicle is faster and heavier, the ration r is still too low to create enough force without increasing driver's efforts, such as a heavy-duty truck. So the conflict between limited human capability of providing force and requirement for a large output force emerge again. Power brake booster is a solution for this. Booster creates force using vacuum produced by engine as a by-product.

6.2.3.4 Disc brake system

In section 6.2.3.2, we have pointed out that the conflict between external brake performance and the environments can be resolved using two strategies: keep water and dust away from the brake using a cover, which led to the design of internal drum brake; or design a new brake that is not sensitive to dust and water, which led to the design of disc brake. The application of internal hydraulic brake, which is more effective than external brake, resolved the conflict between performance and environment components dust, water and air. However, a new conflict arises: the closed structure makes the dissipation of heat very ineffective. High temperature significantly reduces the friction coefficient between the brake shoe and drum. In addition, the closed structure may trap water, which lowers the effectiveness of brake. Similar to Figure 6-4, the diagram of this new conflict can be constructed, shown in Figure 6-6.

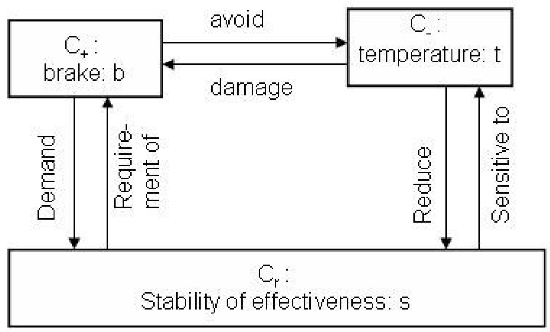


Figure 6-6 New conflict in external drum brake

Mathematically, the conflict is written as

$$\begin{aligned} \oplus C &= \oplus(b \cup t \cup s) = \oplus(c_+ \cup c_- \cup c_r) \\ &= \oplus c_+ \cup \oplus c_- \cup \oplus c_r \cup R \\ where, &\quad c_+ = b; \quad c_- = t; \quad c_r = s \\ R &= \bigcup_{i,j=:+,-,r;i \neq j} c_i \otimes c_j \end{aligned} \qquad \textbf{(6-4)}$$

By analyzing the relations between the conflict components, we can resolve this conflict in two ways. First, use new materials for brake shoe and drum that have better thermal stability; secondly, redesign the structure of drum brake. Along the first path, many new materials such as ceramic and semi-metal are used in brake shoe, which have greatly improved the thermal stability of brake. Following the second way, engineers have presented a brand-new design, disc brake. The application of disc brake resolves the conflicts shown in Figure 6-6.

6.2.3.5 Antilock and dynamic control brake system

All analysis introduced above is based on the assumption that the road provides enough traction. However, in reality, when braking too hard or driving on icy road, the brake force may exceed the traction, and therefore sliding occurs, which greatly reduces the friction between tire and road surface. As a result, the stop distance will increase. This situation indicates a conflict: on the one hand, in order to stop a vehicle fast, we need applying enough force; on the other hand, if the force is too large sliding may occur, and therefore stop distance increases. The three objects of this conflict can be identified as follows.

c_r: braking system

c_+: larger braking force

c_- : smaller braking force

Imagine you are driving a car without ABS on an icy road; when you need to apply brake, you are not able to determine the exact force you need. In this case, because c_+ and c_- cannot be changed, this conflict must be solved by using resolution rule 5.3.1, i.e. to modify or replace c_r. Adding an ABS to the old brake system is an application of rule 5.3.1, and therefore solved this conflict.

6.2.3.6 The trend of next generation of brake system

Analysis from the above sections shows that all types of brakes, from simple pure mechanical brake to advanced hydraulic disc brake, control the vehicle speed using friction. Considering the vehicle speed changes from V_1 to V_2 and disregarding other factors, the energy change can be roughly calculated as follows.

$$\Delta E = \frac{1}{2} m(V_1^2 - V_2^2) \qquad \textbf{(6-5)}$$

This indicates that every time when brake is applied, energy of amount ΔE will be lost in the form of heat. The magnitude of ΔE can be understood better by looking at a car's specification. Take Subaru 2.0T as an example, the specification is 32/24 MPG, which indicates that local drive needs about 30% more gas than a highway drive due to the frequent use of brakes. This creates a conflict situation: on the one hand, we need sufficient friction to lower the vehicle speed; on the other hand, we do not want to waste energy. Let c_r stand for the resource object *brake*; c_+ stands for the action *providing brake force*; c_- stand for the action *wasting thermal energy*, we can formally represent this conflict as

$$C = c_+ \cup c_- \cup c_r$$

This conflict can also be described in Figure 6-7.

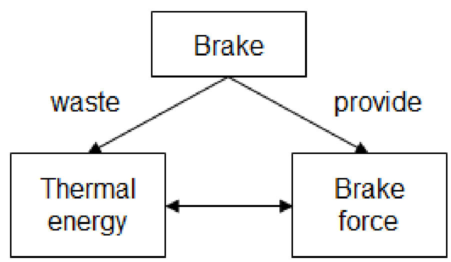


Figure 6-7 A conflict in current vehicle brake design

Comparing Figure 6-7 with Figure 4-5, we notice that this conflict is due to one object (brake) predicates two incompatible actions in the same time. To solve this conflict, we may use resolution principle 5.3.1, modify c_r or replace c_r with a new source c_{rn}. Based on this principle, two ideas can be investigated. First, change the current brake structure or material so the thermal energy can be stored; second, invent a brand-new structure so there is no heat created when using brake.

6.3 Summary

In this chapter, we first reviewed the design development of vehicle brake system, and then we analyzed the evolution process based on environment analysis, conflict identification, and conflict resolution. This analysis, which is based on the formal structure of conflict, demonstrates that conflict is the driving force for product evolution, that the origin of conflicts is the product environment relations, and that the conflict resolution leads to new product design. It is also shown that those conflicts that are not resolved at current stage may indicate a trend of product development.

Chapter 7
Conclusions and Future Directions

7.1 Technical systems

Theory of technical system is a crucial part of any design theory. And representation is the core of technical system. Though there are many representation methods for a technical system such as graphical and textual representation, the formal approach presented in this current research has been demonstrated its uniqueness. First, it can represent any technical system in conceptual design stage; second, it provides logical operation that can perform system integration and decomposition. Third, it is the foundation for formal conflict model.

A proper set of fundamental concepts is indispensable to model a system. In this research five concepts are identified based on the current literature review and analysis. These five concepts are purpose, function, behaviour, structure and state, from which all other concepts related to technical system can be logically derived. Compared with other researchers' definitions, definitions in this research have some semantic differences. First, like a system's structure, the function and behaviour also have a hierarchical structure. Second, the definitions of purpose and function are relative. This relativeness implies that a function of upper level may be a purpose of next hierarchical level. Third, function is objective, not subjective. It originates from the relation between the environment and the technical system to be designed. A clear understanding of the objectiveness of system functions system helps designers focus on the environment analysis when modeling function during conceptual design stage. Otherwise, the designer will focus on their own knowledge and therefore limit their creativity and ignore critical environment elements.

Any representation has advantages and disadvantages. To evaluate a representation, five requirements for an ideal representation are proposed in this research. The five requirements are completeness, clarity, independence, flexibility, and adaptability. Completeness requires that all information about the system must be included in the representation. Clarity requires that a clear and simple representations help understand the system; meanwhile clarity must imply uniqueness of interpretation. Independence requires that the representation does not depend on any technical fields. Flexibility requires that the representation should be able to describe systems in different design states. And adaptability requires that the representation is open to updating the system.

The scope and classification of technical systems is discussed based on the assumption that any man-made object needs a design before it is made. All human-made technical systems are classified into three groups: engineering objects, organizational objects, and software.

Two major relations, causal and constructional, are defined. Causal relation is such a relation between two objects that if one changes the other will change correspondingly. A constructional relation between two objects is such a relation that the two objects both belong to the same upper system, and they are not causal related but are needed to work together so that a system can function properly.

7.2 Design conflicts

Design conflicts drive the development of technical system and the evolution of design process. Representation of design conflict is a crucial part of design theory. This current research shows that any atomic conflict, as a special object, is composed of three elements, among which two elements compete the third one. Based on this conjecture, we propose a formal conflict structure, which can be applied to all conflict situations for its formality. The diverse examples presented in this research demonstrate various applications of this formal structure. There are two types of conflicts: simple conflict and complex conflicts. Simple conflicts can also be divided into atomic and compound conflicts.

Based on the formal structure of conflict and the analysis of relations among the elements in a conflict, six general resolution principles are put forward. The principles include modifying the resource object, separating the two competing relations in time or in space, changing the two competing objects, using mathematical optimization, and replacing the entire conflict. Because of the complexity of modern technical system, many conflicts may emerge in the same time. To allocate design resources efficiently, we need to identify the principal conflict among the many existing conflicts in terms of the proposed basic rules, which include laws, regulations, environment or personal safety, market, and cost.

The research also discusses the criteria for selecting a resolution strategy. According to the six resolution methods and the analysis of the current literature, we present five rules for selecting resolution strategies. The rules serve guidance for a designer’s decision-making.

7.3 Suggestions for future research

In summary, this thesis has contributed to design research in the following aspects.

1. Presented a formal model of technical systems
2. Defined a set of fundamental concepts applied to technical systems
3. Put forward a formal structure of design conflict
4. Proposed general conflict resolution strategies
5. Suggested rules for selecting resolution strategies
6. Demonstrated the applications using an comprehensive design example

Though this thesis has efficiently addressed the issues put forward in Chapter 1, there are some topics worth further investigation. Those topics are listed as follows.

1. Finesse the definition of the fundamental concepts of technical systems.
2. Study the relations exiting in human-made technical systems.
3. Investigate the possibility of applying the formal structure of conflict to other fields; such as project management, social science, or political science.
4. Apply the formal structure to the development of automatic conflict management system by integrating with TRIZ, or other strategies.

References

Eckersley, M. (1988). The form of design processes: a protocol analysis study. *Design Studies, 9*(2), 86-94.

Fishwick, P. (1995). *Simulation Model Design and Execution: Building Digital Worlds.* Upper Saddle River, NJ, USA: Prentice Hall.

Hatchuel , A., & Weil, B. (2003). An new approach of innovation design: An introduction to C-K theory. *International conference on Engineering Design, ICED.* Stockholm August 19-21.

Altshuller, G. S. (1984). *Creativity as an Exact Science: The Theory of the Solutions of Inventive Problems.* Taylor & Francis.

American Systems Corporation. (2006). *System Modeling Language Specifications*. Retrieved from http://www.sysml.org/docs/specs/SysML-v1-Draft-06-03-01.pdf

Braha, D., & Maimon, O. (1998). *A Mathematical Theory of Design.* Kluwer Academic Publishers. Norwell, MA, USA.

Britannica, E. (2007). Encyclopædia Britannica online. Retrieved from http://www.britannica.com.

Broadbent, G. (1966). Creativity. In S. Gregory (Ed.), *The Design Methods* (pp. 111-119). London: Butterworths.

Broenink, J. (1999). Object-oriented modelling with bond graphs and modica. *International conference on Bond Graph Modeling ICBGM'99, Simulation Series*, *vol 31 no.1*, pp. 163-168.

Carley, L., & Mavrigian, M. (1998). *Brake Systems.* New York: HP Books.

Coombs, C., & Avrunin, G. (1988). *The Structure of Conflict.* Lawrence Erlbaum Associates, Inc. Hillsdale, New Jersy USA.

Cross, N. (1989). *Engineering Design Methods.* Chuchester: John Wiley and Sons Ltd.

Cross, N., Dorst, K., & Roozenburg, N. (1992). *Research in Design Thinking.* Delft Univesity Press.

Dan. (2007). *Drum Brakes*. Retrieved from www.dansmc.com/drumbrakes.htm.

Daniela, E., Damian, H., Jonker, C., Treur, J., & Wijngaards, N. (2005). Integration of behahioural requirements specification with compositional knowledge engineering. *Knowledge-Based Systems, 18*(7), 353-365.

Dorst, K., & Cross, N. (2001). Creativity in the design process: co-evolution of problem-solution. *Design Studies,, 22*(5), 425-437.

Eder, W. (1966). Definitions and Methodologies. In S. Gregory (Ed.), *The Design Methods* (pp. 19-31). London: Butterworths.

Eder, W. (1989). Information systems for designers. In *International Conference on Engineering Design 1989: proceedings of the I. Mech. E., Harrogate, UK,* (pp. 1307-1320). Bury St Edmunds: Mechanical Engineering Publications Ltd.

English, H. B., & English, A. C. (1958). *A comprehensive dictionary of psychologycal and psychoanalytic terms.* New York: Longmans, Green.

Erden, M., Komoto, H., vanBeek, T., D'Amelio, V., Echavarria, E., & Tomiyama, T. (2008). A review of function modeling: approaches and application. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 22*(2), 147-169.

Fernandez, M. G. (2005). *A framework for agile collaboration in engineering.* Diss. Georgia Institute of Technology: United States -- Georgia.

Gero, J., & Maher, M. (1997). *A framework for research in computing*. Retrieved from http:// www.arch.usyd.edu.au/john/publications.

Goel, A., & Bhatta, S. (2004). Use of design pattern in analogy-based design. *Advanced Engineering Informatics, 18*, 85-94.

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.

Hanna, R., & Barber, T. (2001). An inquiry into computers in design: attitudes before-attitudes after. *Design Studies, 22*(3), 255-281.

Harrinton, J. V., Soltan, H., & Forskitt, M. (1995). Negotiation in knowledge-based concurrent engineering design environment. *Expert Systems, 12*(2), 139-147.

Hasegawa, I., & Uchida, S. (1999). Railway Technology Today 7-Brake Systems. *Japan Railway & Transport Review.*

Hawking, S., & Mlodinow, L. (2008). *A briefer History of Time.* New York: Mantem Dell.

Hazelrigg, G. (1996). *Systems Engineering: an Approach to Information-Based Design.* Prentice Hall.

Hongo, K. (1985). On the significance of the Theory of Design. In H. Yoshikawa (Ed.), *International Symposium on Design and Synthesis* (pp. 237-245). Amsterdam: Elsevier Science Publishers B. V.

Horváth, I. (2004). A treatise on order in engineering design research. *Research in Engineering design, 15*(3), 155-181.

http://www.informatics.susx.ac.uk/books/computers-and-thought/gloss/nodel.html. (n.d.).

Hubka, V. (1985). Attempts and possibilities for a rationalisation of engieering design. In H. Yoshikawa (Ed.), *International Symposium on Design and Synthesis* (pp. 133-138). Amsterdam: Elsevier Science Publishers B. V.

Hubka, V., & Eder, W. (1988). *Theory of Technical Systems.* Springer-Verlag.

Hubka, V., & Eder, W. (1996). *Design Science.* London: springer-verlag.

Ibrahim, R., & Siow, Y. Y. (2011). A Formal Model for Data Flow Diagram Rules. *ARPN Journal of Systems and Software, 1*(2).

Jin, Y., & Chuslip, P. (2006). Study of mental iteration in different design situations. *Design Studies, 27*(1), 25-55.

Jonathan R. A. Maier, G. M. (2009). Affordance based design: a relational theory for design. *Res Eng Design*(20), 13-27.

Jones, J. (1966). Design Methods Reviewed. In S. Gregory (Ed.), *The Design Method* (pp. 295-310). London: Butterworths.

José, A., & Velasquez, A. (2003). Conflict detection and resolution in distributed design. *Production Planning & Control, 14*(8), 734-742.

Kaplan, C., & Simon, H. (1990). In search of Insight. *Cognitive Psychology*(22), 374-419.

Klein, M. (1990). Conflict Resolution in Cooperation Design . *in Computer Science*. University of Illinois at Urbana-Champain: Urbana.

Klein, M. (1993). Support Conflict management in Cooperative Design Teams. *Group Decision Negotiation, 2*, 259-278.

Klein, M. (2000). Towards a Systematic Repository of Knowledge about Managing Collaborative Design Conflicts. *in Sixth International Conference on Artificial Intelligence in Design.* Worcester MA.

Klein, M. & Lu, S. C.(1989). Conflict resolution in cooperative design. *Artificial Intelligence in Engineering, 4* (4), 168-180.

Kossiakoff, A., Sweet, W. N., Seymour, S., & Stev, A. (2011). *Systems Engineering Principles and Practice.* John Wiley & Sons.

Ku, D. C., & DeMicheli, G. (1991). Constrained resource sharing and conflict resolution in Hebe. *Integration, the VLSI Journal, 12*(2), 131-165.

Kumar, B., & Gero, J. S. (1993). Designing for conflicting constraints. *Computing Systems in Engineering, 4*(1), 75-86.

Lara Gracia, M. A. (1999). *Conflict resolution in collabrative facility design.* Diss. Purdue University, Indiana.

Lara, M., & Nof, S. (2003). Computer-supported conflict resolution for collaborative facility designers. *International of Production Research, 41*(2), 207-233.

Larkin, J., & Simon, H. (1987). Why a Diagram is (sometimes)Worth Ten Thousand Words. *Cognitive Science, 11*, 65-99.

Lottaz, C., Smith, I. F., Robert-Nicoud, Y., & Faltings, B. V. (2000). Constraint-based support for negotiation in collaborative design. *Artificial Intelligence in Engineering, 14*(3), 261-280.

Maher, M., & Tang, H. (2003). Co-evolution as a computational and cognitive model of design. *Research in Engineering Design, 14*, 47-63.

Merriam-Webster. (2007). Merria-Webster online dictionary. Retrieved from http://www.merriam-webster.com.

Middendorf, W. H. (1997). *design of devices and systems.* Marcel Dekker Inc.

Myerson, R. B. (1991). *Game Theory: Analysis of Conflict.* Harvard University Press.

Object Management Group. (2008). *OMG Systems Modeling Language*. Retrieved from http://www.uml-sysml.org/documentation/sysml-tutorial-incose-2.2

Pahl, G., & Beitz, W. (2007). *Engineering Design: a Systematic Approach* (third ed.). (K. Wallace, & L. Blessing, Trans.) Springer-Verlag.

Rittel, H., & Webber, M. (1973). Dilemmas in a general theory of planning. *Policy Science, 4*, 155-169.

Ross, S., Fang, L., & Hipel, K. W. (2002). A case-based reasoning system for conflict resolution: design and implementation. *Engineering Applications of Artificial Intelligence, 15*(3-4), 369-383.

Salustri, F., & Venter, R. (1992). An Axiomatic Theory of Engineering Design Information. *Engineering with Computers, 8*(4), 197-211.

Savransky, S. (2000). *Engineering of Creativity.* Boca Raton, Florida: CRC Press.

Simon, H. (1996). *The sciences of the Artificial* (Third ed ed.). Cambridge, Massachusetts: The MIT Press.

Sreeram, R. T. (2000, print). Resolving conflicts in agent-supported collaborative product development. Diss. University of Bath (United Kingdom): United Kingdom.

Stauffer, L., & Ullman, D. (1991). Fundamental processes of mechanical designers based on empirical data. *Journal of Engineering, 2* (2), 113-125.

Suh, N. (1990). *The Principles of Design.* Oxford University Press.

Taratoukhine, V. (2002). Conflict Management in Computer Supported Collaborative Design: The Methodology and Applications. *Proceedings of the 2002 IEEE International Conference on Artificial Intelligence Systems.*

Tomiyama, T. (1994). From general design theory to knowledge-intensive engineering. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 8*(4), 319-333.

Tomiyama, T., & Yoshikawa, H. (1987). Extended general design theory. *in: H. Yoshikawa,E.A. Warman (Eds.), Design Theory for CAD, Proceedings IFIP WG5.2 Working Conference on Design Theory for CAD* (pp. 95-130). Amesterdam: Elsevier Science Publishers B.V.

UK Marine Data Archive Centre . (2013). Retrieved from The Data Archive for Seabed Species and Habitats (DASSH): http://www.dassh.ac.uk/data/data_management.html

Villacampa, Y., & Uso-Domenech, J. (1999). Mathematical model of complex structure system. A linguistic version. *Int. J. Gen. Syst, 28*(1), 37-51.

Wang, L., Shen, W., Xie, H., Neelamkavil, J., & Pardasani, A. (2002). Collaborative conceptual design--state of the art and future trends. *Computer-Aided Design, 34*(13), 981-996.

Wang, M., & Zeng, Y. (2009). Asking the right questions to elicit product requirements. *International Journal of Computer Integrated Manufacturing, 22*(4), 283–298.

Wikipedia. (2006). *Brake System History*. Retrieved from http://www.bookrags.com/Brake.

Wikipedia. (2011). *Apollo program*. Retrieved from http://en.wikipedia.org/wiki/Apollo_program.

Wikipedia. (2013). *System Context Diagrams*. Retrieved from http://en.wikipedia.org/wiki/System_context_diagram.

Yan, B., & Chen, B. (2005). *Automatic mesh generation.* Montreal, Canada.

Yao, S., Yan, B., Chen, B., & Zeng, Y. (2005). An ANN based element extraction method for automatic finite element mesh generation. *Expert System with Applications, 29*(1), 193-206.

Yesilbas, L. G. (2004). Towards a knowledge repository for collaborative design process: focus on conflict management. *Computers in Industry, 55*(3), 335-350.

Yoshikawa, H. (1981). *General Design Theory and a CAD System.* North-Holland Publishing Company.

Yoshikawa, H. (Ed.). (1985). *International Symposium on Design and Synthesis.* Amsterdam: Elsevier Science Publishers B.V.

Yoshikawa, H. (1989). Design philosophy: the state of the art. *Annals of the CIRP, 38*(2), pp. 579-586.

Yoshikawa, H. (1981). General design theory and CAD system, in: T. Sata, E.A. Warman(Eds.). *Man-Machine Communication in CAD/CAM, Proceedinds IFIP WG 5.2 Working Conference*, (pp. 35-38). North Holland, Amsterdam.

Yu, Z. (1990). *Theory of Motor Vehicle (In Chinese).* Beijing: China Machine Press.

Zeng, Y. (2001). *Axiomatic Approach to the Modeling of Product Conceptual Design Process Using Set Theory.* Ph.D Thesis, University of Calgary.

Zeng, Y. (2002). Axiomatic theory of design modeling. *Transaction of SDPS: Journal of SDPS*, 1-18.

Zeng, Y. (2003). Formalization of design requirements. *Integrated Design and Process Technologies, IDPT-2003*.

Zeng, Y. (2004). Environment-based formulation of design problem. *Transaction of SDPS: Journal of Integrated Design and Process Science, 8*(4), 45-63.

Zeng, Y. (2008). Recursive object model (ROM)—Modelling of linguistic information in engineering design. *Computers in Industry, 59*, 612-625.

Zeng, Y., & Cheng, G. (1991). On the logic of design. *Design Studies, 12*(3), 137-141.

Zeng, Y., & Cheng, G. (1993). Knowledge based free mesh generation of quadrilateral elements in two dimensional domains. *Microcomputers in Civil Engineering, 8*(4), 259-270.

Zeng, Y., & Gu, P. (1999a). A science-based approach to product design theory Part I: Formulation and formalization of design process. *Robotics and Computer-Integrated Manufacturing, 15* (4), 331-339.

Zeng, Y., & Gu, P. (199b). A science-based approach to product design theory Part II: formulation of design requirements and products. *Robotics and Computer-Integrated Manufacturing, 15*, 341-352.

Zeng, Y., & Gu, P. (1999c). General Design Governing Equation. *NSFC Grantee's Conference on Design and Manufacturing Engineering.* Vancouver, BC, Canada.

Zeng, Y., & Yan, B. (2006). Environment-based Design: Concept Generation. *The Ninth World Conference on Integrated Design & Process Technology.* San Diego, California.

Zeng, Y., & Yao, S. (2009). Understanding design activities through computer simulation. *Advanced Engineering Informatics, 23*, 294-308.

Zeng, Y., Pardasani, A., Antunes, H., Li, Z., Dickinson, J., Gupta, V., & Baulier, D. (2004). Mathematical foundation for modeling conceptual design sketches. *Transactions of the ASME: Journal of Computing and Information Science in Engineering, 4*(2), 150-159.

Zhang, J. (1997a). Distributed representation as a principle for the analysis of cockpit information displays. *International Journal of Aviation Psychology, 7*, 105-121.

Zhang, J. (1997b). The nature of external representations in problem solving. *Cognitive Science, 21*, 179-217.

Appendix

A1. A New Method to Solve Design Governing Equation

This appendix is an abbreviation from paper (Zeng & Yan, 2006). In this paper, we derived a formal method to solve Design Governing Equation and proposed a design process based on the Governing Equation.

Design is a recursive process in which design requirements and product descriptions evolve in the same time (Zeng & Cheng, 1991; Zeng & Gu 1999a). This process can be represented in Figure A-1 from which it can be seen that state of design includes two major parts: design requirements and product descriptions.

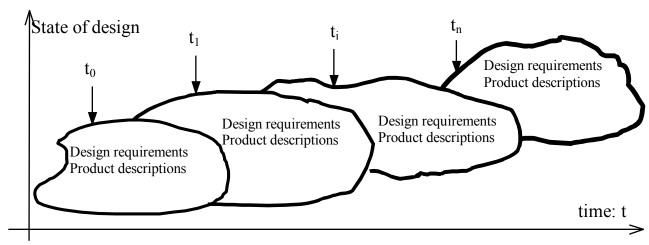


Figure A-1 Evolution of the design process

Since there are various types of design requirements and product descriptions at different stages for a real-life design problem, it is critical to represent the variety of design with a base. This can be achieved by the notion "engineering system", established in the axiomatic theory of design modeling.

An *engineering system* is the structure of an object (Ω) including both product (S) and its environment (E). Mathematically,

$$\Omega = E \cup S, \forall E, S \qquad \textbf{(A-1)}$$

Generally speaking, everything except the product itself can be seen as its environment. Then the engineering system ($\oplus\Omega$) can be represented as in Equation A-2 and illustrated in Figure A-2.

$$\oplus \Omega = \oplus(E \cup S) = (\oplus E) \cup (\oplus S) \cup (E \otimes S) \cup (S \otimes E) \qquad \textbf{(A-2)}$$

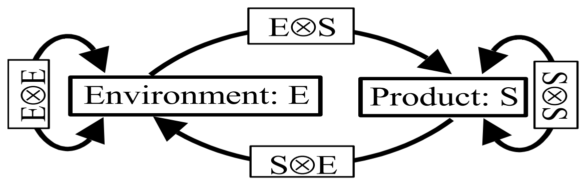


Figure A-2 Engineering system

Informally, a *design problem* can be defined as a proposition in which something has to be designed to meet the descriptions of a request for the design. In (Zeng & Gu, 1999c), the authors pointed out that a design problem can be symbolically represented as

$$P^d = \lambda(\oplus\Omega_s, \oplus\Omega_0) \qquad \textbf{(A-3)}$$

Where λ can be seen as the inclusion relation ($\supseteq$). In the stage of formulating a design problem, $\oplus\Omega_s$ is an unknown; $\oplus\Omega_0$ is the only thing defined. Obviously, $\oplus\Omega s$ should include all the information in $\oplus\Omega_0$. A design problem is solved when P_d assumes the value of "true". Since $\oplus\Omega s$ is an unknown in the problem formulation stage, Equation A-3 can be further transformed into the following algebraic form:

$$P^d = K_e(\oplus\Omega_0) \qquad \textbf{(A-4)}$$

Where, K_e is called evaluation operator, evaluating the results generated by using synthesis operator. The theorem of design problem structure can be derived from Equation A-3 by using the mathematical operations provided in the axiomatic theory of design modeling.

Theorem of Design Problem Structure:

> A *design problem* is composed of three parts: the environment in which the designed product is expected to work, the requirements on product structure, and the requirements on product performances.

Based on this theorem, the state of design can be defined by product environment, product description, performance requirements, and structural requirements. They provide a base for representing design problems (Zeng, 2002, 2003). These components keep on changing until final design solutions are found to solve the design problem. Hence, the evolution of design can be refined as in A-3.

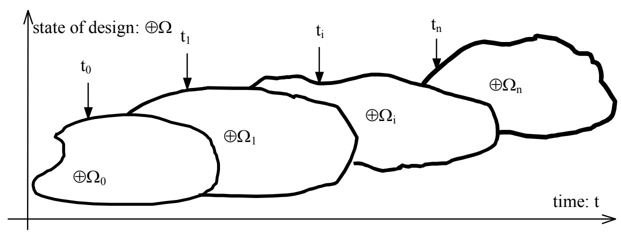


Figure A-3 Evolution of the design process: refined

Each new state of design comes from the design problem defined by its previous state. By applying Equation A-4, we have

$$P_i^d = K_i^e(\oplus\Omega_i), \qquad \textbf{(A-5)}$$

Where K_i^e is the evaluation operator determined by $\oplus\Omega_i$. p_i^d represents the current design problem corresponding the state $\oplus\Omega_i$. On the other hand, a new state of design always comes out of its previous problem definition. This is usually achieved by a synthesis operator K_i^s, so that

$$\oplus \Omega_{i+1} = K_i^s(P_i^d) \quad \textbf{(A-6)}$$

As long as P_i^d is not true, the design process has to continue. By substituting Equation A-5 into Equation A-6, we have

$$\oplus \Omega_{i+1} = K_i^s(K_i^e(\oplus\Omega_i)) \quad \textbf{(A-7)}$$

Equation A-7 is called *design governing equation* (Zeng & Gu, 1999c). It underlies the design process and governs design activities. The basic concept behind this equation is the recursive logic of design (Zeng & Cheng, 1991), which states that design is a recursive process, during which the design solution and design problem interdependently evolve (Zeng & Cheng, 1991, 1993; Maher & Tang, 2003; Dorst & Cross, 2001). Equation A-7 can be further formulated as

$$\oplus \Omega_{i+1} = D^i(\oplus\Omega_i) \text{ where } D^i = K_i^s \bullet K_i^e \quad \textbf{(A-8)}$$

<u>Resolution of Design Governing Equation</u>

With the governing equation, a design problem becomes how to solve this equation. We rewrite Equation A-2 here:

$$\oplus \Omega = \oplus(E \cup S) = (\oplus E) \cup (\oplus S) \cup (E \otimes S) \cup (S \otimes E) \quad \textbf{(A-9)}$$

As we know, the environment is where the product will be put to work. It can be human being for which the product serves, including all human users and operators in the life cycle of a product. It can be other products created or built by human beings. It can also be the nature, which determines all the laws that the product has to obey. These three parts are called human, built, and natural environment, respectively, represented as

$$E = E^h \cup E^b \cup E^n \quad \textbf{(A-10)}$$

where E^h is human environment, E^b is built environment, and E^n is natural environment. In the context of a specific design task, some parts of environment do not affect the product. Hence, the environment can be divided into direct and indirect environments according to their influence on a product. Only direct environment will be considered in a product system. This classification and Equation a-9 can be illustrated in Figure A-4.

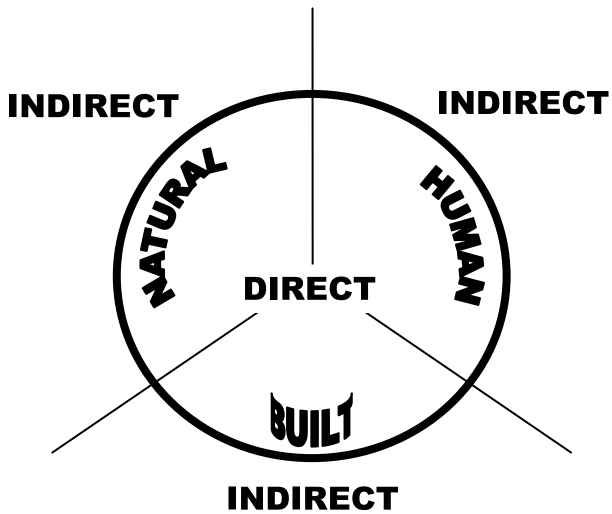


Figure A-4 Classification of environment

A product, either physical or non-physical, is usually an artifact that is designed to provide services to its human or built environment. These services are often called the purposes of the product. To function properly, the product must survive in the natural environment.

The initial and final states of a design can be conceptually written as in equations A-11 and A-12, respectively.

$$\oplus \Omega_0 = \oplus(E_0 \cup S_0) = \oplus(E_0 \cup \Phi) = \oplus E_0 . \quad \textbf{(A-11)}$$

$$\oplus\Omega_s = \oplus(E_s \cup S_s) = (\oplus E_s) \cup (\oplus S_s) \cup (E_s \otimes S_s) \cup (S_s \otimes E_s) \quad \textbf{(A-12)}$$

The design process is a process of transforming the initial product system ($\oplus\,\Omega_0$) into the final product system ($\oplus\ \Omega_s$) such that the final product meets all the product requirements. Since environment is also an object, it includes other objects according to corollary1. In fact, as stated before, $E = E^n \cup E^b \cup E^h$. And furthermore, we have

$$E^a = \bigcup_{i=1}^{a_e} e_i^a \quad \textbf{(A-13)}$$

E^a is one of E^n , E^b or E^h ; e_i^a is called primitive environment of E^a . Similarly, the product to be designed can also be represented as

$$S^a = \bigcup_{i=1}^{a_s} s_i^a \qquad \text{where } s_i^a = f(r_i^a) \quad \textbf{(A-14)}$$

s_i^a is called primitive product. Based on primitive environments and primitive products, we can define primitive relations.

$$R^a = \bigcup_{i=1}^{a_s} r_i^a \quad r_i^a = (v \otimes e_i^a) \cup (e_i^a \otimes v) \ \ v \subset S \quad \textbf{(A-15)}$$

It is noted from Equation A-15 that the definition of primitive relation is of recursive nature. On the one hand, in order to find a primitive relation, we need to have a primitive product; on the other hand, to obtain a primitive product, we need to know the primitive relation. In this situation, an iterative process is needed to solve such a problem. In addition, not every primitive relation r_i^a corresponds to a primitive product; it may exert a design requirement, such as material requirement, on the product to be designed.

In the design process, i.e. in the transformation from $\oplus\Omega_0$ to $\oplus\Omega_s$, there are some intermediate state $\oplus\Omega_i = \oplus(E \cup S_i)$, where S_i is an incomplete product that may satisfy some requirements but not all. Considering two consecutive states, state *i* and state *i+1*, we have

$$S_{i+1} = S_i \cup s_{i+1}^a \tag{A-16}$$

$$\begin{aligned}
\oplus\Omega_{i+1} &= \oplus(E \cup S_{i+1}) = \oplus E \cup \oplus S_{i+1} \cup (E \otimes S_{i+1}) \cup (S_{i+1} \otimes E) \\
&= \oplus E \cup \oplus(S_i \cup s_{i+1}^a) \cup (E \otimes (S_i \cup s_{i+1}^a)) \cup ((S_i \cup s_{i+1}^a) \otimes E) \\
&= \oplus E \cup \oplus S_i \cup \oplus s_{i+1}^a \cup (S_i \otimes s_{i+1}^a) \cup (s_{i+1}^a \otimes S_i) \cup (E \otimes S_i) \\
&\quad \cup (E \otimes s_{i+1}^a) \cup (S_i \otimes E) \cup (s_{i+1}^a \otimes E) \\
&= \oplus\Omega_i \cup \oplus s_{i+1}^a \cup (S_i \otimes s_{i+1}^a) \cup (s_{i+1}^a \otimes S_i) \cup (E \otimes s_{i+1}^a) \cup (s_{i+1}^a \otimes E) \\
&= \oplus\Omega_i \cup \oplus\delta\Omega_{i+1} \cup (S_i \otimes s_{i+1}^a) \cup (s_{i+1}^a \otimes S_i) \\
&= \oplus\Omega_i \cup \oplus\delta\Omega_{i+1} \cup \delta B_{i+1}
\end{aligned} \tag{A-17}$$

Here, we define $\delta\Omega_{i+1}$ as a partial or primitive product system that is built up in design state *i+1*, δB_{i+1} as partial inner boundary between newly-added primitive product and existing product. Equation A-17 shows a synthesis process for solving design governing equation. Depending on the sequences and methods of decomposing the environment, the result of $\oplus\Omega_{i+1}$ may be many. Therefore, an evaluation is needed to select the best solution.

Each time the product-environment boundary is decomposed, a primitive relation is identified and a new corresponding primitive product is recursively generated and integrated into the existing incomplete product system, and then new boundaries are formed. A primitive product is a physical representation of a primitive relation. Each primitive product s_i^a corresponds to one or several primitive relations between the product to be designed and the primitive environment e_i^a . The design process continues until all the relations have a corresponding primitive product and all the primitive products contained in a product are structurally related, that is $\delta\Omega_{i+1} = \Phi$. At this state the design concept is generated.

In summary, in each step of the design process, after a primitive environment is identified, the primitive relation is analyzed, and then the corresponding primitive product is generated. This design process is environment-based, which can be shown in Figure -5.

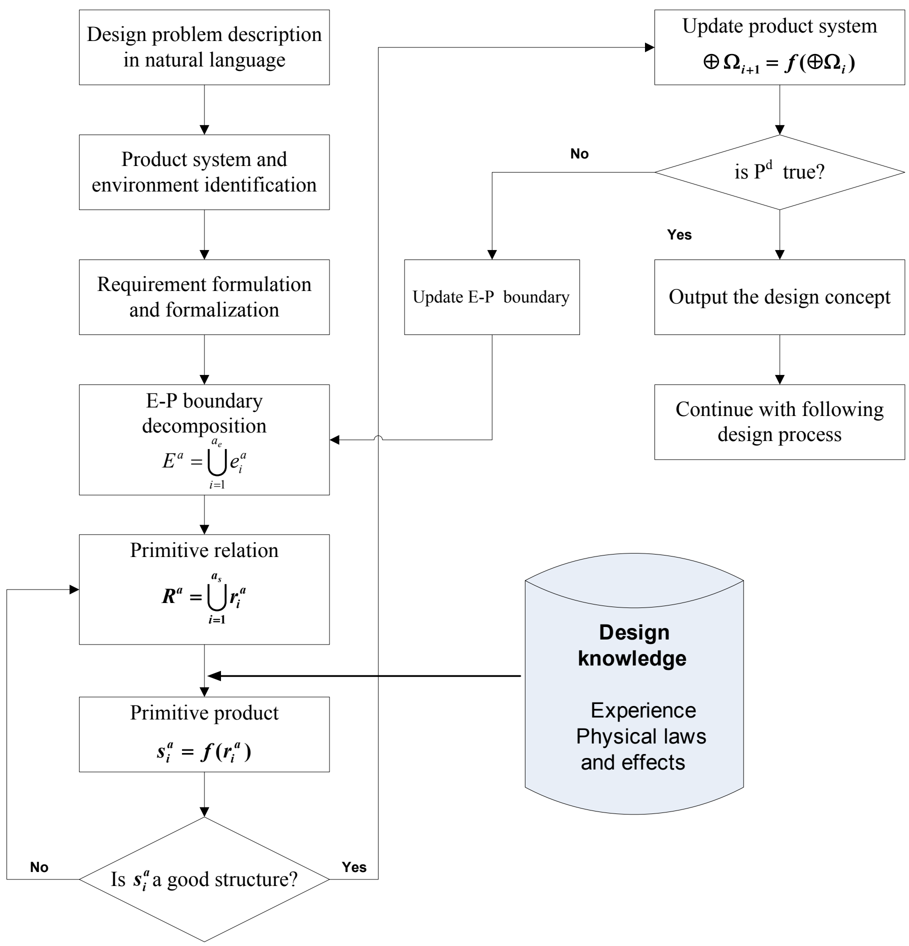


Figure A-5 A process of solving design governing equation

A2. ROM Analysis of Definitions of Conflicts

In the following, we will analyse more conflict definitions from different sources and identify the common structure behind the different definitions of conflict. For each definition, we will draw its ROM diagram first, and then simplify or clarify this definition based on a question asking/answering technology (Wang & Zeng, 2009).

Merriam-Webster dictionary (Merriam-Webster, 2007):

> Conflict is competitive or opposing action of incompatibles, or mental struggle resulting from incompatible or opposing needs, drives, wishes, or external or internal demands.

The ROM diagram of this definition is shown in the following figure.

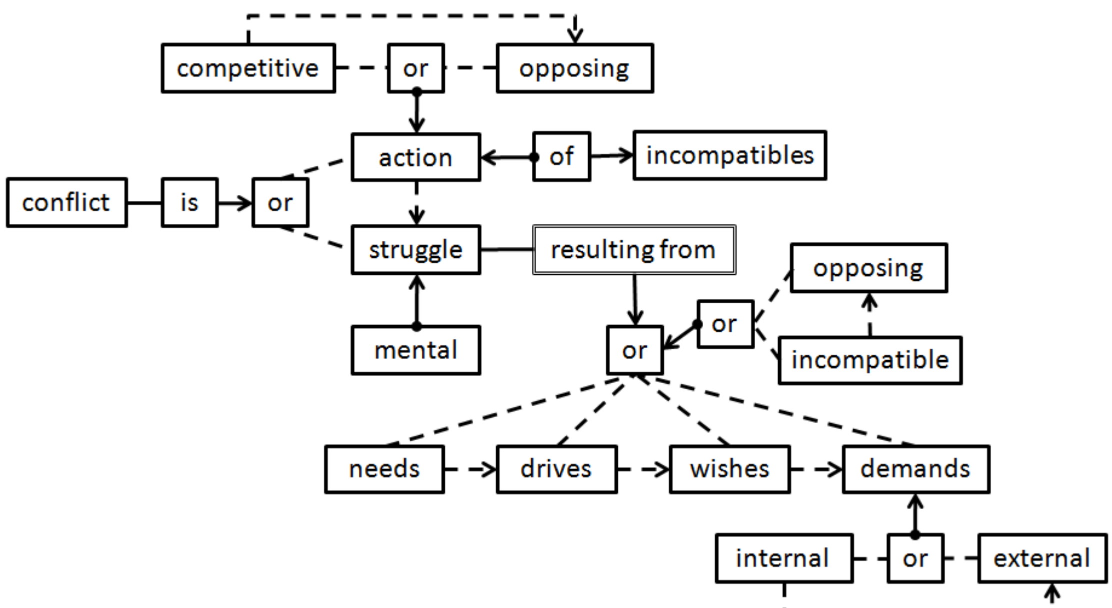


Figure A-6 ROM diagram of conflict definition (Merriam-Webster)

This sentence can be divided into two parallel sentences: 1) conflict is competitive or opposing action of incompatibles. 2) Conflict is mental struggle resulting from incompatible or opposing needs, drives, wishes, or external or internal demands. First, we analyse the first sentence by asking the following questions.

1. What is action?

   ANS: the word *action* in this sentence means doing things.

2. Who is doing the things (or the subject of the ***action***)?

   ANS: the subject of the action may be a person.

3. What does *incompatible* mean?

   ANS: Two actions or believes that etc. that are incompatible cannot exist or be accepted together.

4. What does *action of incompatible* mean?

   ANS: It means that the subject (the person) want to take more than two actions that are not compatible. For instance, the person cannot sleep and work at the same time, i.e. the two actions, *sleep and work,* are not compatible at the same time.

5. How many actions does the subject want to take?

   ANS: At least two.

Therefore, this definition can be rephrased as: conflict is a situation that a person is required to do two oppossing or competitive things that are not compitable to each other.

Britannica (Britannica, 2007):

Conflict is the arousal of two or more strong motives that cannot be solved together in psychology sense.

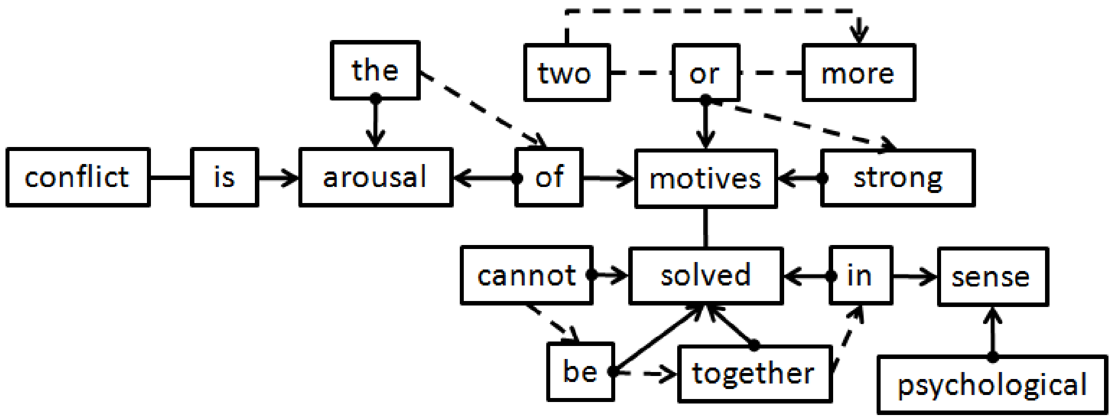


Figure A-7 ROM diagram of definition of conflict from Britannica

Questions for this definition:

1. What are motives?

   ANS: the reason that makes someone do something.

2. What does *arousal* mean?

   ANS: stimulation to action

We noticed that the sentence *that cannot be resolved together* means incompatible. Therefore, a conflict is a situation that two or more incompatible motives exist in one and the same subject.

<u>English and English's definition</u> (English & English, 1958):

Conflict is the opposition of response tendencies, represented as in figure A-8.

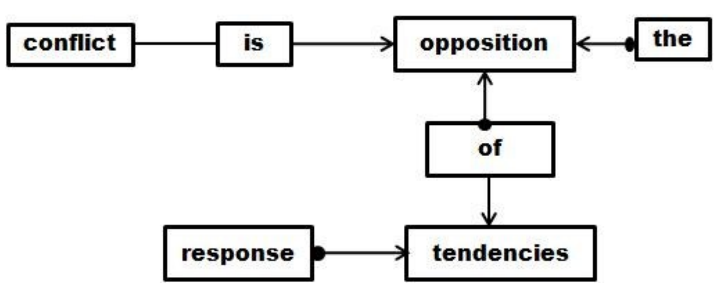


Figure A-8 ROM diagram of definition of conflict by English

Questions about this definition:

1. What does tendency mean?
2. What does opposition mean?

Webster dictionary explains the word tendency as a proneness to a particular kind of thought or action, or the purposeful trend of something written or said. Opposition means disagreement, or he relation between two propositions having the same subject and predicate but differing in quantity or quality or both. Therefore, this definition indicates that conflict is the disagreement between two opposing tendencies. Naturally, we would ask what the subject of these two tendencies is. It must be a person that has two opposite tendencies. At this point three objects are identified, namely, a person and two tendencies that has a competing relation over this particular person.

<u>Harrinton *et al*</u> (Harrinton, Soltan, & Forskitt, 1995):

Conflict is disagreement between two or more viewpoints on some decision or values proposed in design. The ROM diagram is shown in figure A-9.

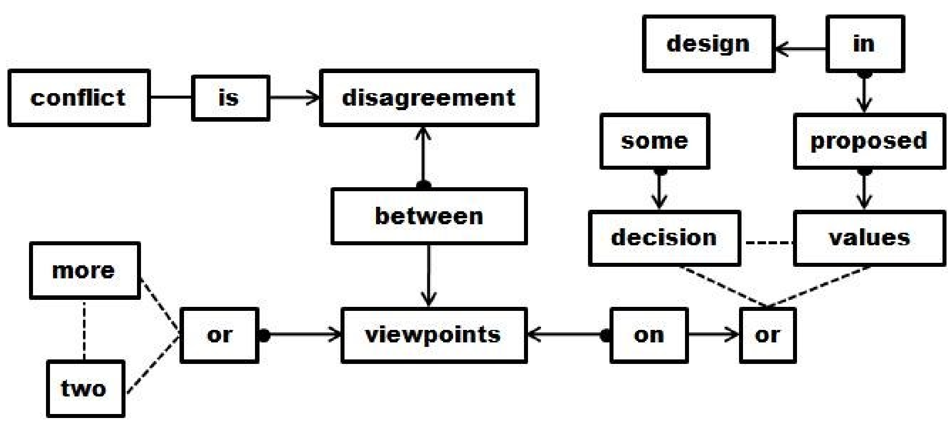


Figure A-9 Harrinton's definition of conflict

By Webster's dictionary, a viewpoint means a position or perspective from which something is considered or evaluated. Therefore, this definition indicates that conflict in design is disagreement between the ways to make decisions or propose values.

A3. A list of publications by the author

1. **Yan, B.** & Zeng, Y. (2011). Design conflict: conceptual structure and mathematical representation. *Transactions of the SDPS: Journal of Integrated Design and Process Science.* 15(1), pp. 75-89.

2. **Yan, B.** and Y. Zeng (2009). A Formal Representation of Technical Systems, in Book Series *Advanced Concurrent Engineering* , ISSN 1865-5440.

3. Y. Zeng, A. Nguyen, **B. Yan** and S. Li (2008). A distance-based parameter free curve reconstruction algorithm. *Computer-Aided Design*, Vol.40, No. 2, 210-222.

4. Zeng, Y., & **Yan, B**. (2006). Environment-based Design: Concept Generation. *The Ninth World Conference on Integrated Design & Process Technology.* San Diego, California.

5. S. Yao, **B. Yan,** B. Chen, and Y. Zeng (2005), An ANN-based element extraction method for automatic mesh generation. *Expert Systems with Applications,* Vol. 29, No.1, pp. 193-206.

6. Y. Zeng, **B. Yan**, B. Chen, and S. Yao (2004). A theoretical and experimental study on design creativity. *Inaugural CDEN Design Conference*, McGill University, Montreal, Quebec, July 29-30, 2004.