Design evolution and knowledge handling for product conceptualization in new product development

Rao, Zhiming

2008

Rao, Z. M. (2008). Design evolution and knowledge handling for product conceptualization in new product development. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/13403>

<https://doi.org/10.32657/10356/13403>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**DESIGN EVOLUTION AND KNOWLEDGE
HANDLING FOR PRODUCT
CONCEPTUALIZATION IN NEW PRODUCT
DEVELOPMENT**

**RAO ZHIMING
SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING
2008**

**Design Evolution and Knowledge Handling
for Product Conceptualization in New Product
Development**

Rao Zhiming

School of Mechanical and Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2008

ACKNOWLEDGMENTS

I hereby wish to express my wholehearted gratitude to those in guiding and supporting the accomplishment of my work.

First and foremost, I would like to express my deep gratitude to my supervisor, A/P Chen Chun-Hsien. He always gives me wonderful and inspiring ideas about my research. His brilliant supervision as well as trust in my work has allowed me to explore many ideas contributed in this work.

I am indebted to NTU for awarding me with a research scholarship.

I am grateful to the technical staff in the Advanced Design and Modeling Laboratory for their kind assistance rendered in the course of my research. I also want to express my appreciation to all of my friends who give me the courage and share with me the joys in NTU, especially to Mr. Chong Yih Tng, Miss Lee Hui Mien, and Mr. Cai Minbo.

Last but not least, I wish to thank my family for their understanding, patience, encouragement, continuous concern and moral support.

TABLE OF CONTENTS

ACKNOWLEDGEMENT	I
TABLE OF CONTENTS	II
ABSTRACT	VI
LIST OF PUBLICATIONS	VII
LIST OF FIGURES	VIII
LIST OF TABLES	XI
LIST OF ABBREVIATIONS	XII
CHAPTER 1 INTRODUCTION	1
1.1 Background	1
1.2 Objective	5
1.3 Significance	7
1.4 Scope	8
1.5 Organization of the Report	9
CHAPTER 2 LITERATURE REVIEW	11
2.1 Methodologies for Design Evolution and Evaluation	11
2.1.1 Concept Development and Selection Methods	11
2.1.2 Concept Development and Selection Methods	12
2.2 Information Reuse in Evolutionary Product Design	19
2.2.1 Design Information Reuse and Recovery	19
2.2.2 Case-Based Reasoning for Evolutionary Design	21
2.3 Product Innovation in New Product Development	25
2.4 Artificial Intelligence (Knowledge-based Approach) in Product Development	27
2.4.1 Knowledge-Based System in Design Evolution	27

2.4.2 Knowledge-Based System in Design Evaluation	29
2.4.3 Blackboard System	31
2.4.4 Knowledge Acquisition for Product Development	33
2.5 Discussions	39
2.6 Summary	41
CHAPTER 3 RESEARCH ORIENTATION	44
3.1 Overview of the Prototype System	44
3.2 Case-Based Design Evolution	45
3.2.1 A Feature-based Design Evolution Approach	47
3.2.2 An Innovative Adaptation Procedure (IAP)	47
3.3 A Novel Knowledge Acquisition Approach for Design Evaluation	48
3.4 Summary	50
CHAPTER 4 A FEATURE-BASED DESIGN EVOLUTION APPROACH VIA CASE-BASED REASONING	51
4.1 Introduction	51
4.2 CBR in New Product Development	52
4.3 A Feature-based CBR Approach	54
4.3.1 Feature Relationship Decomposition (FRD)	55
4.3.2 Feature Priority Determination	59
4.3.3 Feature-based Case Retrieval	63
4.3.3.1 Clustering Analysis and Similarity Reasoning	64
4.3.3.2. Overall Case Retrieval Process	66
4.4 Case Study	71
4.4.1 An Example	71
4.4.2 Results and Discussion	82

4.5 Summary	86
CHAPTER 5 A TRIZ-ENHANCED INNOVATIVE ADAPTATION	
PROCEDURE (IAP)	88
5.1 Introduction	88
5.2 Overview of TRIZ Methodology	89
5.3 Overall Framework of the Proposed Approach	91
5.4 A Case Study	97
5.4.1 How HDD Works	98
5.4.2 Problem Description	100
5.4.3 Problem Solving with TRIZ	102
5.5 Summary	105
CHAPTER 6 A MATRIX REPRESENTATION AND MAPPING (MRM)	
APPROACH FOR KNOWLEDGE ACQUISITION	107
6.1 Introduction	107
6.2 Overview of the MRM Approach	109
6.3 Fundamental Concepts of the MRM Approach	110
6.4 Description of the MRM Approach	111
6.4.1 Identification of knowledge sources	112
6.4.2 Generation of taxonomic tree (general sorting)	112
6.4.3 Identification of linkages between conditions and conclusions sets	114
6.4.4 Generation of sub-clauses of IF and THEN	120
6.4.5 Identification of logical relationships between sub-clauses	122
6.4.6 Generation of rules	125
6.5 A Case Study	127
6.6 Summary	138

CHAPTER 7 CONCLUSIONS AND FUTURE WORK	140
7.1 Conclusions	140
7.2 Limitations	145
7.3 Future Work	146
REFERENCE	148
APPENDIX A GRAPH DECOMPOSITION ALGORITHM	171
APPENDIX B RESULTS OF CLUSTERING ANALYSIS AND FEATURE-BASED CASE RETRIEVAL	179

ABSTRACT

The highly competitive and rapidly changing global business environment requires companies to deliver new products with better performance, lower cost, higher quality and shorter time-to-market. To develop a successful product, conceptualization of the product plays an extremely important role and, hence, deserves much more attention. In essence, this work views product concept development (PCD) as a strategic task that has a major impact on subsequent production related activities.

In PCD, design evolution and evaluation are of great importance to generate satisfactory design concepts. To facilitate the efficiency of PCD and, hence, shorten the time-to-market, this work aims at establishing new approaches and methodologies to address the decisive issues of design evolution and evaluation in product conceptualization. In so doing, it is found that current practices and literature have not adequately addressed the issue of design information reuse in conceptual design evolution. Such difficulties as indexing, retrieval and modification of prior design knowledge remain to be dealt with. On the other hand, a knowledge-based approach has been recognized as a promising and logical alternative to support design evaluation. However, it is well-known that knowledge acquisition is the most time-consuming phase, and, therefore, the bottleneck in constructing a knowledge-based system (KBS). There are two major omissions in existing work. First, the quantitative characteristics of information and interrelationships between various knowledge elements, which are of critical importance in constructing a KBS, have not been investigated. Second, an integrated and easy-to-use knowledge acquisition process with a well-organized knowledge representation scheme to facilitate rule generation needs to be explored.

Accordingly, a feature-based design evolution approach via case-based reasoning (CBR) is proposed to facilitate the evolution process of concept generation. In this approach, a feature relationship decomposition (FRD) procedure is postulated. The FRD provides an indexing scheme of CBR for design reuse with more flexibility of case retrieval. The performance of analytical hierarchy process (AHP) is enhanced with FRD via reducing the lengthy calculations of pair-wise comparisons. To arm the proposed design evolution approach with the capability of innovation, a TRIZ-enhanced innovative adaptation procedure (IAP) is investigated. Additionally, a matrix representation and mapping approach (MRM) is proposed to tackle the difficulty of knowledge acquisition in constructing a KBS to support design evaluation. The MRM approach establishes a systematic procedure to model the rule generation process mathematically. It uses matrix representation and mapping techniques to represent the knowledge attributes and reveal their interrelationships. It addresses the quantitative characteristics of relationship strength between various knowledge elements using a relationship estimation procedure. Three case studies on wood golf club design, hard disk design, and diagnosing automotive systems are employed to verify the proposed approaches, i.e. FRD, IAP, MRM, respectively.

The economic impact of this work is to enhance design efficiency in new product development so as to reduce design time and hence cost by effectively making use of historical design data. In this work, the proposed CBR-based design system has been tested on tangible product design problems. Given the proposed benefits, the possibility of extending the current work to a broader scope will merit further exploration.

LIST OF PUBLICATIONS

1. Rao, Z.M. and Chen, C.-H. (2005). A matrix representation and mapping approach to knowledge acquisition for product design. *Lecture Notes in Computer Science*, Vol. 3681, 311-317.
2. Rao, Z.M. and Chen, C.-H. (2005). A Feature-based CBR Approach for Product Concept Development. *Proc. The 2005 Pan-Pacific Vocational Design Conf. – Design Nexus*, pp. 31-40, Yunlin, Taiwan, October.
3. Rao, Z.M. and Chen, C.-H. (2005). A CBR-based Design Evolution Approach for New Product Development. *Proc. The Int. Design Congress – IASDR 2005 (2005idc)*, 10p, Yunlin, Taiwan, November.
4. Rao, Z.M. and Chen, C.-H. (2006). A feature-based case retrieval approach for conceptual design evolution. *The 36th International Conference on Computers and Industrial Engineering: Challenges for Advanced Technology Industries*, 2589-2997, June 20-23, Taipei, Taiwan.
5. Rao, Z.M. and Chen, C.-H. (2007). A Knowledge Sorting and Matrix Representation Approach for Developing Knowledge-based Product Design Systems. *Lecture Notes in Computer Science*, Vol. 4692, 262-269.
6. Chen, C.-H. and Rao, Z.M. (2007). MRM: A matrix representation and mapping approach for knowledge acquisition. *Knowledge-Based Systems*, Vol. 21, 284-293.
7. Chen, C.-H. and Rao, Z.M. (2006). Developing a Knowledge-Based Product Design System Using a 3D Knowledge Organization Scheme. *Engineering Applications of Artificial Intelligence*. (In review)
8. Chen, C.-H. and Rao, Z.M. (2007). A Feature-based Design Evolution Approach Via Case-based Reasoning. *Knowledge-Based Systems*. (In Review)

LIST OF FIGURES

Figure 1.1 Product Cost Commitment during Different Phases	1
Figure 1.2 Organization of the thesis	10
Figure 2.1 Phase of conceptual design	12
Figure 3.1 The overall framework of CBDES in a design process	46
Figure 4.1 A typical CBR process	53
Figure 4.2 A framework of the feature-based CBR approach	55
Figure 4.3 Graph decomposition for feature relationship	59
Figure 4.4 AHP hierarchy derived from FRD	63
Figure 4.5 Overall feature-based case retrieval process	67
Figure 4.6 Pseudo-code for the core algorithm of feature-based case retrieval process	70
Figure 4.7 Specifications of a wood head golf club (Maltby 1982)	72
Figure 4.8 FRD procedure for golf club features	74
Figure 4.9 Graph decomposition with application program	74
Figure 4.10 AHP hierarchy for feature priority determination	77
Figure 4.11 Screenshot of the GUI for feature-based case retrieval	79
Figure 4.12 A sample of SOM training result (for 3×2 output grid), $t_{\max} = 200$	79
Figure 4.13 Results of clustering analysis and case retrieval for material	81
Figure 4.14 Results of clustering analysis and case retrieval for loft	81
Figure 5.1 Overview of the TRIZ enhanced innovative adaptation procedure	92
Figure 5.2 A typical hard disk drive	99
Figure 5.3 Head slap on a disk drive	101
Figure 5.4 A screenshot of searching a science effect for self-service	104

Figure 5.5 Sketch of open HDD with accelerometer sensor	105
Figure 6.1 Overall procedure of the MRM approach	110
Figure 6.2 A conceptual process for rule formation	111
Figure 6.3 Interactions between knowledge attributes/values	117
Figure 6.4 Sample effects of knowledge attributes/values on system tasks	118
Figure 6.5 Sample relationship matrices for attributes and values	120
Figure 6.6 Sample logical relationships of sub-clauses in A_{Xi} and B_{Yi}	125
Figure 6.7 Logical relationships of sub-clauses in A_{X1} and B_{Y1}	126
Figure 6.8 The taxonomic tree generated from general sorting	128
Figure 6.9 Logical relationship for IF portion of Rule 1	135
Figure 6.10 A screenshot of the testing results	137
Figure A.1 Main interface of graph decomposition software	175
Figure A.2 Inputted feature names	175
Figure A.3 Relationship table for design features	176
Figure A.4 Parameters configuration dialog box	177
Figure A.5 Graph decomposition with application program	178
Figure A.6 Online help	178
Figure B.1 Results of clustering analysis and case retrieval for lie	179
Figure B.2 Results of clustering analysis and case retrieval for price	180
Figure B.3 Results of clustering analysis and case retrieval for sole radius	180
Figure B.4 Results of clustering analysis and case retrieval for face bulge	181
Figure B.5 Results of clustering analysis and case retrieval for roll	181
Figure B.6 Results of clustering analysis and case retrieval for face progression	182
Figure B.7 Results of clustering analysis and case retrieval for neck length	182
Figure B.8 Results of clustering analysis and case retrieval for volume	183

Figure B.9 Results of clustering analysis and case retrieval for weight	183
Figure B.10 Results of case retrieval for flex	184

LIST OF TABLES

Table 4.1 Features grouped in blocks	59
Table 4.2 Numerical scale of the relative importance	61
Table 4.3 Random index for the comparison matrix	63
Table 4.4 Pair-wise comparison matrices between feature alternatives	76
Table 4.5 Results of consistency check for comparison matrices	77
Table 4.6 Feature priorities based on weights	78
Table 4.7 Sample trained weight vectors from the SOM	80
Table 4.8 Final attribute values for each feature	82
Table 4.9 Number of pair-wise comparisons by AHP with or without FRD	84
Table 4.10 Number of case instances for each feature in case study	85
Table 5.1 Partial cells of Altshuller's contradiction matrix	96
Table 5.2 Resolving the problem with corresponding inventive principles	103
Table 6.1 Scaled judgments of various effects	118
Table 6.2 The attributes and values in conditions set	129
Table 6.3 The attributes and values in conclusions set	130
Table 6.4 Linguistic values of effects on knowledge attributes on system tasks	131
Table 6.5 Quantified relationships and significant relationships for attributes	132
Table 6.6 The significant relationships between values	133
Table 6.7 Combinations of attributes and values for each rule	134
Table 6.8 All the rules generated in the case study	136

LIST OF ABBREVIATIONS

AHP	Analytical Hierarchy Process
AI	Artificial Intelligence
ART	Adaptive Resonance Theory
CAD	Computer-Aided Design
CBDES	Case-Based Design Evolution System
CBED	Case-Based Evolutionary Design
CBR	Case-Based Reasoning
CF	Certainty Factor
CI	Consistency Index
CONDENSE	Concurrent Design Evaluation System
CPI	Collaborative Product Innovation
C.R.	Connection Ratio
CR	Consistency Ratio
DCA	Design Compatibility Analysis
DFMA	Design For Manufacture and Assembly
FRD	Feature Relationship Decomposition
GUI	Graphical User Interface
HDD	Hard Disk Drive
HOQ	House of Quality
IAP	Innovative adaptation procedure
KBS	Knowledge-Based System
LCP	Life Cycle Planning
LEERS	Learning from Examples based on Rough Sets

MEMS	Micro-Electro-Mechanical System
MRM	Matrix Representation and Mapping
NNM	Nearest-Neighbor Matching
NPD	New Product Development
PCD	Product Concept Development
QFD	Quality Function Deployment
RBF	Radial Basis Function
RCE	Restricted Coulomb Energy
SOM	Self-Organizing Map
TOC	Theory of Constraints
TRIZ	Theory of Inventive Problem Solving
USSR	Union of Soviet Socialist Republics

Chapter 1 Introduction

1.1 Background

In order to secure a foothold in the highly competitive global market, companies nowadays must develop new products with high quality, low cost and short time-to-market. For this purpose, it is imperative to pay more attention to the design process. In essence, product design is viewed as a strategic task that has a major impact on the subsequent production related activities. High quality and low cost cannot be derived from manufacturing unless they are taken into account during designing (Dixon et al. 1990).

Figure 1.1 Product Cost Commitment during Different Phases (Dieter 2000)

Figure 1.1 illustrates the product cost commitment during different product life cycle phases. As shown in the figure, only a small fraction of the cost, i.e. around 5%, in developing a product is consumed by the product design process, while the rest 95% is

associated with the manufacturing process. In addition, about 70% to 80% of the product final cost is affected by design commitments resulted from decisions made during the product design process, while only about 25% of the product final cost is influenced by decisions made beyond the design phase. Thus, decisions made in the design process cost very little in terms of the overall product cost yet have a major effect on the cost of the product (Dieter 2000).

Product design is a complex task that requires a systematic approach to facilitate the efficiency and effectiveness of its process. Over the past few decades, design has evolved from “very simple” to “very complex” (Prasad 1994). Many researchers have suggested a number of different product design processes (e.g. Pahl and Beitz 1996, Otto and Wood 2001, Ulrich and Eppinger 2004). Among these processes, the most accepted is the one proposed by Pahl and Beitz. It comprises four main phases, namely product planning and task clarifying, conceptual design, embodiment design, and detail design. In the planning and task clarifying phase, such information as customer requirements are collected and initial design ideas are generated. The result of this phase is the formulation of the specifications of the design information in the form of a requirements list. The process of conceptual design is to determine the solution principles for the new design problem, which is achieved through abstracting the essential problems, establishing function structures, searching for suitable working principles and then combining those principles into a working structure. The outputs of this phase are the solution principles or concepts. Embodiment design, starting from a concept, determines the construction structure of a technical system in line with technical and economic criteria. In the fourth phase, detail design, the arrangement, forms, dimensions and surface properties of all the individual parts are finally laid down, the materials specified,

production possibilities assessed, cost estimated and all the drawings and other production documents produced.

During the conceptual design process, the concept solutions are generated to meet the product design specifications. The process begins with a set of customer needs and target specifications. These result in a set of product concepts from which the team will make a final selection. In the concept selection process, concepts are evaluated with respect to customer needs and other criteria, comparing the relative strength and weakness of the concepts, and selecting one or more concepts for further investigation, testing, or development (Ulrich and Eppinger 2004). In order to select valuable solutions and not to rule out any potential good ones, the solution space formed should be as large as possible. Moreover, the concept selection process often requires several iterations between convergence and divergence in the solution space.

Product development is a key mechanism for organizational growth. However, an increasingly globalized and competitive marketplace is forcing companies to develop better performance, lower cost and higher quality products in shorter timescales. To ride on this trend, a more cost- and time-effective strategy is to develop new products through modifying and/or combining existing design solutions. It is necessary to provide the designer with past design solutions so as to reuse design information. Evolutionary product design, instead of designing a product from scratch, is frequently adopted in practice to enhance product features and better serve the customers, while reducing the development cost and time (Tay and Gu 2003, Rao and Chen 2005a; 2005b). In the engineering product design process, it has been realized that reusing existing design information is much more cost-effective and time-saving than totally regenerating new information. The reuse of existing design concepts can be an important supplement to

the creative design efforts of designers who are faced with tight timescales and competitive market pressures. Much work on the reuse of such information in the evolutionary design process has been done by a number of researchers. However, as discussed in Chapter 2, most of these efforts are only effective under certain conditions and some limitations exist.

As aforementioned, most of the product life-cycle cost is committed during the design stage. The product design process, especially in the early design stage, deserves much more attention. The decisions made in the early stages of design have considerable influence on the costs and quality of products (Jared et al. 1994). Generally speaking, the later the changes made in the product design and development cycle, the more the costs and delays in the release of the final product will be. As such, the importance of design evaluation is apparent because a poor selection of a design concept can rarely be compensated at later design stages and incurs great expenses of redesign cost (Wang 1997). During the design evaluation process, designers should not only consider the product functionalities needed, but also take into account various evaluation criteria such as manufacturability, the ease of assembly and other life-cycle concerns as early as possible in the conceptual design stage. In order to conduct design evaluation with respect to different criteria effectively, a knowledge-based approach presents a logical alternative. It is well-known that the knowledge acquisition phase is a bottleneck in constructing a knowledge-based system. As such, in establishing a knowledge-based system to support the process of design evaluation, an approach to facilitate the efficiency of knowledge acquisition and rule extraction is highly desirable.

To sum up, in the early design stage, product concept development (PCD) plays an extremely important role in developing quality competitive products with the lowest cost

and the shortest time-to-market. In PCD, design evolution and knowledge handling are of great significance for generating satisfactory design concepts. Hence, it is highly necessary to investigate into a systematic approach to support the PCD process.

1.2 Objective

This research constitutes contribution towards an investigation into the strategies of conceptual design evolution and knowledge handling for product conceptualization in new product development (NPD). It aims at establishing novel methodologies to systematically improve the design efficiency for PCD. In response to the challenges of design evolution and knowledge handling in PCD, this work involves the following main objectives.

Objective 1: To propose an effective conceptual design evolution approach to facilitate the process of product concept development in NPD.

The approach aims to explore an intelligent way to facilitate the product concept generation via making use of existing design information. In order to effectively utilize the historical design data in designing a new product, the case-based reasoning (CBR) methodology will be adapted. It is established that case indexing and retrieval are crucial for any CBR system, especially when the design problem involves a huge case database. For this purpose, the design evolution approach starts from an in-depth study on the inter-relationships between the various design feature alternatives, based on which a new case retrieval method will be proposed to retrieve the potential design solutions for the new problem efficiently.

Objective 2: To explore a complementary approach to equip the proposed CBR design system with the capability of generating an innovative product concept based on partially available design information in the early design stage.

The approach attempts to incorporate the TRIZ (a Russian acronym for Theory of Inventive Problem Solving) methodology into the proposed CBR design evolution approach. It aims at enabling the design evolution approach to generate innovative design solutions based on partially available design information. It focuses on the adaptation stage of the CBR methodology. For this purpose, an innovative adaptation procedure (IAP) will be incorporated as a complementary module to enhance the capability of the proposed design evolution approach.

Objective 3: To investigate into a knowledge acquisition strategy to facilitate the effectiveness of rule generation in building a knowledge-based system for design evaluation.

The strategy aims to provide an integrated and complete knowledge acquisition process which is expected to ease the bottleneck in constructing a knowledge-based system to support design evaluation in PCD. The knowledge acquisition process will be developed with a well-organized knowledge representation scheme to facilitate the effectiveness of rule generation. In this respect, the approach starts from acquiring the relevant design knowledge from appropriate knowledge sources, followed by employing a knowledge sorting process to organize the acquired knowledge in a taxonomic manner. Based on the generated taxonomic tree, a series of matrix representation and mapping processes will be performed to extract rules effectively.

1.3 Significance

Product design plays an extremely important role in creating the competitive edge and making a difference for business organizations to differentiate their products from competitors. To develop a successful product, the early stage of product development, i.e. product concept development (PCD), is crucial and hence deserves much more attention. A product concept is an approximate description of the technology, working principles and form of the product that will be generated. The degree to which a product satisfies customers and can be commercialized depends to a large measure on the quality of the underlying concept (Ulrich and Eppinger 2004). Furthermore, design decisions made during PCD have a significant influence on product development cost, time to market, product quality and flexibility of a product (Yan 2003). In this work, a novel design evolution approach to facilitate the process of product concept development in NPD is proposed. It emphasizes improving the indexing and retrieval mechanism in CBR methodology so as to exploit the potential of effective design reuse. This work also paves an avenue for designers to acquire innovative solutions for generating new product concept that meets the ever-changing customer demands, based on the partially available existing design information. In addition, it explores a rational mathematic model to facilitate the knowledge acquisition process in constructing knowledge-based systems to support design evaluation. The economic implication of this work is likely to enhance design efficiency and product quality/performance in new product development and to reduce design time and cost by effectively making use of historical design data.

1.4 Scope

This work concentrates on the conceptual design stage in product design, which is regarded as a process starting from identifying the needs of the target market and ending at the beginning of system-level design. It provides a generalized methodology to facilitate product conceptualization in new product development. With the proposed approach, a new design problem can be effectively taken care of and satisfactory design alternatives can be obtained through a design evolution process. The scope of the research involves the following:

- (1) Establishment of an indexing and retrieval scheme for design evolution via information reuse*

The proposed approach focuses on the conceptual design stage, aiming to improve the efficiency of design evolution for product conceptualization using historical design data. It focuses on the issues of indexing and case retrieval on prior design knowledge during design information reuse.

- (2) Enhancement of a CBR design evolution approach with the capability of generating innovative solutions*

An innovative adaptation procedure (IAP) will be established to complement a proposed design evolution approach. In so doing, TRIZ techniques will be incorporated into a CBR methodology to help put forward innovative solutions for product concept development.

- (3) Exploration of a mathematic model to facilitate the rule extraction process in knowledge acquisition*

In this work, the rule generation in knowledge acquisition is defined as a mapping process between a conditions domain and a conclusions domain. It

focuses on modeling the rule extraction process mathematically taking consideration of the interrelationships between the different knowledge elements which are used to form the rules.

1.5 Organization of the Thesis

The thesis is organized into seven chapters. Chapter 1 introduces the background of product concept development, and outlines the objective, significance and scope of this work. Chapter 2 reviews methodologies and systems in product design evolution and evaluation, covering domains related to this work, and discusses the limitations of previous studies for improvement. Chapter 3 gives an overview of the proposed system framework and approaches utilized in this work. It intends to provide a general view and orient the reader in what is covered in more detail in the subsequent chapters. Chapter 4 describes a feature-based design evolution approach to facilitate the process of product concept development in NPD. A case study of wood golf club design is conducted to illustrate the capability of the proposed approach. Chapter 5 presents a TRIZ-enhanced innovative adaptation procedure as a complementary model to enable the proposed design evolution approach to generate innovative design solutions. A case study on hard disk drive design is adopted to demonstrate the procedure. Chapter 6 investigates into a novel knowledge acquisition approach, which is proposed for building a knowledge-based system that provides knowledge support to the multi-attribute design evaluation process. A case study on primarily diagnosing automotive systems is employed to illustrate the performance of the approach. Chapter 7 summarizes the conclusions and main contributions achieved in this work. Suggestions for future work are also provided.

The organization of the thesis is shown in Figure 1.2.

Figure 1.2 Organization of the thesis

Chapter 2 Literature Review

This study focuses on investigating the strategies for product conceptualization, especially on conceptual design evolution and knowledge handling during concept development. In previous research, much work has been done in these areas. This chapter presents a comprehensive review on relevant work. It starts with reviewing the basic knowledge of the conceptual design and the traditional methodologies for concept development and selection, followed by a more in-depth review on existing approaches for design information reuse and recovery in design evolution. Next, the research of product innovation in new product development is reviewed. After that, some special techniques such as artificial intelligence for product development are discussed. Finally, a discussion on these techniques and the motivations for the author to carry out this work are provided.

2.1 Methodologies for Design Evolution and Evaluation

2.1.1 Overview of Conceptual Design

Conceptual design is an integral phase of product design and development. The purpose of this phase is to develop the initial technical concept and rough product specifications by incorporating a company's technological strengths and marketing's assessment of the customer's requirements (Jiao 2004). This stage of design involves a series of steps for transforming the design requirements into a product concept. In this phase, a number of viable product concepts which satisfy the design requirements are generated and then evaluated to decide on the most suitable ones for further development. Hence, conceptual design can be seen as a two-stage process consisting of

concept generation and concept selection. Figure 2.1 shows the phase of conceptual design in the whole product design process.

Figure 2.1 Phase of conceptual design (Pahl and Beitz 1996, Ulrich and Eppinger 2004)

The goal of concept generation is to thoroughly explore the space of product concepts that may meet the customer needs. Once a suitable number of concepts have been generated, it is necessary to choose the most suitable designs to fulfill the design requirements. Following the concept generation, concept selection is the activity in which various product concepts are analyzed and sequentially eliminated to identify the most promising concepts. Several iterations are usually required in this process and additional concept generation and refinement may be initiated.

2.1.2 Concept Development and Selection Methods

Concept development and selection is the process of evaluating concepts with respect to customer needs and other criteria, and selecting one or more concepts for further investigation or development. It is an integral part of the product development process. In the past few decades, researchers have conducted a number of studies on the concept development and selection process. It has been recognized that the most popular methods

for concept selection include Pugh concept selection, concept screening and scoring, morphological analysis and catalog design. These methods are discussed as follows.

Pugh's Concept Selection Method

Design concept selection process can be applied with the decision-making tool developed by Pugh (1990). The tool, known as Pugh charts, is one of the most effective known tools for preliminary concept selection when there is minimal information available. It is also effective as the information quality increases and the selection scale is refined (Otto and Wood 2001). This method compares each concept relative to a reference or datum concept. For each criterion, the user/designer determines whether the concept in question is better than, poorer than or the same as the reference concept. Hence, it is a relative comparison technique and usually involves successive rounds of examination and deliberation. As summarized by Otto and Wood (2001), the main steps included in Pugh's concept selection method are:

Step 1: Establish criteria and alternatives and form concept selection matrix.

Step 2: Select a datum for reference concept.

Step 3: Rate and assess the concepts.

Step 4: Rank and order the alternatives.

Step 5: Combine and improve the concepts

Step 6: Iteration and solution

Numerous designers have used Pugh's concept selection method for conceptual design evaluation to select the promising design concepts (Baxter 1995, Pugh 1996, Huang and Mak 1999, Hsiao 1999, Ulrich and Eppinger 2000, Dieter 2000, Islam 2004). However, experience shows that many designers feel uncomfortable with adopting the highest scoring result, because Pugh's concept selection method does not put faith in the

obtained result. Thus, Wang (2002) suggested a way to extend the Pugh's concept selection method with fuzzy sets theory to measure the quality of a chosen concept. The performance ratings of a design concept are represented with linguistic terms that can be further characterized with fuzzy sets. The proposed concept selection model produces a partial order of design concepts based on the valued preference relations between different design concepts. The best design concepts can be identified for continuous improvement or further development in the next design stage.

Islam (2004) presented a methodology which incorporated quality function deployment (QFD), tree diagrams and Pugh's concept selection method to assist extracting the dimensional requirements for a product from customers' needs. Another discussion on the linkage between QFD and Pugh's concept selection can be found in Al-Mashari et al. (2005).

The Pugh's concept selection method is an iterative and evolutionary design process, during which a dominant concept arrives eventually. Nevertheless, this method is usually performed by the design team members manually and much patience is required during the iterative and continued process and especially when the alternative population is relatively large. As such, it is not so practical for design evolution and evaluation in new product development in today's competitive business environment.

Concept Screening and Scoring

A two-stage concept selection methodology, i.e. concept screening (Pugh 1990) and concept scoring, was presented in Ulrich and Eppinger (1995; 2004). Each stage is supported by a decision matrix, which is used by the design team to rate, rank, and select the best concepts.

During concept screening, rough initial concepts are evaluated with respect to a common reference concept using the screening matrix. Since detailed quantitative comparisons are difficult to obtain and maybe misleading at the early stage, a coarse comparative rating system is used. After some alternatives are eliminated, the team may choose to move on to concept scoring and conduct more detailed analysis and finer quantitative evaluation of the remaining concepts using the scoring matrix as a guide. In the stage of concept scoring, the relative importance of the selection criteria will be weighted and each concept will be assessed with respect to each criterion. The concept scores are determined by the weighted sum of the ratings and calculated as follows:

$$S_j = \sum_{i=1}^n r_{ij} w_i \quad (2-1)$$

where r_{ij} : raw ratings of concept j for the i th criterion

w_i : weighting for i th criterion

n : number of criterion

S_j : total score for concept j

Finally, each concept is given a rank corresponding to its total score. Some research of product design regarding concept screening and scoring methodology can be found in Huang (1999), where twenty items are recommended to organize customer requirements, and the relative importance of those items is evaluated using pair-wise comparison method. The weighting factors for all recommended items are then calculated. Graham et al. (2001) developed a computer-aided design (CAD) system that aids designers in generating the form of a product using genetic algorithm. In the system, user scoring of each object or an objective function is used to determine which objects are considered to be the 'fittest', and thus likely to become parents of the next generation. Basically, this approach has the limitation similar to the Pugh's concept selection method, i.e. much

time and patience are required during the manual selection process. Hence, it is more applicable to relatively small selection problems, but not promising to reduce the product development cost and time in NPD.

Morphological Analysis

Morphological analysis (Norris 1963) is a systematic method and tool that provides a structured evolution and combination of concepts in product design. For solving a complicated problem, this means it could list out all kinds of solution combinations and then analyze, synthesize and eventually obtain the promising design concepts. The main steps of applying this method in concept selection process are:

Step 1: Obtain all kinds of possible solution combinations.

Step 2: Thin out the whole theoretically possible solution space using fit analysis and compatibility analysis.

Step 3: Obtain promising design alternatives after an in-depth search process.

During fit analysis, the options are evaluated against the functional requirements and those with low “fit” levels or average values below a threshold are dropped from the list. On the other hand, in order to remove incompatible solutions, all compatibilities within possible overall solutions are to be checked with a compatibility scale and those whose values are below the threshold will be dropped. However, this approach has some limitations in the shortlist process and fit analysis procedure. In many practical cases, it may not be reliable to drop the options whose “fit” values are below a certain threshold because these options maybe even better than those with higher “fit” values when combined with others to firm up the whole solution. In addition, the shortlist process may cause a combinatorial explosion problem making the search process very time-consuming and effort-exhausting if performed manually.

A number of morphological analysis applications in engineering design can be found in the literature (e.g. French 1985, Pahl and Beitz 1996, Huang and Mak 1999, Belaziz et al. 2000, Cross 2000, Dieter 2000, Otto and Wood 2001, Devdas Shetty 2002, Lai et al. 2005). Belaziz et al. (2000) presented a form of feature-based tool to aid the integration of analysis in the design process. It allows producing an analysis model out of a part solid model. This tool is based on a morphological analysis of the solid model followed by a two-phase process: simplification and idealization. Huang and Mak (1999) proposed an approach to implement the morphological analysis method on the Internet. In the approach, the incorporation of various decision-making activities of concept design into one integrated web-based system allows the designer to choose the most appropriate idea from potential alternatives in an objective and systematic way. However, as stated by the authors of the paper, the proposed system suffered a number of limitations. The two major limitations are: (1) no sound strategies are available for short-listing the solutions in the search field, and (2) the combinatorial explosion problem is still not solved and currently relies on the control of a human user.

Lai et al. (2005) presented an approach to determine the best design combination of product form elements for matching a given product image. In the approach, an experimental study on the form design of mobile phones is conducted and a morphological analysis is used to extract form elements from these sample mobile phones. Some case studies and examples, such as the potato harvesting machine and forklift truck, applying morphological analysis in engineering design were reported in Cross (2000).

Catalog Design Method for Concept Selection

Catalog design is the engineering task in which a configuration is created by assembling off-the-shelf components into a functional system (Carlson-skalak et al. 1998). During the procedure of catalog design, the system is assembled by selecting standard components from catalogs of available components (Vadde et al. 1992). Thus, this approach tries to select alternatives through the combination of different components in designing a product.

Some researchers have adopted the catalog design method for product design and most of the efforts focused on component selection and configuration. Carlson-skalak et al. (1998) described an evolutionary algorithm for catalog design. This algorithm is based on genetic algorithms. It uses an object-oriented coding scheme to represent a design and adopts unique crossover and mutation operators. The approach allows the consideration of alternate configurations, which can be evolved to make the best use of the available components. Vadde et al. (1995) extended the traditional compromise decision support problem to represent uncertainty for catalog design using fuzzy sets theory to model imprecision and Bayesian statistics to model stochastic information. Singh and Betting (2004) reported the evaluation and comparison on different schemes for capturing the attributes of assembly interfaces of catalog parts and appending that information to the solid part models in order to allow for automated exploration of alternative configurations in a computer-aided design environment.

Although catalog design works very effectively during the stage of conceptual design for small scale concept selection situation, it suffers the combinatorial explosion problem when the design solution space becomes overwhelming. When the number of solution principles and sub-functions to meet the overall requirements of the whole system is relatively large, this phenomenon is prone to occur. Also, this approach has the

limitation that it needs a complete set of reasoning knowledge which may not be possible to be obtained during the early stages of product design.

2.2 Information Reuse in Evolutionary Product Design

2.2.1 Design Information Reuse and Recovery

The major goals of corporations nowadays are to significantly reduce the costs and time-to-market of their products in order to survive the intensive competition. To meet these goals, reuse of existing design information is essential for new product development and much more cost-effective than regenerating that information. In the perspective of evolutionary product design, a number of researchers have made contributions to design information reuse and recovery. Vinoid et al. (1992) conducted an experimental research on the nature of design information reuse during the redesign process. A design reuse model for engineering design which consists of six knowledge-related components and three processes was developed by Duffy et al. (1995). In the reuse model, the processes are proposed as the essential aspects of supporting reuse and used to indicate the failure of existing support to recognize the totality of design reuse. Otto and Wood (1998) proposed a ten step reverse engineering and redesign methodology with three primary phases for design information reuse. Each phase provides a clear set of tasks to seek new product configurations, combining and developing different techniques, such as black box model and functional analysis. The reverse and redesign process provided a systematic approach for industrial applications and engineering design education. Troussier et al. (1999) dealt with the different uses of mechanical analysis in embodiment design and proposed an approach of information structuring in order to allow mechanical analysis reuse during engineering design.

Xu (2000) adopted neural networks to learn the empirical or experimental design information that already exist, and then to predict the occurrence of burning events and the voltage endurance of new positive temperature coefficient devices at the design stage. Ball et al. (2001) described the development of a design-reuse system that maximizes the benefits of rationale capture and information retrieval whilst minimizing the costs to the designer that might arise from disruption to natural design work. Tay and Gu (2003) proposed a function-based approach for supporting conceptual design in the context of evolutionary product development. The main objective of this approach is to improve a designer's productivity by the effective reuse of existing design information in design alternative identification, evaluation, and modification. The focus was put on the multilevel function-based product information organization, which facilitates design alternative identification and adoption in the redesign phase according to their functional content. However, this approach mapped the function topology into physical structure manually and hence would not be efficient when the design problem becomes large and complex. Zhang et al. (2003) reported an approach to the reuse of standard components in progressive die design systems. In the approach, the standard components are represented with their order names and classified according to their generic information, i.e. their functions and the geometrical characteristics of their mating relationships with other components. When a user designs a particular die structure for a certain function, some design options and requirements are used as inputs. Using this information, the generic information of the standard component is derived and the order name is selected.

Ong and Guo (2004) developed an integrated framework to computerize the design reuse process and a prototype online design reuse system was implemented. The system embodies the evolutionary process of design reuse and supports design reuse from the

original requirements through the concept and embodiment stage to achieve a final detailed design.

The reuse of previous design knowledge is a potentially important way to improve product design efficiency. In practice, however, design reuse suffers a lot of difficulties, including those associated with the indexing, retrieval, understanding and modification of prior design knowledge. Hence, a powerful and effective methodology for design reuse is highly desirable in the process of evolving design concepts.

2.2.2 Case-Based Reasoning for Evolutionary Design

Among various methodologies used in the design information reuse, case-based reasoning (CBR) is a promising methodology for solving many complex engineering design problems. CBR employs past problem-solving experiences when dealing with new problems. In design, the use of cases allows the designers to concentrate on merging and adapting old solutions, and suggest efficient ways of constructing solutions and warning of the potential failure, allowing reasoners to anticipate or avoid problems that have arisen in the past. Moreover, CBR has more advantages on formalization of the generalized design experience than other artificial intelligence methods, such as rule-based and model-based reasoning techniques (Kolodner 1993, Mary and Pearl 1997). Several other advantages of the case-based reasoning approach are summarized by Jeng and Liang (1995). Firstly, it provides a means for storing experience in expert systems, which relies mainly on experience but not on the rules. Furthermore, the case-based approach enables knowledge engineers to handle poorly structured domains. Finally, the case-based approach makes self-learning of expert system easier because learning is only a matter of representing and storing previously solved cases in the case base.

Case-based reasoning has grown wide spread interest since 1990's and is a mature subfield of artificial intelligence. CBR means adapting old solutions to meet new demands, using old cases to explain new situations, using old cases to criticize new solutions, or reasoning from precedents to interpret a new situation or create an equitable solution to a new problem (Kolodner 1993). Basically, all CBR methods mainly consist of the following common processes when solving new problems:

- Represent the design cases for the case-based design system;
- Retrieve the most similar cases which are compared with the library of past cases;
- Reuse the retrieved cases to solve the current problem;
- Revise and adapt the proposed solution if necessary;
- Retain the final solution as part of a new case.

Much research on case-based reasoning in product design has been conducted in past years. Tseng and Jiao (1997) proposed a case-based evolutionary design (CBED) approach to achieving the goal of mass customization. Issues associated with the approach are discussed in the context of CBED for mass customization, involving case content, memory organization, indexing, retrieval, and adaptation. Zeid et al. (1997) postulated a CBR approach to deal with plans for disassembly problems based on the fundamental principle that problem solving can benefit from solutions to past problems that have been attempted. Suh et al. (1998) presented an approach for developing new product properties that can effectively support all the steps in the design process from storing past cases, through retrieving similar cases, to adapting the retrieved case for the new product. Liao et al. (1998) reviewed a number of methods for measuring similarity during the CBR process and proposed a hybrid similarity measure for comparing cases with a mixture of crisp and fuzzy features.

Rosenman (2000) developed a case-based model of design using an evolutionary approach for the adaptation of previously stored design solutions. The model aims to provide a solution to the current problem of adaptation in case-based design. Kraslawski and Kudra (2001) suggested two adaptation methods in CBR based on fuzzy and rough sets for the design of drying equipment. Vong et al. (2002) described the application of CBR, which helps to solve problems using past experience, to a hydraulic circuit design for production machines. Chen et al. (2002c) presented a parallelized indexing method for efficiently retrieving similar cases in a large-scale CBR system.

Qin and Regli (2003) conducted a case study of how to apply CBR to a specific engineering problem: mechanical bearing design. A system is developed that retrieves previous design cases from a case repository and uses adaptation techniques to modify those cases to satisfy the current problem requirements. The application of CBR to decision support for design engineers during the early phases of new product development, in a concurrent engineering environment can be found in Belecheanu et al. (2003). The paper discussed the rationale of using CBR, emphasizing its suitability for ill-defined and unstructured problems, in comparison with traditional knowledge-based systems. Roh et al. (2003) proposed a three-step model, which combines a collaborating filtering (CF) algorithm with two machine learning processes, i.e. self-organizing map and case-based reasoning, for CF recommendation. Madhusudan et al. (2004) presented a CBR framework to support workflow modeling and design by adapting workflow cases from a repository of process models.

Chougule and Ravi (2005) developed a computer-aided casting process-planning system using CBR. The system has been implemented in a web-based framework to enable early manufacturability assessment of castings through cost estimation and process simulation. Tseng et al. (2005) conducted a study on applying CBR for product

configuration in mass customization environments. In the study, the CBR algorithm was used to construct and generate the right bill of material that fits the new situation. The aim of the study is to effectively reduce the time and cost of design under the premise to manufacture an accurate new product.

Boyle et al. (2006) suggested a computer-aided fixture design methodology that adopted a rigorous approach to define indexing attributes based upon axiomatic design functional requirement decomposition. In the methodology, a design requirement is decomposed in terms of functional requirements, physical solutions are retrieved and adapted for each individual requirement, and the design is then reconstituted to form a complete fixture design. Han and Lee (2006) presented a case-based approach for reusing previous design concepts in conceptual synthesis of mechanisms. The basic idea of the approach is to provide a computational framework for design synthesis by imitating the common design method of reviewing past designs to obtain solution concepts for a new design problem. The reuse of prior design concepts is accomplished within a framework of CBR.

As reviewed above, CBR has numerous applications in design. However, most of those applications are only effective in some small design cases. When the new design problem involves a huge case database, the similar case search and retrieval process will be very time- and effort-consuming without an efficient guiding methodology. Moreover, the similarity measurement and matching process are also plagued with difficulties such as imprecision and uncertainty emerging at the preliminary design stage during the product design process.

2.3 Product Innovation in New Product Development

In the increasingly competitive and rapidly changing global business environment, organizations rely on product innovation for success and survival. Design innovation in new product development is one of the most important competitive challenges facing firms (Kessler 2000). Products are required to be constantly created, enhanced and adapted to meet new market demands. Due to the rising significance of product innovation, there have been a number of publications contributed to this topic.

Roy and Riedel (1997) investigated into a conceptual analysis of the role of design and innovation in product competition. The concepts are employed to conduct an analysis of a sample of new and redesigned products. Legardeur et al. (2003) designed a new tool, so called Innovation Development and Diffusion (ID²) to coordinate the development of new solutions during the early phases of design projects. The tool is geared to help innovative solutions emerge, and consolidate these solutions by circulating them to the company's different specialists. It is based on a concepts/criteria table enabling the viewpoints of the different actors involved to be summarized during the design phase. Cormican and O'Sullivan (2003) reported the concept of knowledge management for product innovation and presented a collaborative knowledge management tool specifically designed to help manage a portfolio of product innovation projects in a distributed environment. Hsiao and Chou (2004) proposed a design method based on the naturally sensuous ability of human beings. In the paper, a creativity-based design process integrating some systematic design methodologies was proposed for innovative product design.

Carayannis and Coleman (2005) explored the impact creative engineering design methodologies would have on the development of conceptual system designs for technically complex systems. A creative engineering design method to increase the

innovativeness of these technically complex systems was proposed. This design methodology was tailored to work with the processes, content, context, and provide the desired impact. Sharma (2005) studied three topics, viz. collaboration, product development and innovation, separately and then integrated these elements into a model, so called collaborative product innovation (CPI), via a product life cycle management framework. Liu and Tang (2006) presented a novel evolutionary design approach in a multi-agent design environment. Rather than implementing the innovative design by computers, this environment was used to stimulate the imagination of designers and expand their thinking space. It aimed at exploring a feasible and useful evolutionary approach in a distributed environment that will give the designers concrete help for creative designs.

By incorporating TRIZ, the Russian acronym of the Theory of Inventive Problem Solving, Wang et al. (2005) developed an algorithm that integrates QFD (Quality Function Deployment) and TRIZ effectively for an innovative design process. The core of the algorithm is to identify the contradictions in TRIZ by defining rules based on HOQ (house of quality) in QFD. Kobayashi (2006) suggested an innovative product eco-design methodology based on a LCP (life cycle planning) framework. For supporting innovation at the product level, idea generation using TRIZ, design uncertainty evaluation, and an eco-efficiency indicator of a product was proposed. TRIZ is a powerful tool that can be used for problem solving in the design innovation process. More work that employs TRIZ for design innovation is reviewed in Chapter 6.

Based on the above discussion on product innovation in NPD, it has been established that design innovation is an essential factor in getting the edge over competing products for business organizations. Design innovation enables the product to place itself on higher ground with fewer competitors and a more welcoming market. As such, in order

to carry out a successful product conceptualization, product innovation is of high necessity to be incorporated into the design process.

2.4 Artificial Intelligence (Knowledge-based Approach) in Product Development

2.4.1 Knowledge-Based System in Design Evolution

Knowledge-based system has been widely applied in the product design and development process since the past few decades. Kusiak et al. (1991) developed a rule-based expert system to decompose design specifications, requirements and functions as well as match the requirements with functions. The design specifications are represented with an AND/OR logical tree, which allows for the representation of alternative solutions in the process of formulating design specifications. Subsequently, a generalized search algorithm incorporating the principles of A* algorithms jointly with production rules is used to guide the search through the solution space. The search of the AND/OR tree is also known in artificial intelligence as the explicit search (Nilsson 1980). The advantage of explicit search is that it allows viewing of a number of alternative solutions. However, as pointed out by Kusiak et al. (1991), the number of potential solutions generated in a complex system will be very large and a huge design solution space will be formed. Furthermore, the qualitative information for selecting alternative solutions is not considered in the explicit search. Steadman and Pell (1995) proposed a prototype expert system for injection-molded plastic parts, which demonstrated the utility of expert systems for design applications. The prototype was implemented in an object-oriented and rule-based environment.

Tor et al. (2002) presented a behaviour-driven functional modeling framework for functional design of mechanical products with a rule-based causal behavioural reasoning step to guide the design process. A new behavioural representation was developed to facilitate causal behavioural reasoning, with which the interconnected physical behaviours can be reasoned out from a desired function.

Zhang et al. (2001; 2002) suggested a knowledge-based functional design automation system for redundant engineering products. In the system, an integrated knowledge representation scheme combines rule-based and object-oriented representation methods to represent functions and function related design characteristics in an intelligent design environment. Another application of knowledge-based system in design can be found in Zha et al. (2001), in which a design expert system was developed to support a top-down design for assembly.

Wang and Yan (2002) postulated a rule-based computerized symbolic carrier regeneration method for mechanical system conceptual design. In this method, design requirements of mechanical system are expressed by regeneration rules which are composed mainly of symbols and logical codes. Lee et al. (2005) reported an object-based knowledge integration system, which supports the early stages of product development. This proposed system characterizes the “dynamic” information exchange capability through its distinct features to include executable tasks within product information script that is being utilized in various functional groups, thereby introducing the action items to be carried out in relevant areas.

As for product design evolution, rule-based and model-based reasoning techniques have been commonly applied to build design automation or design decision support systems (Maher and Pu 1997). Although such systems have achieved a certain degree of success, difficulties were encountered in terms of formalizing such generalized design

experiences as rules, logic, and models. Moreover, some potential good design solutions, which might become better ones when combined with others, are ruled out too early.

2.4.2 Knowledge-Based System in Design Evaluation

Artificial intelligence (AI) has been widely applied in design evaluation to facilitate the decision making process. Most of these applications aimed to establish a knowledge-based system or model an “expert” in the design evaluation process. Rosenman (1990) demonstrated the applicability of expert systems to design analysis and evaluation. For this purpose, three examples of applying expert systems for design analysis and evaluation in different domains were described.

Tian et al. (1994) suggested an expert system for design, which explicitly addressed the individual user's attitude towards uncertainty. The attitude is assessed through a utility function by a spreadsheet user-interface, which linked those evaluation results to a heuristic rule base. The heuristic rule base contained only technical expertise used to determine a set of feasible alternatives, while the process of evaluating and ranking these feasible alternatives is driven by the user's individual preferences and risk-taking profile.

Chin and Wong (1996) postulated a framework of a knowledge-based evaluation system for new product concepts focusing on the conceptual design stage. Ishii and Barkan (1987) proposed a knowledge-based simultaneous engineering approach using Design Compatibility Analysis (DCA). The framework of DCA provided a model of design reviews in which design alternatives could be judged from various life-cycle aspects by experts with different responsibilities (Ishii 1996). The main objectives of DCA is to use the compatibility data to compute an overall “goodness” of designs, give reasons, and provide suggestions for improvements. However, the approach did not

consider the issues of selecting the most promising design alternatives based on the measurement of the degree to which each alternative satisfies each of the weighted objectives.

Green (2000) highlighted the importance of combining separate models into a possible future integrated design evaluation tool based on the philosophy that no one model or method should be relied upon during the evaluation activity. In this respect, three competing theoretical models were compared and validated against a common data set in order to support an approach that triangulates the outputs from a number of models to achieve robust evaluation of competing design concepts from a supporting knowledge base.

Moulitanitis et al. (2004) presented an approach for modeling the evaluation process in the conceptual design phase. The generated candidate solutions are evaluated by calculating a score that is based on specific criteria that are presented as elements of a vector. Das and Gami (2004) established a set of expert rules for manufacturability analysis of misalignment defects in assembly during product design. The expert rules were developed via an analytical simulation of the assembly process and an investigation of assembly practice. Zha et al. (2004) reported a knowledge decision support approach to product family design evaluation and selection for mass customization process. A knowledge support framework and its relevant technologies are developed for module-based product family design.

Zha and Du (2006) developed a web-based knowledge-intensive distributed modeling and evaluation framework which intends to link distributed, heterogeneous knowledge-based design tools and assists designers in evaluating design alternatives, visualizing trade-offs, finding optimal solutions and making decisions on the web. The framework also enables designers to build integrated design models using both the local and

distributed resources and cooperate by exchanging services. Park and Seo (2006) constructed a knowledge-based approximate life cycle assessment system to assess the environmental impacts of product design alternatives. It aims at improving the environmental efficiency of a product using artificial neural networks, which consist of high-level product attributes and life cycle assessment results.

Additionally, an AI-based concurrent product design evaluation system (CONDENSE) was developed by Chen et al. (2001). The system was established based on a blackboard architecture and it contained two main models: a qualitative aspect evaluation model and a quantitative aspect evaluation model. The former is used to help determine the design specifications and the latter is used to provide the information on performance, assemblability, manufacturability and costs with respect to design alternatives to facilitate design selection. More information related to this system can be found in Chen (1996) and Chen et al. (1999, 2000, 2002b). Nevertheless, as stated by Chen et al. (2001), the current system did not provide the design selection module and some improvements have to be made before a generic system can be fully developed.

2.4.3 Blackboard System

A blackboard system is a special type of knowledge-based system, which is based on a metaphor of a team of experts co-operating to solve a difficult problem. Each expert is competent in different domains which are relevant to the problem, and they communicate ideas for the solution of a problem by writing text or drawing diagram. The problem is solved when the experts agree that an adequate solution has emerged on the blackboard (Craig 1991).

Blackboard system is particularly suitable to support the philosophy of concurrent engineering. It utilizes multiple knowledge bases that interact through a “blackboard”, much like the way team members interact in concurrent engineering (Chen et al. 2001). Chen and Occeña (2000) described a non-directed graph decomposition procedure for transforming the knowledge into knowledge bases as part of an approach for developing a product design blackboard expert system. Based on this procedure, a further study was conducted to investigate the relationship between graph decomposition and the resultant blackboard system (Chen et al. 2002b). In addition, McManus and Bynum (1996) proposed a formal model that provides a consistent method for describing a blackboard system. Kwong (1997) presented a blackboard-based system for concurrent process design of injection moulding. Readle and Henry (1998) used a blackboard system, which contains knowledge sources with algorithmic, fuzzy-logic and evolutionary reasoning, to identify the time-varying process model in an adaptive control system. Roy and Liao (1998) proposed an approach to apply the blackboard framework as a problem solving model to realize a cooperative fixture design system. The creation of functional knowledge sources for fixture design was described in detail and their applications in a cooperative problem solving environment were demonstrated.

Chau and Albermani (2004) proposed a hybrid knowledge representation scheme in a blackboard knowledge-based system for liquid retaining structure design. In developing the knowledge-based system, the knowledge representation schemes such as production rules, object-oriented programming and procedural methods, are employed to express engineering heuristics and standard design knowledge. Tor et al. (2005) presented a knowledge-based blackboard framework for stamping process planning. The proposed approach aims to speed up the progressive die design process by automating the strip layout design.

Besides, a number of researchers have attempted to integrate blackboard system with agent technology to increase the capability of distributed problem solving in product development (e.g. Weiss and Stetter 1992, Mani 1994, Lander 1997, Cao and Jiang 1998, Kao et al. 2002, Jiang et al. 2004). Cao and Jiang (1998) reported the design and implementation of a multi-agent cooperative problem solving expert system tool. A blackboard system was adopted in the system as a data sharing and information exchanging center, to coordinate cooperative problem solving. Kao et al. (2002) postulated a framework of blackboard-based multi-agent system to facilitate the communication and coordination of design projects. However, the framework has some limitations during the implementation, which merits further improvements. Jiang et al. (2004) suggested a model for constructing an agent blackboard communication architecture based on graph theory. The model computed the location of central blackboard or sub-blackboards based on median location method, and computed the communication topology among sub-blackboards.

2.4.4 Knowledge Acquisition for Product Development

In constructing a knowledge-based system, knowledge acquisition covers the activities of collecting knowledge and transforming it into a form that can be processed by a computer. It is a very important phase and has been recognized as the bottleneck of building a knowledge-based system. Traditionally, there are two common approaches to knowledge acquisition: (1) acquire knowledge directly from the domain expert(s); (2) acquire knowledge through the historical records including rule induction, data mining techniques, etc. Both of the two approaches are widely employed and sometimes used in combination in developing rule-based systems (Ignizio 1991). The two common

approaches and previous work to facilitate the knowledge acquisition process are reviewed in this section.

(i) Acquire knowledge directly from the domain expert

A number of guidelines have been reported in the published literature for knowledge acquisition directly from the domain expert(s) (Prerau 1987, Surko1989, Benfer et al. 1991, Ignizio 1991). Ignizio's is one of the most comprehensive and generic. It comprises the following consecutive steps:

- (1) Selection of the domain
- (2) Selection of the knowledge engineers
- (3) Selection of the domain experts
- (4) The initial meeting with the domain experts
- (5) Learning the background of the domain
- (6) Organization of follow-up meetings
- (7) Conduct of the follow-up meetings
- (8) Documentation of each meeting

The guidelines in this approach were obtained from observations of those who have used such procedures in earlier efforts. The guidelines are understandingly incomplete and subjective. The only way that one can truly appreciate this task is through experience (Chen and Occeña 1999). In the technological level of this approach, Wu et al. (2000) developed a knowledge acquisition toolkit using a goal-driven strategy. The toolkit was designed to help eliciting and storing experts' declarative and procedural knowledge in knowledge bases for a user-defined domain. Xing et al. (2003) presented an integrated (automatic/ interactive) knowledge acquisition method to rapidly develop knowledge-based systems. The method integrated domain experts' subjective knowledge with

general knowledge embedded in examples and relied on several key algorithms including data discretization, rule extraction, and rule simplification and pruning.

(ii) Acquire knowledge from historical records

To convert an existing and appropriate database into a set of production rules is an alternative to acquire knowledge directly from the domain expert(s). A variety of research has been conducted on the problem of knowledge elicitation via rule induction (Quinlan 1983; 1987, Thompson and Thompson 1986, Carter and Catlett 1987). The following is one of the most popular approaches to rule generation from data (Ignizio 1991):

- (1) Identification of objects, attributes and values
- (2) Construction of a decision tree based on objects, attributes and values
- (3) Generation of rules from the tree

In constructing decision trees, different trees may result from exactly the same data and thus different production rule sets will be generated. In order to deal with this problem, a much more systematic way to develop an efficient production rule set is required. One approach known as ID3 algorithm has been proposed to address this problem (Quinlan 1983). The ID3 algorithm is a data-driven approach that uses a top-down induction technique to generate its decision trees from a set of data. Instead of the arbitrary procedure used in tree construction, the ID3 algorithm constructs trees based on a measure of the entropy of each attribute. The higher the entropy of an attribute, the more uncertain its value. Thus, attributes are selected in order of increasing entropy, with the root node of the tree corresponding to the attribute with lowest entropy value. The inductive bias implicit in ID3 includes a preference for smaller trees; that is, its search

through the hypothesis space grows the tree only as large as needed in order to classify the available training examples.

The ID3 algorithm, which was developed for determinate data, has been extended to deal with statistical data by Mingers (1987). Grzymala-Busse (1995) presented results of experiments to show how machine learning methods, one of which is ID3 algorithm, are useful for rule induction in the process of knowledge acquisition for expert systems. In addition, a system known as LERS (Learning from Examples based on Rough Sets) was developed for inductive learning. LERS is able to deal with inconsistencies in the training data but become impractical when the training data set is huge. Khoo and Zhai (2001) developed a prototype system that discovers rules from inconsistent empirical data using an integrated approach that combines rough set theory, genetic algorithms and Boolean algebra for inductive learning. In the implementation level, Shao et al. (2001) suggested an application of ID3 algorithm in knowledge acquisition for the tolerance design of injection-molded parts. Wu and Kao (2002) proposed a mechanism that combines the utilization of membership function and the ID3 algorithm to generate decision rules from a set of data. Based on the mechanism, an ID3 Rule Generation System was developed to support the generation of a decision tree and correspondingly exact/approximate rules.

However, as pointed out by Gonzalez and Dankel (1993), although the inductive tools for knowledge acquisition can often assist in the development of a knowledge-based system, these tools are not useful in all cases, but are appropriate for classifying tasks only. The inductive tools are often most helpful in the development of small systems. Moreover, rule induction systems have stood accused of forming only hyper-rectangular regions in the example space and not recognizing exceptions in small, low frequency sections of the domain (Cercone et al. 1999). In addition, Wong (1998) summarized that

the typical existing learning systems such as CART (Breimen et al., 1984), C4.5 (Quinlan, 1992), ASSISTANT (Cestnik et al., 1987), AQ15 (Michalski et al., 1986), and CN2 (Clark and Niblett, 1989) used attribute-value language for representing the training examples and the induced knowledge and allowed a finite number of objects in the universe of discourse. This way of representation limits these systems to learn only propositional descriptions in which concepts are described in terms of a fixed number of attributes.

From the aspect of facilitating the efficiency of knowledge acquisition, Chen and Occeña (1999) proposed a so-called knowledge sorting process. The process capitalizes on the relationships between attributes and factors, dependent and independent variables, and interrelationships between attributes. It consists of three major consecutive steps:

- (1) Identification of knowledge sources
- (2) Generation of taxonomic tree (general sorting)
- (3) Organization of acquired knowledge

After a preliminary general sorting (Step2), a more specific sorting process is used to organize the knowledge in three stages according to the relationships of attributes and factors, the interrelationships between factors in both matrix and graph form, and the relationships between dependent and independent variables. This procedure was developed to facilitate the extraction of rules from knowledge in the product design domain. Aiming at attribute reduction and rule generation, Pan et al. (2002) proposed a self-optimizing method based on a difference comparison table for knowledge acquisition. Wu et al. (2000) summarized some knowledge acquisition models, most of which still have some limitations that need to be overcome. The limitations include the errors in the generated rules, knowledge incompleteness, incompatibility, incorrectness etc.

Besides, as a process of extracting desirable knowledge from existing databases for specific purposes, data mining has been widely used in knowledge discovery and rule mining. Catral et al. (2002) reported a data mining system that combines evolutionary and symbolic machine learning methods to extract comprehensible and strong rules from a dataset. The system relies on evolutionary search to highlight strong rules to which symbolic generalization techniques are applied between generations. Tsay and Chang-Chien (2004) presented a cluster and decomposition algorithm for mining association rules. The algorithm combines both the cluster concept and decomposition of larger candidate itemsets. By using the notion of density to capture the characteristics of quantitative attributes, Lian et al. (2005) developed a procedure to locate the dense region and scale up for high-dimensional cases in mining quantitative association rules. In the same vein, some other data mining methodologies have been suggested for knowledge discovery and association rule mining (Berzal et al. 2001, Li et al. 2003, Lee et al. 2005, Hu 2006).

However, although much previous work has contributed to ease the knowledge acquisition bottleneck, the quantitative characteristics of information and interrelationships between various knowledge elements, which is of critical importance in constructing knowledge-based systems, has not been well addressed. Moreover, it is imperative to provide an integrated and complete knowledge acquisition process with well-organized knowledge representation scheme to facilitate the effectiveness of rule generation.

2.5 Discussions

Basically, the above review focuses on product design with special emphasis on the design evolution strategy for product conceptualization in NPD. It covers a broad domain from product concept development and selection, design reuse for evolutionary design, product innovation in NPD as well as artificial intelligence (knowledge-based approach) in product design. Conventional and many well-received methodologies and techniques for product conceptualization in NPD are reviewed. The review enables the gaps pertaining to the research in product conceptualization to be identified, and accordingly, provides the motivations for the subsequent investigations. Based on the literature reviewed in this chapter, the following issues and gaps need to be addressed.

(i) *Design Evolution in PCD*

Product conceptualization is an essential stage in new product development. An efficient design evolution approach to facilitate new product concept development is highly necessary for organizations to deliver products with ultimate performance, low cost, high quality and shorter time-to market. Although a number of methodologies and approaches such as concept development and selection methods, design information reuse and recovery, and artificial intelligence have been contributed to this research area, some limitations still exist as discussed in previous sections. The gaps in research include:

- Lacking of an efficient design evolution methodology for product concept development in NPD.
- The issue of design information reuse for new product development has not been well addressed. It still suffers a lot of difficulties such as indexing, retrieval and modification of prior design knowledge.

- Due to lack of systematic approach to make a design reusable, reuse process remained opportunistic and full potential for successive reuse has not been exploited.

(ii) *Indexing and Retrieval mechanism in CBR*

Among the varieties of methods used in the design information reuse, CBR, which employs past problem-solving experiences to deal with new problems, is a promising methodology for tackling many complex engineering design problems. As pointed out in previous sections, the previous research and applications of CBR in design still have the following limitations and gaps:

- Most of the existing search and retrieval strategies are not efficient when the design case repository is huge and the new design problem is very complex.
- In the indexing mechanism, few of the previous research on CBR system consider the interrelationships among various indexing design features, which may lead to additional difficulties and inefficiency in case retrieval.
- In selecting similar cases, most previous work focused on the search and retrieval of a complete case. However, it is possible that not all the attributes and values corresponding to each design feature in the retrieved cases are desirable. These unsuitable attributes and values may cause extra time and cost during adaptation.

(iii) *Product Innovation in New Product Concept Development*

Although the CBR methodology is promising to deal with new design problem via utilizing previous design experiences, it has difficulties when product innovation is required in NPD. New technical innovations through the development of new technologies and new knowledge derived from the accumulation of existing information should be introduced into the design reuse process for new product development.

However, few of previous studies have considered the innovation aspect in applying CBR. As such, it is worthwhile to investigate into an effective approach to help the CBR systems overcome the weakness in product innovation.

(iv) Knowledge Acquisition Bottleneck

In developing a knowledge-based system for design evaluation, the knowledge acquisition phase is an activity that can be extremely frustrating as well as effort-consuming. Many researchers have attempted to deal with the knowledge acquisition bottleneck. However, it appears that a promising approach for knowledge acquisition is still in demand. As aforementioned, the gaps in the research of knowledge acquisition include:

- An integrated and complete process for knowledge acquisition to facilitate the rule extraction is still lacking.
- The quantitative characteristics of information, which is of critical importance in constructing knowledge-based systems, has not been well addressed.
- An effective methodology to well represent and organize the knowledge acquired from a variety of knowledge sources is highly necessary to make the rule extraction easier.

2.6 Summary

Much research has been contributed to the issue of product design evolution and evaluation process. As a result, there are many methods and techniques available for this purpose. This chapter reviewed the fundamentals of conceptual design, concept development and selection method, design information reuse in evolutionary product

design, product innovation in new product development, as well as the techniques of artificial intelligence such as knowledge-based approach in product design. From the review and the discussions presented in Section 2.5, the gaps between the topics of design evolution and knowledge handling in product conceptualization have been identified. These gaps provide the motivations for the author to carry out an in-depth study on product conceptualization for NPD as summarized below:

- (1) A feature-based approach (see Chapter 4) which employs the techniques of case-based reasoning is proposed to aid design evolution in developing new product concepts. The proposed approach aims to fill up the gaps in design reuse, such as the problems of case indexing and retrieval, in CBR methodology. It also attempts to enable the design engineers to improve the efficiency of product development through reusing historical design experiences. Specifically, a feature relationship decomposition (FRD) procedure is proposed to decompose the complex interrelated design feature structure. Besides, an analytical hierarchy process (AHP) process is used for feature priority determination. For effective case adaptation, a novel feature-based case retrieval process is presented to select the appropriate attribute cases to deal with new design problem. The capability of the proposed approach is illustrated using a case study on wood golf club design.
- (2) A novel model, known as Innovative Adaptation Procedure (IAP) is suggested as a complementary module for the proposed feature-based design evolution approach in Chapter 5. The purpose of IAP is to enable the design evolution approach to come up with innovative design solutions when product innovation is required in NPD. A case study on hard disk drive design is employed to illustrate the performance of the proposed approach.

- (3) A mathematic model, so-called matrix representation and mapping (MRM) approach (see Chapter 6), is explored to analyze and represent the rule extraction process so as to facilitate the knowledge acquisition phase in building a knowledge-based system for design evaluation. The approach aims at tackling the gaps as mentioned in Section 2.4.4. A case study on diagnosing automotive problems is used to demonstrate how the proposed approach works.

Chapter 3 Research Orientation

Base on the relevant work reviewed in Chapter 2, it has been established that (1) the existing systems for PCD are far from being perfect and the potential in design reuse for NPD has not been fully exploited; (2) such issues as indexing and retrieving mechanism as well as innovation capability in CBR-based design evolution system have not been well addressed; and (3) an integrated and complete process to ease the knowledge acquisition bottleneck in constructing a knowledge-based system is still lacking.

This research aims at addressing the aforementioned observations in product conceptualization and developing a novel methodology that helps designers to achieve better design solutions in an efficient manner. The design evolution and knowledge handling in product conceptualization form the kernel of the methodologies described in this work. On the basis of the problems, objectives and research scope described in previous chapters, corresponding approaches and methodologies are proposed to bridge the gaps in product conceptualization identified in Chapter 2. The major approaches and considerations for achieving the objectives of this thesis are described as follows.

3.1 Overview of the Prototype System

In this work, a systematic conceptual design evolution system based on a case-based approach, so-called case-based design evolution system (CBDES), is proposed. Figure 3.1 shows a framework of CBDES. The CBDES focuses on the conceptual design stage. It comprises three major approaches, namely a feature-based design evolution approach, an innovative adaptation procedure (IAP) and a matrix representation and mapping (MRM) approach. These three approaches are depicted as shaded ellipses in Figure 3.1.

As shown in Figure 3.1, given a set of new design specifications, the case-based design evolution process will be performed to identify the most similar design solutions from a case database for further investigation. Subsequently, the potential design solutions or concepts derived by the case-based design evolution process will be assessed in terms of their appropriateness according to the design attributes such as manufacturability, assemblability and cost. Specifically, to facilitate the design evolution process for concept generation, a feature-based design evolution approach (Chapter 4) is proposed. To arm the proposed design evolution approach with the capability of innovation, a TRIZ-enhanced innovative adaptation procedure (IAP) (Chapter 5) is proposed. In addition, a knowledge-based approach is suggested to support the multi-attribute design evaluation. Particularly, a matrix representation and mapping (MRM) approach (Chapter 6) is proposed to tackle the difficulty of knowledge acquisition in constructing a knowledge-based system.

3.2 Case-Based Design Evolution

As bespoke, design reuse has been recognized as an effective strategy that provides a company with competitive edge to out-perform its competitors in creating new products. Reusing past design solutions is highly necessary for designers to improve the efficiency of product design. Case-based reasoning (CBR), which employs past problem-solving experiences in solving new problems, is a promising methodology for dealing with many complex engineering design problems. Owing to the preeminent advantages of CBR, which were discussed in Chapter 2, the proposed design evolution process employs a case-based approach to facilitate the effectiveness of product concept

development. The purpose of the case-based design evolution process is to effectively utilize the existing and previous related design information in designing a new product.

Figure 3.1 The framework of CBDES in a design process

3.2.1 A Feature-based Design Evolution Approach

To achieve the efficacy of reusing historical design information, in this work, the CBR techniques are adapted to facilitate the process of design information reuse. Hence, a feature-based design evolution approach via case-based reasoning is proposed. The approach is the kernel of the so-called case-based design evolution system. It attempts to address the issues of case indexing and retrieval mechanism in CBR systems for product concept development. In developing the feature-based design evolution approach, the following work will be involved:

- To handle the interrelationships among the various design features, which may cause additional difficulties and inefficiency in case retrieval;
- To derive the relative priority of the individual design feature effectively; and
- To select the most appropriate design attributes corresponding to each design feature.

Accordingly, three major functions, viz. feature relationship decomposition, feature priority determination and feature-based case retrieval, are integrated into the proposed approach. The details of these functions will be described in the next chapter, Chapter 4.

3.2.2 An Innovative Adaptation Procedure (IAP)

It is well known that the CBR methodology is based on the existing design information in solving new problems. In new product development, the major task of CBR is to adapt existing design solutions to satisfy the requirements of new design problems. In this work, the proposed design evolution approach adopts a design reuse process in designing a new product. In recent years, it has become an imperative for companies to deliver innovative products to meet the ever-changing requirements of

customers. However, few of the conventional CBR methodologies are capable of generating innovative design solutions in product development. Based on this understanding, the proposed design evolution approach is enhanced with TRIZ solutions in this work. Accordingly, an innovative adaptation procedure (IAP) is proposed as a complementary module of the design evolution approach. The IAP procedure aims to provide innovative solutions for new product concept development, based on the partially available design information. Specifically, the TRIZ techniques will be incorporated into the adaptation stage of the proposed CBR methodology so as to suggest innovative solutions in new product development. The functions of the IAP procedure are elaborated in Chapter 5.

3.3 A Novel Knowledge Acquisition Approach for Design Evaluation

Nowadays, product design cannot be viewed as a stand-alone process and must be considered in the context of integration with other product development activities, such as manufacturing, costing, quality control, etc. (Chen et al. 2001). Otherwise, the products designed maybe difficult to manufacture or assemble, require high material or equipment cost, or contain some design flaws that may cause a lot of extra work for rectification. Hence, in order to improve the quality of product design, it is imperative for designers to take such attributes as manufacturability, assemblability, cost, etc. into consideration during the evaluation process. In a case-based reasoning process, design evaluation can help assess the appropriateness of a proposed solution, as well as point out the need for additional adaptation or repair of the solution. The purpose of evaluating manufacturability and assemblability is to assure that the product designed can be manufactured and assembled easily and effectively. The cost comparison procedure is

applied to consider the manufacturing costs for different design alternatives on a comparison basis. In this respect, Chen et al. (2001) developed a concurrent design evaluation system (CONDENSE) to help product designers in evaluating possible design solutions and design alternatives during the early design stage. The methodology encompasses a two-stage design evaluation. A qualitative aspect evaluation is applied during the stage of searching for combinations of solution principles to help determine the design specifications, and then a quantitative aspect evaluation is employed to provide information on performance, assemblability, manufacturability and costs to facilitate design selection. A prototype CONDENSE, which is a knowledge-based system (KBS), was constructed using a blackboard architecture (Craig 1991) that requires the classification of knowledge into appropriate knowledge sources. The results of the prototype CONDENSE were promising and revealed that the KBS approach is effective for multi-attribute design evaluation.

However, the experience in building the prototype CONDENSE confirms the well-known fact that knowledge acquisition is the most time-consuming phase as well as the bottleneck in developing a KBS. Knowledge acquisition is especially important in the multi-attribute design evaluation process due to the complex design information and various knowledge sources involved. To facilitate the knowledge acquisition process in constructing a KBS for supporting multi-attribute design evaluation, a novel knowledge acquisition approach using matrix representation and mapping (MRM) techniques is proposed in this work. In this respect, the following work will be carried out:

- To define the rule generation process in a mathematical way; and
- To express the rule extraction process using matrix representation and mapping techniques.

For this purpose, six consecutive steps for rule extraction are proposed in the approach. The procedure of the proposed MRM approach is elaborated in Chapter 6.

3.4 Summary

This chapter briefly discussed the main considerations and approaches to achieve the objectives of this work. The framework of the proposed system was described. The proposed prototype CBDES comprises three major approaches, viz. a feature-based design evolution approach, an innovative adaptation procedure (IAP) and a matrix representation and mapping (MRM) approach. The feature-based CBR approach will be described in more detail in Chapter 4. To enhance it with the capability of innovation, an innovative adaptation procedure (IAP), which incorporates TRIZ techniques into the adaptation stage of CBR methodology, is suggested. The IAP procedure is presented in Chapter 5.

To evaluate the appropriateness of the proposed solutions derived from the case-based evolution model with respect to multiple attributes such as assemblability, manufacturability and costs, a knowledge-based approach has been proven effective in CONDENSE (Chen et al. 2001). However, it is highly desirable to overcome the tediousness and improve the efficiency of the knowledge acquisition process proposed in CONDENSE. Therefore, a novel knowledge acquisition approach is proposed and described in Chapter 6.

CHAPTER 4 A FEATURE-BASED DESIGN EVOLUTION APPROACH VIA CASE-BASED REASONING

4.1 Introduction

As aforementioned, a consequence of rapidly changing business environment requires organizations to deliver new products with better performance, lower cost, higher quality and shorter time-to-market. To meet these demands, new products are often created through modifying and/or combining existing design solutions. Thus, design information reuse is of critical importance in new product development (NPD). In practice, evolutionary product design, instead of designing a product from scratch, is frequently adopted in NPD to reduce the development time and cost (Tay and Gu 2003). It has been established that reuse of existing design information is much more cost- and time- effective than regeneration of totally new information.

A number of approaches have been postulated to effectively reuse design information in NPD. Among these efforts, Otto and Wood (1998) suggested a ten step reverse engineering and redesign methodology with three primary phases for design information reuse. Ball et al. (2001) presented a design-reuse system that maximizes the benefits of rationale capture and information retrieval whilst minimizing the costs to the designer that might arise from disruption to natural design work. Tay and Gu (2003) proposed a function-based approach for supporting conceptual design in the context of evolutionary product development. The approach aims to improve a designer's productivity via effective reuse of existing design information in design alternative identification, evaluation, and modification. Ong and Guo (2004) developed an integrated framework to computerize the design reuse process and support design reuse from the original

requirements through the concept and embodiment stage to achieve a final detailed design.

As already mentioned in Chapter 2, although much research has been conducted on design information reuse, the issues on indexing, retrieval, and adaptation of prior design knowledge have not been well addressed. As such, in this chapter, a feature-based CBR (case-based reasoning) approach to facilitate conceptual design evolution in NPD is proposed. In Section 4.2, some relevant CBR methodologies are described. Section 4.3 provides the details of the proposed CBR approach. A case study on wood golf club design is presented in Section 4.4 to illustrate the capability of the proposed approach. The last section, Section 4.5, summarizes the conclusions reached in this chapter.

4.2 CBR in New Product Development

Among the various design reuse methods, case-based reasoning (CBR) has been recognized as a promising methodology for dealing with many complex engineering design problems. In product design, the use of cases allows the designers to concentrate on merging and adapting old solutions and suggest effective ways to provide appropriate solutions for new design problems. As suggested by Maher and Pu (1997), CBR has more advantages on formalization of the generalized design experience than other artificial intelligence methods, such as rule- and model-based reasoning techniques. Other advantages of the CBR approach can be found in Riesbeck and Schank (1989), as well as Bing and Liang (1995).

In general, a CBR process comprises four phases, viz. representation of case base, case retrieval and reuse, case revision and evaluation, and retainment of the final solutions to update the case base, as shown in Figure 4.1. A successful CBR system is

the one which is capable of effectively and efficiently retrieving similar cases for dealing with a new design problem. In a CBR process, case indexing and retrieval is of crucial importance. This is especially true for a new design problem that involves a huge case database. In this case, without an effective guiding methodology, searching and retrieving similar case(s) would be very time- and effort-consuming. Moreover, the interrelationships among various design features and cases can be very complicated, which may worsen the situation. These interrelationships need to be well handled, as they may lead to additional difficulties and inefficiency in case retrieval.

Figure 4.1 A typical CBR process

As reviewed in Chapter 2, CBR has been applied in various aspects of product development, including mass customization (Tseng and Jiao 1997, Tseng et al. 2005), drying equipment design (Kraslawski and Kudra 2001), production machines design (Vong et al. 2002), decision support (Belecheanu et al. 2003), printed circuit board design (Tsai et al. 2005) and fixture design (Boyle et al. 2006). In addition, Rao and

Chen (2005b) proposed a CBR-based design evolution approach attempting to improve the design efficiency via using historical design experiences. In the approach, a similarity measurement method based on fuzzy sets theory is suggested for case retrieval under uncertainty in the preliminary design stage. Although CBR has been applied successfully in various domains, to date, the indexing and retrieval issues have not been fully addressed. For example, in product design domain, little of the previous research in CBR systems considers the qualitative relationships among various design features. Moreover, a systematic and effective approach to handle large-scale design problem in CBR systems is still lacking. To bridge these gaps, a feature-based design evolution approach via case-based reasoning (Rao and Chen 2005a, 2005b, 2006), which incorporates graph decomposition and clustering analysis, is presented in this chapter.

4.3 A Feature-based CBR Approach

The proposed approach aims at effectively utilizing the existing and historical design information in designing a new product. In so doing, CBR techniques are adopted to facilitate the process of design information reuse. As shown in Figure 4.2, there are three major components in the proposed approach, viz. feature relationship decomposition (FRD), feature priority determination, and feature-based case retrieval. Additionally, corresponding methods and algorithms are developed to realize these components. As can be seen, a graph decomposition algorithm is employed for feature relationship decomposition. For feature priority determination, the analytical hierarchy process (AHP) is adopted to rank the priorities of features derived from the FRD procedure. Clustering analysis is performed on the case repository to facilitate the subsequent

feature-based retrieval process. Details of the proposed approach are presented in the following sub-sections.

Figure 4.2 A framework of the feature-based CBR approach

4.3.1 Feature Relationship Decomposition (FRD)

As aforementioned, the case retrieval process is critical and could be difficult when there are many design cases associated with the new design problem. To deal with this problem, an effective indexing mechanism should be adopted to well organize the design cases in the case repository so as to facilitate the case retrieval process. In this respect, the cases are oftentimes indexed by various design features, which are extracted from the design cases or design specifications for a new design problem. However, in many

product design projects, the relevance of a case to a new problem does not always depend on the surface features, but may hinge on abstract relationships between them (Maher et al. 1995).

The indexing features for design cases are usually interconnected with abstract relationships and interdependent with each other. For example, the shape of a product may be interdependent with its geometric parameters, material, target users, etc. As a result, the interrelationships among the various design features make the indexing scheme appear to be a fixed structure, such as a hierarchical or networked structure. For a complex product design problem, it is not uncommon having a large number of features and hence a very complicated indexing structure. The interrelationships among the indexing features may decrease the effectiveness and accuracy of case retrieval. Furthermore, the hierarchical or networked indexing schemes are too structured and, therefore, unable to provide the sufficient flexibility needed in product design. Based on this understanding, a feature relationship decomposition (FRD) procedure is proposed to separate the complex interrelated indexing structure and consequently provide the indexing scheme with flexibility for case retrieval and retention.

With the proposed FRD procedure, a large and complex feature relationship structure can be decomposed into a set of blocks, which consist of a group of design features. The feature blocks obtained are easier to be managed because of their smaller size. There are a number of graph decomposition algorithms in the literature (e.g. Owen 1970, Bosak 1990, Guan and Liu 1993, Cohen 1996, McCreary et al 1998, Chen and Occeña 2000). Compared with other methods, the graph decomposition algorithm proposed by Chen and Occeña (2000) possesses the following advantages in decomposing the feature relationship structure: (1) the features that were considered to have closer relationships can be kept in the same block; and (2) the graph decomposition algorithm can

decompose the relationship structure into fewer feature blocks. As a result, fewer feature blocks need to be considered and less efforts are required in the subsequent process of feature priority determination, during which the features are prioritized based on the various feature blocks decomposed. As such, the algorithm suggested by Chen and Occeña (2000) presents a good choice for the FRD procedure. The algorithm is a modified version of a non-directed graph decomposition algorithm developed by Owen (1970). The revised Owen's algorithm has been applied to transform a very large knowledge base into a number of smaller knowledge bases in constructing a product design blackboard system (Chen et al. 2001, Chen et al. 2002b). Being applied in execution of the FRD procedure, the graph decomposition algorithm comprises thirteen consecutive steps as follows (Chen and Occeña 2000):

Step 1. Set the value for Connection Ratio (C.R.)¹ and each kind of relationship.

Subsequently, according to the relationship between two design features, assign a weight to all the links (a link is a relationship between design features).

Step 2. Assign a weight to all the vertices (a vertex represents a feature in the feature relationship graph).

Step 3. Sort vertices in the descending order in terms of the weight assigned.

Step 4. Apply a strong-link decomposition pass.

Step 5. Reconsider all the vertices in order and apply follow-up decomposition passes.

Step 6. Compare the subgraphs found in the present series of decomposition passes with those saved from previous passes.

Step 7. Compute the value of the cutting function, k_i ; for a possible i th cutting pass.

¹ The connection ratio (C.R.) is the ratio of actual incident internal links to possible incident links for the vertex under consideration in a graph.

- Step 8. Select the largest previously unselected subgraph (or subgraphs if there is more than one of the same size) from the list compiled in Step 6. Ignore all subgraphs that intersect previously selected subgraphs
- Step 9. Remove the selected subgraph (or subgraphs) from the graph and make it a block of the final partition.
- Step 10. Examine the remaining vertices in the graph. If only disconnected singleton or doubleton subgraphs remain, go to Step 11. If larger structures are still present, return to Step 1.
- Step 11. Use the largest subgraph (or subgraphs if more than one of the same size exists) found in partition as the first block of the non-disjunctive decomposition.
- Step 12. From the list of Step 6, select and remove the largest remaining subgraph (or subgraphs) covering previously uncovered vertices. Ignore all subgraphs that are proper subgraphs of those previously selected.
- Step 13. Examine the remaining graph structure. If all vertices are covered, the decomposition is complete. Otherwise return to Step 12.

Appendix A provides more information regarding the graph decomposition algorithm. Figure 4.3 shows the fundamental concept of using a decomposition process to decompose a feature relationship graph. Once decomposed, the features are grouped into different blocks. For instance, as shown in Figure 4.3 and Table 4.1, features F_1 , F_2 and F_m are clustered in Block-1, whereas F_8 , F_9 and F_n are grouped in Block-3.

Figure 4.3 Graph decomposition for feature relationship

Table 4.1 Features grouped in blocks

Block-1			Block-2					Block-3			...	Block-(n)	
F ₁	F ₂	F _m	F ₃	F ₄	F ₅	F ₆	F ₇	F ₈	F ₉	F _n	...	F _(m+1)	...

4.3.2 Feature Priority Determination

A case retrieval process is used to identify similar design cases or sub-cases from a set of cases retrieved from the design case repository according to the established indexing scheme. The most similar one is then selected. During the retrieval process, the typical features representing the major functions or specifications of the design problem are given a prior consideration. For example, in wood golf club design, the feature “material” should be determined prior to the price, since material has a significant impact on the price. However, it is tedious and even error-prone in prioritizing each feature of a large and complex design system. As such, the number of features may be huge and the interdependency among different features can be extremely complicated.

Hence, it would be difficult for the decision makers to assign relative weights for all features simultaneously. To deal with this problem, the analytical hierarchy process (AHP) (Saaty 2000) presents a logical alternative.

AHP is a powerful tool that supports decision making via setting priorities. It is especially suitable for complex decisions that involve a series of pair-wise comparisons between decision elements, when both qualitative and quantitative aspects of a decision need to be considered. However, while AHP is a suitable method for alternative ranking, it can still be extremely tedious. This is because as the number of the ranking alternatives increases, the dimensions of the problem naturally expand to yield an evaluation matrix with many columns and rows (Ayag 2005). This leads to lengthy calculation and, hence, increases computation cost. To deal with this problem, the proposed FRD procedure can be very helpful. With the FRD procedure, a feature relationship structure is decomposed into a set of partitioned blocks, each of which contains a number of feature alternatives. Therefore, even if a large number of feature alternatives are involved, the pair-wise comparisons between the alternatives can be carried out in each of the smaller separated blocks.

In determining feature priorities, which are ranked in terms of the weights associated with each feature, a square matrix is constructed to compare the feature alternatives pair-wisely. The weights associated with each feature are then calculated geometrically. In pair-wise comparison, AHP does not directly use fuzzy number or membership functions to express the uncertainty and fuzzy information about the relative importance of different elements. Instead, a 9-point pair-wise comparison ratio scale (Table 4.2) is used to measure the fuzziness. The ratio scales represent the impact of elements with each other.

Table 4.2 Numerical scale of the relative importance (Saaty 2000)

Numerical Value	Importance definition	Explanation
1	Equal importance	Two activities contribute equally to the objective
3	Moderate importance	Experience and judgment slightly favor one activity over another
5	Strong importance	Experience and judgment strongly favor one activity over another
7	Very strong importance	An activity is strongly favored and its dominance is demonstrated in practice
9	Extreme importance	The evidence of favoring one activity over another is of the highest possible order of affirmation
Reciprocals	If activity i has one of the above numbers assigned to it when compared to activity j , then j has the reciprocal value when compared with i	

Notes: Numerical values 2, 4, 6 and 8 stand for intermediates.

According to the AHP techniques summarized by Chougule and Ravi (2005), the weight summations of both the individual feature alternatives in a particular block and all feature blocks can be normalized to one. The overall weight of a particular feature alternative is equal to the product of its own weight and the weight of the feature block it belongs to. Figure 4.4 shows an AHP hierarchy derived from the FRD procedure for feature priority determination based on Figure 4.3 and Table 4.1. The absolute weight of each feature can be determined using Equation (4-1).

$$W_{Fk} = W_{Bi} \times W_{Fij} \quad (4-1)$$

where W_{Fk} is the absolute weight for each feature alternative, $k = 1, 2, \dots, n$; W_{Bi} is the relative weight for each feature block, $i = 1, 2, \dots, p$; W_{Fij} is the relative weights for each

feature alternative in a particular block, $i = 1, 2, \dots, p$, $j = 1, 2, \dots, q$. In terms of AHP, W_{Bi} and W_{Fij} can be determined as follows:

$$A \hat{W} = \lambda_{\max} \hat{W} \quad (4-2)$$

where A is the pair-wise comparison matrix, $\lambda_{\max}$ is the maximum eigenvalue, and $\hat{W}$ is the eigenvector representing the relative weights of feature blocks or the feature alternatives in a block. In determining the relative weights between feature blocks, $\hat{W} = [W_{B1}, W_{B2}, \dots, W_{Bp}]^T$. Or else, $\hat{W} = [W_{Fi1}, W_{Fi2}, \dots, W_{Fiq}]^T$, representing the weight vector for feature alternatives.

In order to validate whether the pair-wise comparison matrix provides a consistent assessment, the consistency ratio (CR) is checked using the maximum eigenvalue ($\lambda_{\max}$) and calculated as per the following two steps:

1. Compute the consistency index (CI) for each matrix using Equation (4-3):

$$CI = (\lambda_{\max} - n) / (n - 1) \quad (4-3)$$

where n is the order of the matrix.

2. Subsequently, the consistency ratio (CR) is calculated using Equation (4-4):

$$CR = CI / RI \quad (4-4)$$

where RI is the random consistency index, which can be obtained from a large number of simulations in terms of the order of matrix. Tables 4.3 shows the value of RI obtained by approximating random indices using a sample size of 500 for matrices of order 1 to 10 (Saaty 2000, Atthirawong 2002). As suggested by Saaty (2000), a CR value of 0.10 or less is considered acceptable. Otherwise it would bias the result by considerable margin, and the pair-wise comparisons must be repeated.

Table 4.3 Random index for the comparison matrix
(Saaty 2000, Atthirawong 2002)

Order of Matrix (n)	1	2	3	4	5	6	7	8	9	10
Random index(RI)	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Figure 4.4 AHP hierarchy derived from FRD

4.3.3 Feature-based Case Retrieval

In selecting similar cases, most previous work focused on the search and retrieval of a complete case. However, it is possible that not all the attributes and values corresponding to each feature in the retrieved cases are desirable. These unsuitable attributes and values may cause extra time and cost during adaptation, which is a major challenging part of a CBR system. As such, for effective adaptation, it would be highly

desirable to select the most appropriate attributes corresponding to each feature in each case in selecting similar cases. In doing so, for a particular design problem, the most appropriate attributes and values corresponding to the indexing features in different design cases are selected and classified. Subsequently, the potential design solutions are generated by combining those selected attributes and values for further investigation. For this purpose, a feature-based case retrieval process incorporating the techniques of clustering analysis is developed and embedded as a module of the proposed CBR-based system. The retrieval process aims to select the most appropriate attributes corresponding to each design feature, which has been grouped into different blocks by the FRD procedure.

4.3.3.1 Clustering Analysis and Similarity Reasoning

As aforementioned, the process of case retrieval can be extremely time-consuming for a complex design problem that involves a huge case repository. To deal with this problem, clustering analysis is employed to quickly locate and discover similar cases. With clustering analysis, the design cases in the case repository are partitioned into different clusters according to the corresponding indexing features. The huge case search space is thus narrowed down into smaller case groups. Subsequently, the case retrieval process will focus on exploring these case clusters to search for similar cases. Based on the indexing mechanism and results of clustering analysis, the case retrieval efficiency can be improved for the large-scale CBR-based design problem.

Clustering analysis refers to those methods that identify and classify individual objects into different clusters on the basis of the similarity among them. After classification, objects within a cluster should be as similar as possible, while the

difference among clusters should be as much as possible. Clustering analysis methods are generally characterized by similarity measurements that are employed to perform clustering. Most of the existing algorithms for clustering analysis can be categorized into two general categories (Braha 2001): (1) partitioning clustering algorithms, such as *k*-means (Kanungo et al. 2002), (2) hierarchical clustering algorithms, such as BIRCH (Zhang et al. 1996) and CURE (Guha et al. 2001).

Apart from the two general categories, Kohonen's self-organizing map (SOM), has been proven to be one of the most popular and powerful unsupervised competitive neural network learning models for clustering and visualization (Kohonen 2001). Mangiameli et al. (1996) compared SOM neural network with hierarchical clustering methods and concluded that the former is superior to the latter. The effectiveness of clustering messy empirical data in decision-making and research problems can be improved with the SOM method, thanks to the superior accuracy and robustness of neural networks. The main function of SOM is to visualize the topologies of higher dimensional input spaces into the appropriate classes of similar objects by creating a two-dimensional feature map. It is used to map a multi-dimensional data set onto a typically two-dimensional surface. In product design, the cases may have multiple dimensional qualitative attributes. For instance, the following four attributes may be included in selecting a material for a wood golf club head: profitability, process-ability, toughness and durability. For more effective and accurate clustering analysis, in this work, the SOM method is adopted. The advantages of adopting the SOM for clustering analysis include (Chen et al. 2006):

- Similar to other neural network strategies, it can be easily adapted to handle such complex (often uncertain or inconsistent) and correlated (non-linear and not isolated) situations in decision-support compared to those linear functions such as *k*-means clustering model.

- In contrast to supervised neural networks, it possesses faster incremental learning ability without requiring large samples, as well as self-organized inheritance without any desired outputs.
- Compared with other self-organizing neural networks such as adaptive resonance theory (ART) network, it is more controllable and targeted in clustering.

In a CBR-based design evolution approach, it is critical to measure the similarity and identify the most appropriate cases for the new problem during the case retrieval process. In the proposed approach, the similarity reasoning method is based on the Euclidean distance (Kantardzic 2003), which is commonly used as a dissimilarity measure between a pair of objects. The most appropriate cases are defined as the cases that have the minimum dissimilarity distance with the new design specifications. In the following section, the overall case retrieval process including clustering analysis and similarity reasoning is elaborated.

4.3.3.2. Overall Case Retrieval Process

Figure 4.5 shows the overall case retrieval process on the basis of feature indexing. The process comprises the following steps.

1. For a particular design problem, the case repository can be represented with a series of data sets, which include the attribute cases corresponding to each feature. The case repository is defined as:

$$CR = \{C_i \mid i = 1, 2, \dots, n\}, \text{ and}$$

$$C_i = \{F_i, A_i\}, \text{ and}$$

$$A_i = \{a_{i1}, a_{i2}, \dots, a_{is}\}$$

Figure 4.5 Overall feature-based case retrieval process

where C_i denotes the i th past case in the case repository, F_i stands for the feature extracted from the design specifications, A_i is the attribute set corresponding to a particular feature, and a_{is} is the value of attribute i .

2. Based on the defined case repository, clustering analysis is performed. As shown in Figure 4.5, the attribute cases are classified using Kohonen's SOM method.
3. For each cluster C_{ik} , the centroid value, which is used to represent the cluster, is determined on the basis of square-error criterion. The cluster centroid is defined as a hypothesis attribute vector that has the minimum sum of the squared Euclidean distance between itself and all the other attribute cases in the cluster.

It can be determined by the following equation,

$$M_{ik} = v_{ik},$$

$$\text{subject to: Minimum } \{ \sqrt{\sum (\text{Dist}(v_{ik}, a_{ix}))^2} \} \quad (4-5)$$

where M_{ik} denotes the centroid value of C_{ik} , v_{ik} represents the hypothesis attribute vector and a_{ix} is an attribute case in cluster C_{ik} , $x = 1, 2, \dots, s$.

$\text{Dist}(v_{ik}, a_{ix})$ is the Euclidean distance between v_{ik} and a_{ix} .

4. Let $S = \{S_i | i = 1, 2, \dots, n\}$ denote the data set representing the design specifications for the new design problem. For each design specification S_i , perform the similarity measurement with respect to cluster C_{ik} by calculating the distance $\text{Dist}(S_i, M_{ik})$.
5. Given a threshold δ , if $\text{Dist}(S_i, M_{ik}) \leq \delta$, the attributes in C_{ik} can be considered as the candidate similar cases, otherwise, the attributes cluster C_{ik} can be ignored because all attribute cases in C_{ik} are not similar enough to S_i and thus not suitable for the new design problem. The clusters consisting of a set of candidate similar cases are regarded as similar clusters and selected for further investigation.

6. After the similar attribute cases in a similar cluster are identified, the most similar attribute case $a_{ik(r)}$ in the cluster can be determined by comparing the dissimilarity distances between each case, denoted as $a_{ik(t)}$, in the cluster with its centroid. The one with minimum dissimilarity distance from the centroid is selected as the most similar case.
7. Due to the possibility of a number of similar clusters being generated in Step 5, the most similar cases in each similar cluster will be inter-compared to determine the most appropriate case corresponding to each feature. The most appropriate case is identified as the one with the minimum dissimilarity distance with the specification attribute S_i .
8. After the most appropriate attribute for each feature is determined, the final solution for the new design problem can be composed of those retrieved most appropriate attribute cases.

The proposed case retrieval process described above is a computer-oriented technique, and thus a computer program is developed to implement it. To clarify the procedure, a pseudo-code which describes the core algorithm of the feature-based case retrieval process is illustrated in Figure 4.6.

```

Input: threshold  $\delta$ ; Attribute case set;
      case design specification  $S_i$ 
SOM clustering analysis
For each cluster  $C_{ik}$ 
    Compute the centroid value  $M_{ik}$ 
    For each design specification  $S_i$ 
        Calculate  $\text{Dist}(S_i, M_{ik})$ 
        If  $\text{Dist}(S_i, M_{ik}) \leq \delta$ 
            For each attribute case  $a_{ik(t)}$ 
                Calculate  $\text{Dist}(a_{ik(t)}, M_{ik})$ 
            End
             $a_{ik(r)} = \text{Minimum}(\text{Dist}(a_{ik(t)}, M_{ik}))$ 
        Else
            disp('no attributes cases can be retrieved');
        End
    End
End
For each similar case  $a_{ik(r)}$ 
    Calculate  $\text{Dist}(a_{ik(r)}, S_i)$ 
End
Output Minimum ( $\text{Dist}(a_{ik(r)}, S_i)$ )
% Notes: The code is in terms of the grammar of the Matlab programming language

```

Figure 4.6 Pseudo-code for the core algorithm of feature-based case retrieval process

However, it is not uncommon that the most appropriate solutions identified do not perfectly fit the specifications of the new design problem. As such, those unsuitable solutions selected need to be modified in order to fully satisfy the new design specifications. There are several adaptation strategies that can be used in a CBR system. These strategies include Simple Substitution, Parameter Adjustment, and Constraint Satisfaction (Kolodner 1993). More information about the adaptation can be found in Chapter 5. During case adaptation, the substitution of attributes and values from the new design specifications to the design case is often required, and a propagation of the effect of the new values on continuous variables with associated procedural attachments also occurs. Subsequently, based on the final solution derived from the case retrieval process, different conceptual design alternatives can be generated.

4.4 Case Study

Conceptual design evolution is an important stage in the development of successful wood golf clubs. In recent years, the golf business has enjoyed dynamic growth and is continuing to expand and change daily. Owing to the rapidly increasing golfing populations and the intensive competition from domestic and global manufacturers, in order to survive, golf club manufacturers have substantially increased expenditures in research to advance their knowledge in design and manufacturing technology. As such, a powerful and efficient tool for golf club design is highly desirable. The proposed feature-based CBR approach can be used to meet this demand.

4.4.1 An Example

A hypothetical wood golf club design case is used to demonstrate how the proposed approach works. The purpose of the case study is to design a new wood golf club for the young male amateur golfers in the US market. The proposed price level for this golf club is medium, and the performance requirement is ‘weight adjustable’. The new wood golf club is designed using the proposed CBR methodology based on previous golf club design information. The general specifications of a wood golf club are shown in Figure 4.7. More information about golf club design can be found in Maltby (1995) and Chen et al. (2001).

Figure 4.7 Specifications of a wood head golf club (Maltby 1982)

The case study follows the steps described in Section 4.3. In this case, twelve features are identified for the wood golf club design. These features include: material, weight, volume, sole radius, face bulge, face roll, face progression, lie, loft, neck length, flex and price. The relationships among different features are obtained by consulting the golf club designers. The feature relationships and decomposition results by the FRD procedure are shown in Figure 4.8, where C.I.R, V.I.R, I.R, R and N.R represent ‘critical important relationship’, ‘very important relationship’, ‘important relationship’,

‘relationship’ and ‘no relationship’, respectively. As an example, it can be observed that material has very important relationships with both the weight and flex of a golf club, whereas it has critical important relationship with the price of a golf club. The features without a link in-between indicate that there are no relationships (N.R) between them. Since N.R means no relationship exists, it does not appear in the figure. There are three major factors that may affect the result of decomposition, namely, magnitude and levels of the assigned weights, the links to which the weights are assigned and the value of the connection ratio (C.R.) (Chen and Occeña 2000). In this case, three levels of the assigned weights are used (see Appendix A): the value 1 representing “important relationships”, the value 5 representing “very important relationships”, and the value 10 representing “critical important relationships”. A weight of 1 will guarantee that the links with an assigned weight will remain after all non-weighted links are cut. A weight of 10 is large enough to separate the second level from the first level. The weight of 5 is regarded as the medium level. The graph decomposition algorithm with a C.R. value of 0.75 is applied to decompose the feature relationship graph. The sensitivity tests on the C.R. value can be referred to Chen and Occeña (2000).

Figure 4.9 shows the result of decomposing the graph shown in Figure 4.8 using an application program developed by Chen et al. (2002). As shown in Figures 4.8, each curved region that envelops the feature nodes represents a separated feature block. Figure 4.9 illustrates a main window of the application program implemented in the FRD procedure. In the left side of the window is a relationship table that describes the relationships between the feature alternatives in the graph (Figure 4.8). After decomposition, the resultant subgraphs are listed in a tree structure on the right side of the main window. Each subgraph represents a feature block corresponding to Figure 4.8. As can be seen, such features as material, flex, weight and price are grouped into

subgraph1 which is considered as block-1 (Figure 4.10), whereas sole radius, volume and loft are grouped into subgraph2 (block-2).

Figure 4.8 FRD procedure for golf club features

Figure 4.9 Graph decomposition with application program

Once the feature relation structure has been decomposed, the feature alternatives of the golf club are grouped into a number of blocks. The AHP hierarchy for feature

priority determination is shown in Figure 4.10. After that, the feature blocks as well as feature alternatives in each block are compared reciprocally between each other. According to the AHP ratio scales (Table 4.2) denoting the relative importance of the elements to be compared, the pair-wise comparison matrices between the feature blocks and feature alternatives in each block are established. Table 4.4 shows the comparison matrices for feature blocks and the four separated blocks. As an example in Block-1, feature ‘Material’ is given importance of 4 when compared with feature ‘Weight’, whereas it has moderate importance over feature ‘Flex’. Similarly, both ‘face roll’ and ‘face progression’ are assigned equal importance of 1 over each other shown in Block-3 (note that 1 means equal importance, and 9 means extreme importance compared to the other). According to these comparison matrices and Equations (4-2), (4-3) and (4-4), the maximum eigenvalue ($\lambda_{\max}$), consistency index (CI), consistency ratio (CR) and eigenvector ($\hat{W}$) for feature blocks and feature alternatives are calculated and illustrated in Table 4.5. As shown in the table, both the CR values for the comparison matrices between feature blocks and feature alternatives in blocks 1 and 2 are much less than 0.10. It implies very slight inconsistency in the pair-wise comparisons, but well within the acceptable limit of 0.10. Besides, since the maximum eigenvalues for blocks 3 and 4 are both equal to the orders of the comparison matrices, the pair-wise comparisons in these two blocks are absolutely consistent. Furthermore, based on the comparison matrices and Equation (4-2), the weight vector $\hat{W}$ can be derived (Table 4.5). The relative weights for feature blocks and feature alternatives in a particular block are then obtained after normalizing the vector $\hat{W}$. According to Equation (4-1), the final feature priorities based on the associated weight of each feature are calculated using the AHP process. The results are summarized in Table 4.6.

Table 4.4 Pair-wise comparison matrices between feature alternatives

Blocks	Block-1	Block-2	Block-3	Block-4	Block-5
Block-1	1	2	4	3	2
Block-2	1/2	1	3	2	1
Block-3	1/4	1/3	1	1/3	1/2
Block-4	1/3	1/2	3	1	1/2
Block-5	1/2	1	2	2	1

(a)

Block-1	Material	Weight	Flex	Price
Material	1	4	3	2
Weight	1/4	1	1/2	1/4
Flex	1/3	2	1	1/2
Price	1/2	4	2	1

(b)

Block-2	Sole Radius	Volume	Loft
Sole Radius	1	3	1
Volume	1/3	1	1/2
Loft	1	2	1

(c)

Block-3	Face Roll	Face Progression
Face Roll	1	1
Face Progression	1	1

Block-4	Lie	Neck Length
Lie	1	2
Neck Length	1/2	1

(d)

Figure 4.10 AHP hierarchy for feature priority determination

Table 4.5 Results of consistency check for comparison matrices

	Blocks	Block-1	Block-2	Block-3	Block-4
Maximum Eigenvalue ($\lambda_{\max}$)	5.116	4.046	3.018	2	2
Consistency Index (CI)	0.029	0.015	0.009	0	0
Consistency Ratio (CR)	0.026	0.017	0.016	-	-
Eigenvector ($\hat{W}$)	[0.748, 0.430, 0.151, 0.271, 0.400] ^T	[0.802, 0.152, 0.271, 0.510] ^T	[0.724, 0.276, 0.632] ^T	[1, 1] ^T	[2, 1] ^T

Table 4.6 Feature priorities based on weights

Block No	Relative Weight	Feature	Relative Weight	Final Feature Weight
1	0.374	Material	0.462	0.173
		Weight	0.088	0.033
		Flex	0.156	0.058
		Price	0.294	0.110
2	0.215	Sole Radius	0.444	0.095
		Volume	0.169	0.036
		Loft	0.387	0.084
3	0.076	Face Bulge	1.000	0.076
4	0.136	Face Roll	0.500	0.068
		Face Progression	0.500	0.068
5	0.199	Lie	0.667	0.133
		Neck Length	0.333	0.066

In this case, forty previous case instances for each feature, except the feature “flex” which has only five generally used ratings, viz. Extra Stiff, Stiff, Regular (ReFlex), Senior and Ladies, are chosen and stored in the case repository. Subsequently, the most appropriate attribute cases corresponding to each feature are selected to generate the final potential solution to satisfy the requirements of the new design problem. The case retrieval process is in accordance with the derived feature priorities shown in Table 4.6, i.e. the cases corresponding to features which are associated with higher weights are retrieved first. For this purpose, a MATLAB program is developed to implement the similarity reasoning and feature-based case retrieval process. Figure 4.11 illustrates the graphical user interface (GUI) of the computer-aided case retrieval process. The twelve features of golf club are listed on the right side of the window. The circles in the left window represent the various cases corresponding to each feature, which are expressed with multi-dimensional vectors. Choose each feature in terms of the priority sequence derived by the AHP process (Table 4.6), i.e. material, lie, price, sole radius, loft, face

bulge, face roll, face progression, neck length, flex, volume and weight. Then press the “start” button to implement the SOM neural network training and clustering process. Figure 4.12 and Table 4.7 show a sample of SOM training result for the cases corresponding to the feature “material”.

Figure 4.11 Screenshot of the GUI for feature-based case retrieval

Figure 4.12 A sample of SOM training result (for 3×2 output grid), $t_{\max} = 200$

Table 4.7 Sample trained weight vectors from the SOM

Output node	Trained Weight Vector			
1	5.5439	9.4246	6.4830	3.1810
2	5.5136	9.2756	6.6881	3.4205
3	5.6929	9.0054	6.9808	3.5416
4	5.3187	9.3556	6.6534	3.3093
5	5.4625	9.1230	6.9406	3.4623
6	5.6851	8.9290	7.1439	3.4165

Subsequently, the final results of clustering analysis using the SOM function and case retrieval for feature material are presented in Figure 4.13. As shown in the figure, the solid circles with same colors are classified into a cluster. Two symbols, “+” and “*” , are used to represent the centroid of each cluster and the retrieved most appropriate case for each feature, respectively. The symbol “x” is used to stand for a design specification attribute. The specific value of the appropriate case is shown in the middle window on the right side, i.e. the solution “Composite Material” is selected for the feature “material”. By the same token, the solutions for the rest features can be retrieved using the application program. The screenshots showing the results of clustering analysis and case retrieval for the rest features can be found in Appendix B.

Figure 4.13 Results of clustering analysis and case retrieval for material

Figure 4.14 Results of clustering analysis and case retrieval for loft

The retrieved cases, which do not satisfy the design specifications need to be modified and some parameters might be adjusted or substituted in adaptation by the

designer. As aforementioned, three means for case adaptation can be adopted (Kolodner 1993): (1) Perform simple parameters substitution, i.e. substitute parameters of the old problem for new user input; (2) Perform an old solution adjustment to make it fit the substituted user input according to the designer's knowledge; and (3) Check design constraints to ensure no inconsistency between the solutions and design specifications. For example, the retrieved most appropriate solution for the feature "loft" is "12°" (Figure 4.14). However, this value is considered slightly strong as the golf club is intended for young male amateur golfers. Hence, it needs to be adapted to a relatively weak loft. Based on the domain knowledge, "13°" of loft is a better choice. In this case, only the solution for loft is adapted. After the adaptation procedure, the final attribute values for each feature are acquired and listed in Table 4.8. Based on the results, different design alternatives can be generated.

Table 4.8 Final attribute values for each feature

Material	Face Bulge	Head Volume	Angle		Face Roll	Face Progression	Neck Length	Sole Radius	Flex	Weight	Price
			Loft	Lie							
Composite Material	10"	260cm ³	13°	57°	10"	1/16"	4"	6"	ReFlex	266g	\$320

4.4.2 Results and Discussion

The proposed FRD procedure can be very helpful in reducing the number of pairwise comparisons, which may consume lots of time and cost for computations in the AHP process. The effectiveness of the FRD procedure in supporting AHP can be shown as follows.

Assume there are M feature alternatives, which are divided into X blocks. The numbers of alternatives in each block are represented as $Y_1, Y_2, Y_3, \dots, Y_X$, respectively.

Thus, $Y_1 + Y_2 + Y_3 + \dots + Y_X = M$. Therefore, with the FRD procedure, the total number of pair-wise comparisons is

$$N_w = C_{X,2} + C_{Y_1,2} + C_{Y_2,2} + \dots + C_{Y_X,2} \quad (4-6)$$

where $C_{X,2}$ is the number of pair-wise comparisons between the separated blocks; $C_{Y_1,2}$, $C_{Y_2,2}$, $\dots$, $C_{Y_X,2}$ denote the number of pair-wise comparisons between the alternatives in the X blocks, respectively.

On the other hand, using the conventional AHP without FRD procedure, the total number of pair-wise comparisons is

$$N_{wo} = C_{Z,2} + Z \times C_{M,2} \quad (4-7)$$

where Z is the possible number of groups for the feature alternatives in the AHP hierarchy; $C_{Z,2}$ is the number of pair-wise comparisons between these groups; $C_{M,2}$ denotes the number of pair-wise comparisons between all the alternatives corresponding to each group.

Using the FRD procedure, the feature alternatives are grouped into 5 independent blocks. The numbers of features in each block are 4, 3, 2, 2 and 1, respectively. According to Equation (4-6), it can be found that 21 pair-wise comparisons are required in the AHP process. The number of pair-wise comparisons required by using AHP without the FRD procedure can be calculated using Equations (4-7). Table 4.9 shows the total number of pair-wise comparisons needed by AHP with or without FRD and the reduced numbers with respect to different Z values. It can be observed that the number of pair-wise comparisons has been significantly reduced with the FRD procedure.

Table 4.9 Number of pair-wise comparisons by AHP with or without FRD

$M=12; \quad X=5$			
Z	AHP (N_{wo})	AHP with FRD (N_w)	Δ (Reduced No.)
2	133	21	112
3	201	21	180
4	270	21	249
5	340	21	319
6	411	21	390
7	483	21	462
8	556	21	535
9	630	21	609
10	705	21	684
11	781	21	760
12	858	21	837

In most of conventional CBR systems, the similar case retrieval process focuses on the search and selection of the entire similar case. The most common and popular approach used in CBR is the nearest-neighbor based retrieval methods (Watson 1997), which are employed to generate similarity between the target problem and existing solutions in the case repository. In so doing, the similarity of all the attributes is summed up to provide a measurement between existing cases and new cases. Particularly, the nearest-neighbor matching (NNM) algorithm (Dasarathy 1991, Kolodner 1993) has been widely applied to evaluate the similarity between two cases. The basic conception of the NNM is to compare the attribute value of each feature of each case in the similar case set with every corresponding feature's attribute of the input case and calculates the comparison values, which are then summed up as a total comparison value. An application of the NNM in CBR for similarity measurement in product concept development can be found in Rao and Chen (2005b).

Despite of the popularity of NNM methods, the similarity measurement could result in huge computation time and effort, when a large number of cases are stored in the case repository. Assume there are M features extracted from the new design problem, and the number of attributes corresponding to each feature are $N_1, N_2, N_3, \dots, N_m$. Therefore, with the NNM, the number of possible cases, N , can be calculated as $N = N_1 \times N_2 \times N_3 \times \dots \times N_m$. It is obvious that as the attribute's number increases the number of possible cases can be exponentially increased. For instance, twelve features are selected in this case study, i.e. $M = 12$. The numbers of case instances for each feature are listed in Table 4.10. If the NNM method is used, the number of possible combinations would be $N = 5 \times 40^{11}$. This huge number of combinations would cause extremely heavy computation time and cost during the similarity measurement.

Table 4.10 Number of case instances for each feature in case study

Material	Face Bulge	Head Volume	Angle		Face Roll	Face Progression	Neck Length	Sole Radius	Flex	Weight	Price
			Loft	Lie							
N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}	N_{12}
40	40	40	40	40	40	40	40	40	5	40	40

In contrast with NNM, the feature-based case retrieval process proposed in this study concentrates on the individual attribute cases corresponding to each separated feature. Through clustering analysis, the huge case search space can be narrowed down into smaller case clusters. Those clusters that obviously do not contain potential cases will be filtered out. Hence, the case candidates can be more easily located by focusing on exploring those potential case clusters corresponding to each design feature. Moreover, with the approach, the potential design solutions for a specific design problem are generated by combining those selected appropriate cases. This can be very helpful to

reduce the extra adaptation efforts. From the case study, the proposed approach appears to be an effective tool for case retrieval in CBR systems.

4.5 Summary

In this chapter, a feature-based CBR approach is proposed to facilitate the conceptual design evolution process in NPD. The three major components, namely feature relationship decomposition (FRD), feature priority determination, and feature-based case retrieval of the approach have been investigated and implemented. The work is summarized as follows:

(1) A feature relationship decomposition (FRD) procedure using graph decomposition method is proposed to separate the complex interrelated feature indexing structure in a complex design problem. By decomposing the interrelationships between various design features, FRD provides the indexing scheme with more flexibility for case retrieval and retention.

(2) Analytic hierarchy process (AHP) is adopted for feature priority determination. In order to tackle the problem of lengthy calculation for pair-wise comparison which may be caused as the number of ranking alternatives increases, the function of AHP is enhanced through implementing the FRD procedure. As a result, with the proposed FRD procedure, the number of pair-wise comparisons in AHP has been significantly reduced.

(3) For effective case retrieval and adaptation, a feature-based case retrieval process is established and served as a module within the whole feature-based design evolution approach. The proposed process is aimed to select the most appropriate attribute cases corresponding to each separated design feature. It provides a new avenue for case retrieval focusing on selection of individual case attributes for design features. In addition, the retrieval process enables the potential design solutions for a specific design

problem to be generated by combining those selected appropriate cases. Thus, in turn, the extra adaptation efforts, which are a challenging problem in CBR, can be saved.

(4) To demonstrate the capability of the proposed approach, a case study on wood golf club design is exemplified. The result of the case study is analyzed and it reveals the effectiveness of the proposed approach by comparing with conventional CBR methodologies.

As already mentioned in the literature review, product innovation is vital for organizations to gain competitive advantages in delivering a new product to the market. While the proposed design evolution approach shows the effectiveness in reusing the design information for product conceptualization, it is still restricted with difficulties in design innovation which is highly desired in new product development. In response to this, the next chapter, Chapter 5, will carry out an investigation into the possibility of enhancing the CBR-based design system with the capability of product innovation.

CHAPTER 5 A TRIZ-ENHANCED INNOVATIVE ADAPTATION PROCEDURE (IAP)

5.1 Introduction

Product innovation is an essential strategic approach for creating competitive advantages in today's dynamic, globalized business environment. Innovation and creativity are highly valued in new product development (NPD) by companies. Product innovation involves providing the creative responses and solutions to meet the needs and expectations of customers and market(s) (Rainey 2005). It is regarded as the most cost effective and efficient means of achieving and sustaining the prosperity of most business organizations.

As mentioned in previous chapters, reuse of design information is highly important in NPD to reduce development time and cost. Case-based reasoning (CBR) is a promising methodology for effectively utilizing the historical design experiences in designing a new product. However, since CBR techniques are based on the existing case repository, it would be plagued with difficulties when innovation is required in developing new product concepts. For example, it would be difficult to design a new product with cutting-edge technology based only on the historical design information. New technical innovations through the development of new technologies and new knowledge derived from the accumulation of existing information should be introduced into the design reuse process for new product development. Therefore, an approach to enhance the CBR systems to reason with limited available knowledge is highly desirable. Such an approach would attempt to enable the generation of innovative product concepts on the basis of partially available design information. Additionally, as the feature-based design

evolution approach presented in Chapter 4 provides a rational way to support the design reuse process for design evolution, an attempt is made in this chapter to incorporate TRIZ (a Russian acronym for Theory of Solving Inventive Problems, Altshuller 1990; 1996) solutions to enhance the evolution approach with innovation capability.

5.2 Overview of TRIZ Methodology

Developed by Genrich Altshuller and his colleagues in the former USSR (Union of Soviet Socialist Republics) since 1946, TRIZ has become a powerful methodology that can be used to support innovative idea creation in the conceptual design process. It classifies innovative problems and offers corresponding principles by employing a knowledge base built from the analysis of over two million patents, primarily on mechanical design. TRIZ consists of the following major tools: Altshuller's matrix and 40 principles, substance-field analysis, ideal final result and laws of engineering system evolution. The contradiction matrix and 40 principles form the core tool of TRIZ methodology that is used to resolve technical contradictions. Those contradictions are represented using a contradiction table which is composed of 39 engineering parameters describing the contradiction uniformly, and 40 inventive principles offering possible solutions.

A number of researchers have applied TRIZ methodologies in various aspects of new product development. Most of the work can be found in the TRIZ journal. Yamashina et al. (2002) proposed an innovative product development process by integrating QFD (Quality Function Deployment) with TRIZ to enable the creation of technical innovation for new product development. Stratton and Mann (2003) brought together two parallel, but independent theories on inventive problem solving: TRIZ and the Theory of

Constraints (TOC), which was originated in manufacturing management. A term “systematic innovation” was used to describe the use of common underlying principles within these two approaches. The paper focused on the significance of trade-off contradictions to innovation in these two fields and explored their relationship with manufacturing strategy development. Chang and Chen (2004) presented a conflict-problem-solving CAD software, so-called Eco-Design Tool, which integrates TRIZ into the ecology design innovation. Cascint and Rissone (2004) proposed an approach in plastics design by integrating TRIZ creativity and semantic knowledge portals. The approach illustrates how the integration of modern design tools with creativity management techniques constitutes an effective procedure to enhance the capability of developing innovative products. Bariani et al. (2004) investigated into an approach that combines the design for manufacture and assembly (DFMA) method with TRIZ. The approach was developed by merging the common characteristics and connecting the complementary aspects of the two methods, which were then applied to the redesign of a satellite antenna. Sarno et al. (2005) reported a hybrid methodology for conceptual design of large systems with the goal of enhancing system reliability. The methodology considers the temporal quality characteristic "reliability" as the main objective and determines the optimal system design. Key ideas from several design methodologies, namely axiomatic design, robust design and TRIZ, have been integrated into the methodology.

Besides, Bohuslav et al. (1999) conducted a case study on heat exchanger design to demonstrate how TRIZ was used to break out of existing heat exchanger conceptual paradigms to produce a novel design solution. Eckhard and Ahrend (2004) described the experience of applying TRIZ principles of technological evolution to customer requirement based vehicle concepts. Gao et al. (2005) used the welding fixture design as

an engineering example to show the process of applying TRIZ into generating creative conceptual design ideas. By applying the TRIZ contradiction table to select proper innovative principles, Yan and Lee (2006) designed a straightline intermittent reciprocator with a polytropic motion curve that possesses the function of a variable velocity power source, under the condition of a fixed velocity power source.

5.3 Overall Framework of the Proposed Approach

On rare occasions, a retrieved case is exactly the same or absolutely suitable for the specifications of the new design problem. That means a retrieved case can only provide approximate solutions most of the time. Therefore, retrieved corresponding solutions need to be modified and adapted in order to fully satisfy the new design specifications. Case adaptation is the process of fixing up an old solution to meet the demands of the new design requirements. The adaptation of old cases is one of the major responsibilities of any CBR system (Al-Shihabi and Zeid 1998). The crucial issues in adaptation are identifying inconsistent parts that need to be adapted and then choosing a strategy to adapt them. Garza and Maher (1999) presented a case adaptation method that employs ideas from the field of genetic algorithms. Two types of adaptation, case combination and case mutation, are used to evolve variations on the contents of retrieved cases to reach a satisfactory solution. Rosenman (2000) developed a case-based design model, which uses an evolutionary approach for the adaptation of previously stored design solutions. Some other methods for case adaptation have been reported in Kolodner (1993), Leake (1996) and Maher and Pu (1997). However, few of previous research efforts have considered the innovation aspect in adaptation, which is important to eliminate the inconsistencies in generating creative design solutions. For this purpose, in

view of the aforementioned difficulties in CBR, an innovative adaptation procedure (IAP) incorporating TRIZ techniques is proposed as a complementary module of the proposed feature-based design evolution approach, which was described in Chapter 4. Figure 5.1 shows a framework of the IAP procedure.

Figure 5.1 Overview of the TRIZ-enhanced innovative adaptation procedure

The IAP procedure consists of three major parts, viz. Conflict Checking, Ordinary Reparation, and Resolving Design Contradictions with TRIZ. In turn, each part consists of a series of activities and steps.

(1) Conflict Checking

The adaptation function performs as a constraint satisfaction procedure (Maher et al. 1995), which eliminates constraints violations in a potential solution by searching for acceptable substitutions for the design attributes. At the beginning of the adaptation process, a preliminary or potential design solution is generated through retrieving the most similar case obtained from the case retrieval mechanism. Thereafter, the potential design solution is checked against the design constraints. The main purpose of using design constraints is to identify the invalid attributes and values within the potential solution. By identifying the differences in specifications between the most similar cases and new design problem, the inconsistent and unsuitable attributes to be modified can be determined. When the design constraints are violated, the invalid attributes and values must be changed and modified in order to reach a feasible solution. The modification process removes the inconsistencies detected during conflict checking so that a consistent and valid design solution can be attained. The checking process will not end until all the violated design constraints have been eliminated.

(2) Ordinary Case Reparation

This stage of adaptation is characterized by eliminating the differences between the specifications of the new design problem and the similar case through adapting the previous solutions of the similar case to satisfy the new problem. During adaptation, if the retrieved similar case can be easily adapted to satisfy the required design specification, an ordinary case reparation procedure will be carried out. In the case reparation, the substitution of attributes and values from the new design specifications to the design case is often required. In addition, a propagation of the effect of the new values on continuous variables with associated procedural attachments also occurs. Two means for ordinary case reparation are defined as follows (Kolodner 1993):

1. Perform simple parameters substitution, i.e. substitute parameters of the old problem for new user input;
2. Perform an old solution adjustment to make it fit the substituted user input according to the designer's knowledge.

(3) Resolving Design Contradictions with TRIZ

While a portion of design specifications can be easily satisfied through ordinary case reparation, some specifications with special performance demand might not be that easy to attain. This is especially the case when a physical contradiction (Savransky 2000) occurs if a recommended solution intensifies the useful function in a subsystem while simultaneously intensifying the existing harmful function in the same key subsystem. Thus, an innovative solution would be highly desirable to solve this problem. For this purpose, the TRIZ contradiction matrix is employed to resolve the design contradictions.

The contradiction matrix is the kernel of TRIZ methodology. In using TRIZ for innovative adaptation, the contradiction matrix is a promising method to deal with the inconsistency between similar cases and the new design specification. The horizontal (row) elements of the contradiction matrix are the engineering parameters to be improved, and the vertical (column) elements contain the engineering parameters that can be adversely affected and/or degraded as a result of improving the parameters. In general, the engineering parameters belong to the three clusters: (1) common physical and geometric parameters; (2) technique-independent negative parameters; (3) technique-independent positive parameters. The numbers at the intersection cells in the matrix guide a solver to a number of inventive principles that might be of help in resolving the contradiction.

In applying the matrix, the improving and worsening engineering parameters, which are used to describe the contradictions, should be chosen in advance. Corresponding

inventive design principles can be determined according to the selected engineering parameters. The principles are a set of heuristic rules which are useful for eliminating conflicts in a design task and creating a high level concept that is a possible inventive solution. These principles are applied to suggest ways in which the conflicts may be resolved. To determine the inventive principles, look at the cell corresponding to the row and column numbers, which are associated with the engineering parameters, and a list of numbers will be located in the cell. These numbers in the cell are the numbers of solution principles. The principles in the matrix cells are presented in order of decreasing frequency of their use in patents selected by Altshuller. Oftentimes, more than one principle can be identified. Since these principles show only the promising direction of the solutions, it would be better to try to apply each of the suggested inventive principles and select the most appropriate one so as to find the best solutions to resolve the contradictions. If all the suggested principles fail to solve the problem, the contradiction needs to be reformulated and resolved.

After that, a new matrix, which integrates the engineering parameters and corresponding inventive principles, is constructed to analyze the problem based on which an innovative solution can be generated. Table 5.1 shows partial cells of Altshuller's contradiction matrix. Based on the matrix and the principle selected, the solutions can be generated by searching the corresponding science effect from the effects database.

Table 5.1 Partial cells of Altshuller's contradiction matrix (Altshuller 1996)

<div style="display: flex; align-items: center; justify-content: center;"> <div style="text-align: right; margin-right: 10px;">Improving Feature ↓</div> <div style="text-align: left; margin-left: 10px;">Worsening Feature →</div> </div>		...	Speed	Force (Intensity)	...	Loss of time	Quantity of substance/the matter	...
		...	9	10	...	25	26	...
1	Weight of moving object		2, 8, 15, 38	8, 10, 18, 37		10, 35, 20, 28	3, 26, 18, 31	
2	Weight of stationary object			8, 10, 19, 35		10, 20, 35, 26	19, 6, 18, 26	
...	...							
14	Strength		8, 13, 26, 14	10, 18, 3, 14		29, 3, 28, 10	29, 10, 27	
15	Duration of action of moving object		3, 35, 5	19, 2, 16		20, 10, 28, 18	3, 35, 10, 40	
...	...							
39	Productivity			28, 15, 10, 36			35, 38	

To sum up, seven steps are involved in the IAP procedure as follows.

- Step 1: Corresponding to the design specifications for a particular design problem, the preliminary potential solutions are derived via the design evolution approach.
- Step 2: Based on the results, identify the constraints and/or contradictions between the proposed design solutions and design specifications.
- Step 3: Form a list of design constraints and contradictions through the conflict checking process.
- Step 4: Apply an ordinary case reparation procedure to resolve the identified constraints and/or contradictions. Two sub-steps could be required: (1) simple parameters substitution; and (2) old solution adjustment. If the ordinary reparation procedure fails, go to Step 5. Otherwise, go to Step 7.

Step 5: Apply the TRIZ solutions to resolve the contradictions. To deal with the contradictions, innovative solutions might be put forward. Three sub-steps are involved: (1) choose the appropriate engineering parameters, i.e. the improving and worsening engineering parameters, to describe the contradictions; (2) select the corresponding inventive principles based on the engineering parameters and Altshuller's contradiction matrix; and (3) form a matrix which incorporates the engineering parameters and corresponding inventive principles to resolve the contradictions. More information on applying the contradiction matrix can be found in Savransky (2000). If TRIZ solutions failed to resolve the contradictions, go to Step 6. Otherwise, go to Step 7.

Step 6: Once the TRIZ techniques fail, the design constraints or contradictions which require innovative solutions, may be left to human capabilities. The designer's own creative thinking will be incorporated to solve the problem.

Step 7: Finally, the feasible solutions can be generated and retained.

5.4 A Case Study

In order to illustrate the capability of the proposed IAP procedure, a case study on hard disk drive (HDD) design is conducted here. The objective of this case is to design the major components of a HDD based on previous design information. Moreover, an innovative solution will be provided to improve the shock protection system for the HDD by applying TRIZ techniques.

Since the major task of this case is to illustrate how the proposed IAP procedure works, it is assumed that the potential solutions of the majority of HDD components can be retrieved using the suggested feature-based case retrieval process (Rao and Chen

2005b, 2006), which was presented in Chapter 4. In this case, such features of a HDD as disk platter, spindle shaft, electromagnetic read/write head, suspension, flexure, carriage, ribbon cable, base plate, pivot, spindle motor and disk clamp are considered to be retrieved from the case base. In the following section, how the TRIZ techniques are used for the innovative adaptation of the hard disk arm and actuator will be demonstrated.

5.4.1 How HDD Works

In a computer system, a disk drive writes and reads information through a read/write head. The head is mounted at the end of an arm that swings over the surface of a rotating disk in much the same way as a phonograph needle and arm reach over a record. The disk is coated with magnetic material. The major components of a hard disk drive consist of at least two electric motors, a spin motor that drives the rotation of the disks and an actuator motor that moves the head to the desired position over the disk, and a variety of electronic circuits that control the drive's operation and its interface with the computer. Figure 5.2 presents a typical hard disk drive.

The read/write head is a tiny electromagnet whose polarity changes whenever the direction of the electrical current running through it changes. Because opposite magnetic poles attract, when the polarity of the head becomes positive, the polarity of the area on the disk beneath the head switches to negative, and vice versa. By rapidly changing the direction of current flowing through the head's electromagnet as the disk spins beneath the head, a sequence of positively and negatively oriented magnetic domains are created in concentric tracks on the disk's surface. Disk drives can use the positive and negative domains on the disk as a binary numeric system -1 and 0- to "write" information onto disks. Drives read information from disks in essentially the opposite process: Changes in

the magnetic flux fields on the disk surface induce changes in the micro current flowing through the head (Christensen 1997).

Figure 5.2 A typical hard disk drive (<http://www.fraserking.co.uk>)

To design a hard disk drive, the major parts to be considered in the design process are listed as follows (<http://www.umich.edu/~archive/msdos/info/disk/hdtech02.txt>):

1. Disk Platter(s), separated by spacers and held together by a clamp.
2. Spindle shaft onto which platters are mounted.
3. Spindle motor for rotating the platters.
4. Electromagnetic read/write heads (one per surface).
5. Access arms or armatures from which the heads are suspended.
6. Actuator for moving the arms (with heads attached).

5.4.2 Problem Description

A hard disk drive has an actuator arm, which can move relative to a magnetic disk that is rotated by an electric motor. The read/write head which can sense the magnetic field of the disk is located at the end of the arm. When the disk is in operation, the head is lifted and separated from the disk surface by airflow generated by the rotating disk. A tiny gap between the head and the disk would be formed. However, any contact between the head and the magnetic disk may cause loss of data (Royzen 1998).

When the disk is not spinning, the head is positioned in a landing zone of the disk platter where no data is stored. Once the computer is switched on, the disk starts spinning and generates the airflow to lift the arm and create the required gap. Then the arm is moved away from the landing zone to read and write data. While the hard drive is not operating, the head and arm are at rest over the dedicated laser-textured landing zone in the disk.

Dropping, striking or bouncing a drive against a hard surface can damage it internally with no external evidence of damage. In these kinds of incidents, the head may be lifted off the disk and then slap back down onto the disk, which is called a head slap. A head slap may cause loss of data. As such, it is highly important to keep the head stay in contact with the disk to prevent “head slap” damage. As shown in Figure 5.3, head slaps may damage the disk and the head and may even generate head or disk particles, which may become lodged under the head and then result in data loss.

In order to avoid the occurrence of a head slap, the following improvements might be made:

1. Reduce the mass of the head.
2. Reduce the size of the head.
3. Increase the thickness of the head-arm.

By reducing the size and mass of the head or increasing the thickness of the head-arm, there is less chance for the head to get in touch with the disk during a shock event. These improvements can increase the amount of nonoperational shock that the HDD can withstand before damage occurs. However, if simply increasing the thickness of the head-arm, it would not only increase the stiffness of the arm, but also increase the inertia that must be overcome to move the head during a seek. This higher inertia increases the time it takes to complete a seek operation, as well as the power required to move the head into place. Based on this understanding, it seems that the hard drive arm design is unable to be adapted from the previous design case via simple ordinary reparations. Thus, an innovative solution is required to improve the reliability of the hard drive without any deterioration of its performance. For this purpose, the TRIZ methodologies are incorporated into the IAP procedure to solve the problem.

Figure 5.3 Head slap on a disk drive (<http://storage.ittoolbox.com>)

5.4.3 Problem Solving with TRIZ

Applying an external force to the drive (such as moving or knocking the computer), can move the head away from the landing zone and destroy data on the disk. Damage to the head and the disk occurs when these two components contact each other. The occurrence of a head slap could be avoided by replacing with a thicker head arm. However, as mentioned previously, when the thickness of the head arm increases, the inertia that must be overcome to move the head also increases. Thus, an attempt to improve the reliability of the hard drive causes deterioration of its performance. The contradiction in this case is the shape of the arm and its mobility. Hence, the Contradiction Table (Alshuller's Matrix) is chosen to resolve the contradiction in this case.

To establish the contradiction matrix, the right engineering parameters should be chosen from Alshuller's matrix to describe the contradiction. In the HDD, to prevent the head arm from moving while the computer is not in use, the arm should possess a certain kind of ability to resist changing in response to the external force. Hence, the engineering parameter 'Strength (No.14)' is identified as the improving engineering parameter. Similarly, since the deterioration of the HDD's performance caused by improving the reliability of the HDD, has a large influence on the operation of the HDD, the parameter 'Ease of operation (No. 33)' is chosen to be the worsening engineering parameter. Thus, both the engineering parameters of 'Strength (No.14)' and 'Ease of operation (No. 33)' are chosen to describe the contradiction correctly. After that, the corresponding inventive principles can be selected based on the Altshuller's contradiction matrix. From the matrix, the inventive principles can be located in the cell of intersection between the horizontal row (corresponding to parameter 'No.14') and vertical column (corresponding to parameter 'No.33'). As a result, principles 32, 40, 25

and 2 are selected as the potential principles to resolve the contradiction. Table 5.2 shows the problem solving process with the corresponding inventive principles selected.

Table 5.2 Resolving the problem with corresponding inventive principles

Contradiction Identification	To prevent the arm from moving while the computer is not in use (Improved)	Attempt to improve the reliability of the hard drive causes deterioration of its performance (Degraded)
The chosen engineering parameter describing the contradiction	No.14. Strength (Ability to resist changing in response to force)	No. 33. Ease of operation (The process is not easy as it requires many steps and needs special tools)
The corresponding inventive principles	No. 32, No. 40, No. 25, No.2	
To resolve the problem with the inventive principles	No. 32 Color change No. 40 Composite materials No. 25 Self-Service <ul style="list-style-type: none"> • Make the object serve itself by performing auxiliary helpful functions • Use waste resources, energy, or substances. No. 2 Taking out	

Among the four proposed inventive principles, No. 25 (Self-service) is considered to be the most appropriate one because it has the best possibility of resolving the problem. The kernel of 'Self-service' is to make an object serve itself by performing auxiliary helpful functions. In this case, it means that the head arm should serve itself to park the head at the right place and thus avoid a head slap. To meet this requirement, the TechOptimizerTM, which is a software package provided by Invention Machine

Corporation to assist in solving innovative engineering problems based on TRIZ methodology, is used to search the effects database for the appropriate science effect (Figure 5.4). According to the suggested science effects, magnetic force, which is a kind of physical effect, presents a logical alternative to provide a field force to correct the mispositioning of the head arm.

Figure 5.4 A screenshot of searching a science effect for self-service

Based on this understanding, a device that allows compensation for the head mispositioning, which is caused by the internal/external physical vibration or shock to the hard disk, is suggested to provide the magnetic force that places the head arm at the correct position. This device is a capacitive rotational accelerometer sensor, which makes use of a micro-electro-mechanical system (MEMS). With the accelerometer

sensor, any tiny rotational displacements of the parts in the MEMS structure can be detected by sensing the small variations in capacitance caused by the displacements. A feed forward correction signal is then generated from the voice coil drive circuit to keep the head at the right position. The MEMS accelerometer can be deployed over the read/write head arm or the actuator. By adopting the MEMS accelerometer-based motion sensor system, the risk of a HDD crash can be minimized. Figure 5.5 shows a sketch of the final conceptual design of the HDD which indicates the MEMS accelerometer sensor on the head arm.

Figure 5.5 Sketch of open HDD with accelerometer sensor
(adapted based on Tandon et al. 2006)

5.5 Summary

Product innovation is highly important for the success of new product development (NPD). While CBR has been recognized as a promising methodology to employ past

design experiences to solve new design problems, it has limited capability in creating innovative designs. This chapter presents a TRIZ-enhanced CBR approach which focuses on the adaptation stage of CBR so as to enhance the innovation capability for the proposed CBR-based design evolution system. Accordingly, an innovative adaptation procedure (IAP) incorporating TRIZ techniques is proposed. The IAP procedure consists of three major parts, namely Conflict Checking, Ordinary Case Reparation and Resolving Design Contradictions with TRIZ. Initially, the IAP performs as a constraint satisfaction process, in which an ordinary case reparation process might be applied if the derived potential solutions can be easily adapted to satisfy the new design requirements. Otherwise, an innovative solution should be sought to solve the inconsistency between the potential case and new design specifications, especially when a contradiction exists. In order to provide the innovative solution, the TRIZ methodology, which contains a set of powerful problem solving tools for product design innovation, is employed in the IAP procedure.

With the TRIZ-enhanced IAP, it is expected that the adaptation function, which is a challenging part of the CBR process, can be improved. The IAP equips the CBR system with the capability of suggesting innovative solutions to deal with the inconsistencies between the new design specifications and the potential solutions derived from the case repository. This greatly assists the CBR-based design evolution approach to generate innovative product concepts, despite it being based on historical design information. A case study on hard disk drive design is employed in this work to demonstrate how the IAP procedure works. It appears that the proposed TRIZ enhanced IAP is able to assist designers to put forward innovative solutions in handling design contradictions.

CHAPTER 6 A MATRIX REPRESENTATION AND MAPPING (MRM) APPROACH FOR KNOWLEDGE ACQUISITION

6.1 Introduction

Product design is a complex process that involves a variety of knowledge and considerations. A knowledge-based approach is often employed to provide the knowledge support for a design process. As reviewed in Chapter 2, the knowledge-based approach has a lot of applications in the domain of product design. It presents a promising and logical alternative to deal with the decision making issues in product design, especially in design evaluation (Chen et al. 2001). In constructing a knowledge-based system, knowledge acquisition plays a critical role. However, it is well-known that knowledge acquisition is the most time-consuming phase, and hence the bottleneck in constructing a knowledge-based system (Wagner et al. 2002, Shao et al. 2001). Generally, a knowledge acquisition process covers the activities of collecting domain knowledge and transforming it into a form that can be processed by a computer. The barriers in knowledge acquisition are threefold: (1) the difficulty in soliciting knowledge from domain expert(s) due to inexperienced knowledge engineer(s) used; (2) the difficulty in extracting useful knowledge from the solicited domain knowledge owing to the limitation and bias of human cognition; and (3) the difficulty in conveying knowledge from a human to a machine due to the different knowledge representation formats.

A number of researchers have suggested various measures to tackle the problems encountered in knowledge acquisition. From the aspect of facilitating the efficiency of

knowledge acquisition, Chen and Occeña (1999) proposed a so-called knowledge sorting process. The process capitalizes on the relationships between attributes and factors, dependent and independent variables, and interrelationships between attributes. It consists of three consecutive stages, viz. identification of knowledge sources, generation of taxonomic trees, and organization of acquired knowledge, to extract rules from the sorted knowledge. Yan et al. (2005) established a QFD (quality function deployment) - enabled product conceptualization system, in which design knowledge is elicited using the laddering technique, represented using a so-called design knowledge hierarchy and organized using the RCE (restricted coulomb energy) neural network. Rao and Chen (2005c) conducted a preliminary investigation into a matrix representation and mapping approach to facilitate the process of knowledge acquisition. Other related work on knowledge acquisition has been reviewed in Chapter 2. However, there are two major omissions in existing work. First, the quantitative characteristics of information and interrelationships between the various knowledge elements, which is of critical importance in constructing knowledge-based systems, has not been well addressed. Second, it is imperative to provide an integrated and complete knowledge acquisition process with a well-organized knowledge representation scheme to facilitate the effectiveness of rule generation. Moreover, complex design information and various knowledge sources are often involved in the multi-attribute design evaluation process. To acquire the required evaluation knowledge from different knowledge sources, an approach to facilitate the process of knowledge acquisition is highly desirable.

Based on this understanding, a knowledge acquisition approach using matrix representation and mapping (MRM) (Rao and Chen 2005c) for constructing knowledge-based systems is proposed in this chapter. Section 6.2 provides an overview of the proposed MRM approach. Section 6.3 introduces the fundamental concepts of the

approach. Section 6.4 describes the procedure of the approach in detail. Section 6.5 presents a case study on the primary diagnosis of automotive systems to illustrate how the proposed approach works. The last section, Section 6.6, summarizes the conclusions reached in this chapter.

6.2 Overview of the MRM Approach

A knowledge-based system is a computer program used to solve problems requiring expert knowledge in some application domains. In order to construct such a system, knowledge is elicited from domain expert(s) or related literature. The proposed MRM approach for knowledge acquisition begins by identifying the appropriate knowledge sources, which may include domain expert(s), historical records, related literature, etc. After the relevant knowledge is acquired from different knowledge sources, it is preliminarily organized in a taxonomic manner with a general sorting process. Based on the taxonomic trees generated, a series of matrix representation and mapping operations will be performed to extract the rules. The proposed approach involves six consecutive steps, namely, identification of knowledge sources, generation of taxonomic tree, identification of linkages between the conditions and conclusions sets, generation of sub-clauses to form IF and THEN clauses, identification of logical interrelationships among sub-clauses and generation of rules. Figure 6.1 shows the overall procedure of the MRM approach.

Figure 6.1 Overall procedure of the MRM approach

6.3 Fundamental Concepts of the MRM Approach

In building a knowledge-based system, rule is an important and widely used knowledge representation scheme. People find it is natural to express knowledge using the IF-THEN format. The IF portion of a rule is a condition, which tests the truth values

of a set of facts. If these are found to be true, the THEN portion of a rule is inferred as a new set of facts. Figure 6.2 shows a conceptual process for rule formation.

$$\begin{array}{c} \text{Rules} \\ \hline \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{bmatrix} \end{array} = \begin{array}{c} \text{IF} \\ \hline \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} \end{array} + \begin{array}{c} \text{THEN} \\ \hline \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \end{array}$$

Figure 6.2 A conceptual process for rule formation

In the MRM approach, the rule generation is defined as the mapping process between the conditions domain and conclusions domain. This process can be characterized mathematically. The characteristics of the conditions domain are represented by a set of IF clauses, and these may be treated as a vector with n elements. Similarly, the THEN clauses in the conclusions domain also constitute a vector with n elements. As a result, the rule generation process involves selecting a set of right conditional IF clauses to deduce a set of corresponding conclusive THEN clauses, which can be expressed as:

$$\mathbf{Y}_{\text{THEN}} = \mathbf{M} \mathbf{X}_{\text{IF}} \quad (6-1)$$

where $\mathbf{Y}_{\text{THEN}}$ is the vector comprising the THEN clauses in the conclusions domain; $\mathbf{X}_{\text{IF}}$ is the vector comprising the IF clauses in the conditions domain; $\mathbf{M}$ is the mapping matrix.

Equation (6-1) can also be written as:

$$[Y_1 \ Y_2 \ \dots \ Y_n]^T = \mathbf{M} [X_1 \ X_2 \ \dots \ X_n]^T \quad (6-2)$$

where $Y_{\text{THEN}} = [Y_1 \ Y_2 \ \dots \ Y_n]^T$, and $X_{\text{IF}} = [X_1 \ X_2 \ \dots \ X_n]^T$. The mapping matrix comprises a relationship array and each element of the matrix relates an element of the IF vector to an element of THEN vector.

6.4 Description of the MRM Approach

6.4.1 Identification of knowledge sources

The knowledge sources required to build a knowledge-based system may include domain experts, historical records, and related literature, etc. In constructing a specific knowledge-based system, knowledge engineers should identify the appropriate knowledge sources required, and then plan for strategies to acquire knowledge from these sources.

6.4.2 Generation of taxonomic tree (general sorting)

After the knowledge is elicited from various knowledge sources, the acquired knowledge is organized in a taxonomic manner to facilitate the efficiency of subsequent activities such as rule generation. Well-organized knowledge will make the extraction of rules easier and more effective. A general sorting process (Benfer et al. 1991) is adopted to generate the taxonomic tree(s) for organizing the knowledge structure of a specific problem. The basic notion behind general sorting techniques is a process where objects such as terms are sorted into groups by domain experts or designers. The sorting process is one way of learning how the vocabulary of a domain is organized. It helps the developer to identify covert categories and provide an excellent framework for use in

discussing the organization of the knowledge with the expert (Rugg and McGeorge 1997, 2005).

Sorting techniques has been widely used in knowledge and requirements acquisition in product design. Chen and Occeña (1999) reported a knowledge sorting process that was developed to facilitate the rule extraction for a product design expert system. Chen et al. (2002a) established a prototype web-based customer-oriented product concept formation system, which involved customers in generating functional attributes and design alternatives as well as selecting preferred design solutions. In the system, the laddering technique was employed to set up a so-called functional attribute hierarchy comprising different levels ranging from generic to specific abstraction. Chen et al. (2003) developed a prototype system that combines the strengths of the laddering technique and the radial basis function (RBF) neural network for customer requirements acquisition and multicultural factors evaluation. Chen et al. (2005) proposed a product definition and customization system, in which the laddering technique is used to acquire designer's knowledge for product definition. Chen et al. (2006) employed a sorting technique, i.e. picture sorts, for acquiring customer's affective requirements in a prototype system, which emphasizes the solicitation of affective attributes from customers.

In addition, the general sorting technique has been employed to form a generic product platform, so-called design attributes hierarchy, in product conceptualization (Yan et al. 2006). As a well-established technique, general sorting has the following advantages in knowledge organization (Yan et al. 2006):

1. It has a wide coverage of domain, yet requires less 'effort' (in terms of mean total time for elicitation and recording) than non-contrived techniques (e.g. interview and observation).

2. It is flexible and simple for most novice elicitors to utilize; it assists elicitors to identify covert categories; and it provides an excellent prop for use in discussing the categorization of knowledge with domain experts.
3. It also requires less effort to transform outputs into alternative formats of hierarchies, and imposes more categorical hierarchies or discrete classes.

The adopted general sorting process consists of the following three consecutive steps (Chen and Occeña 1999, Yan et al. 2006):

- (1) *Obtain the terms used for the specific problem.* The terms can be elicited from the domain expert(s) and/or acquired from related literature, historical records and/or any other sources. The result of this step is a list of terms used for the specific problem.
- (2) *Sort the terms with certain criteria.* An elicitor asks domain experts themselves to sort and subdivide the terms until no terms can be divided any further. The elicitor then determines the reasons or criteria by which experts grouped or separated the terms.
- (3) *Generate taxonomic trees.* Once the terms have been sorted, they are organized into a hierarchical structure (or taxonomic tree). More than one taxonomic tree may be required if many terms are used for the specific problem.

6.4.3 Identification of linkages between conditions and conclusions sets

Based on the taxonomic tree generated in the previous step, the attributes and their corresponding values can be extracted and then classified. These extracted attributes and values are classified into two categories, i.e. a conditions set and a conclusions set. As an example, for a specific design problem, the attributes and corresponding values

representing the design information, such as specifications and design parameters, can be classified into the conditions set. On the other hand, the attributes and corresponding values representing the design task or objectives can be classified into the conclusions set. However, the classification of the attributes and corresponding values can be exchanged between the conditions set and conclusions set. That is to say, if the attributes and corresponding values representing the information on design specifications and parameters are classified into the conclusions set, the attributes and corresponding values representing the design task or objectives will be classified into the conditions set and vice versa. The attributes and corresponding values in the conditions set are used to represent the IF clauses in rule generation, while those in the conclusions set are used to represent the THEN clauses in rule generation.

Let S_{CD} denote the set of arrays representing attributes and their corresponding values in the conditions set;

S_{CC} denote the set of arrays representing attributes and their corresponding values in the conclusions set.

$$S_{CD} = \{[a_{xp}, u_{xp}] \mid p = 1, 2, \dots, m\}$$

$$S_{CC} = \{[b_{yq}, v_{yq}] \mid q = 1, 2, \dots, n\} \quad (6-3)$$

where a_{xp} and u_{xp} denote the attributes and their corresponding values in the conditions set, respectively; b_{yq} and v_{yq} stand for the attributes and their corresponding values in the conclusions set, respectively. According to Equations (6-1) and (6-2), in order to generate the rules, the relationship and linkage between the conditions set and conclusions set need to be identified. Therefore, an array set to express the linkages of the attributes and values between the conditions set and conclusions set is defined below.

Let D represent the array set expressing linkages of attributes and values between the conditions set and conclusions set

$$\begin{aligned} D &= \{ S_{CD} \dot{ : } S_{CC} \} \\ &= \{ [a_{xp}, u_{xp} \dot{ : } b_{yq}, v_{yq}] \mid p = 1, 2, \dots, m; q = 1, 2, \dots, n \} \\ &= \{ [a_{xp}, b_{yq} \dot{ : } u_{xp}, v_{yq}] \mid p = 1, 2, \dots, m; q = 1, 2, \dots, n \} \end{aligned} \quad (6-4)$$

To obtain the relationship between S_{CD} and S_{CC} , the corresponding attributes and values in the conditions set and conclusions set are compared respectively. Those attributes and values which have significant relationships between the conditions set and conclusions set will be selected to generate the rules. The relationships are determined by taking into account the related design knowledge and/or experts' opinions, which are elicited from the selected knowledge sources for a specific problem.

To facilitate the relationship determination, a relationship estimation procedure (Owen 1986), which is derived from a methodology developed by Sato and Owen (1981), is employed to deduce the strength of relationship between each pair of knowledge attributes/values. In Owen's methodology, the interactions between any two functions that a system must perform are categorized into five regions: (1) f: reinforcement (+,+); (2) g: conflict (+,-); (3) s: independence (+,0); (4) h: conflict (-,+); and (5) t: independence (0,+). The measurement of the interaction between any two functions is estimated as the ratio of interaction areas (reinforcement and conflict areas) to the entire area of five regions. Figure 6.3 illustrates the concept of the interactions between each pair of knowledge attributes/values. As shown in the figure, the reinforcement region (+, +) means that both the knowledge attributes/values KA_u/KV_u and KA_v/KV_v have positive effects on a task of the knowledge-based system to be constructed, whereas the conflict region (+, -) means that knowledge attribute/value

KA_u/KV_u has a positive effect on a specific system task yet knowledge attribute/value KA_v/KV_v has a negative effect on it. For example, in developing a knowledge-based system for wood golf club design (Chen and Occeña 1999), suppose the system task ST_k is to reduce the wind resistance, then the knowledge attribute KA_u (smoothness of the crown curvature) has a positive effect on ST_k , whereas the knowledge attribute KA_v (requirement for increasing the sole width) has a negative effect on it.

Figure 6.3 Interactions between knowledge attributes/values

In terms of Owen's methodology, the interaction between each pair of knowledge attributes/values (KA_u/KV_u , KA_v/KV_v) can fall into only one of the five regions. The strength of the dependence can be measured as follows (Sato and Owen 1981, Chen et al. 2004):

$$V_k^f = \text{Reinforcement (f region, + +)} = (a_k + b_k - r_k) / r_k$$

$$V_k^g = \text{Conflict (g region, + -)} = |a_k - b_k| / r_k$$

$$V_k^h = \text{Conflict (h region, - +)} = |a_k - b_k| / r_k$$

$$V_k^s = \text{Independence (s region, + 0)} = (a_k - r_k/2) / (r_k/2)$$

$$V_k^t = \text{Independence (t region, 0 +)} = (a_k - r_k/2) / (r_k/2)$$

where $a_k = a'_k - \min_k$, $b_k = b'_k - \min_k$, a'_k and b'_k represent the total effects of knowledge attributes/values KA_u/KV_u and KA_v/KV_v on the system task ST_k . The effects are assessed using scaled judgments. Table 6.1 shows the numerical scale of the effects. Figure 6.4 shows a matrix that contains sample results of the effects. $\max_k$ and $\min_k$ are the maximum and minimum values among knowledge attributes/values for system task ST_k ; r_k stands for the range of effect for ST_k and $r_k = \max_k - \min_k$.

Table 6.1 Scaled judgments of various effects (a'_k and b'_k)

+2	Strongly supports
+1	Supports
0	No effects
-1	Obstructs
-2	Strongly obstructs

	ST ₁	ST ₂	...	ST _K	
Importance	W_I	W_2	...	W_K	Strength of effect
KA ₁ /KV ₁	●	○	...	△	● Strongly positive
KA ₂ /KV ₂	○	●	...	○	● Positive
⋮	⋮	⋮	⋮	⋮	△ No effect
KA _U /KV _U	●	△	...	●	○ Negative
					○ Strongly negative

Figure 6.4 Sample effects of knowledge attributes/values on system tasks

As a result, the overall relationships between two knowledge attributes/values can be obtained using Equation (6-5):

$$R_{uv} = \frac{\sum_{k=1}^K W_k (V_{uvk}^g + V_{uvk}^f + V_{uvk}^h)}{\sum_{k=1}^K W_k (V_{uvk}^s + (\delta_{uvk}^g + \delta_{uvk}^f + \delta_{uvk}^h) + V_{uvk}^t)} \quad (6-5)$$

where W_k represents the importance of the system task ST_k . It adopts numerical values 1-5 to estimate the scaled importance, where 1 means the least importance and 5 means the most importance. δ_{uvk}^g , δ_{uvk}^f and δ_{uvk}^h are regional factors. According to Owen's methodology, their values are defined as follows: (1) If for ST_k , the interaction between knowledge attributes/values KA_u/KV_u and KA_v/KV_v falls into Region g(+ -), then $\delta_{uvk}^g = 1$; otherwise $\delta_{uvk}^g = 0$. (2) If for ST_k , the interaction between knowledge attributes/values KA_u/KV_u and KA_v/KV_v falls into Region f(+ +), then $\delta_{uvk}^f = 1$; otherwise $\delta_{uvk}^f = 0$. (3) If for ST_k , the interaction between knowledge attributes/values KA_u/KV_u and KA_v/KV_v falls into region h(- +), then $\delta_{uvk}^h = 1$; otherwise $\delta_{uvk}^h = 0$.

Consult the domain expert(s) and define a threshold value θ . If $R_{uv} \geq \theta$, the corresponding knowledge attributes and values will be considered to have significant relationships. Subsequently, the two matrices, $P_{m \times n}$ and $Q_{m \times n}$, shown in Figure 6.5 are defined to record the relationships between the conditions set and conclusions set for attributes and values respectively. For simplification, the numbers "1" and "0" are used in both of the matrices to indicate the strength of relationship. Where "1" represents if and only if the corresponding attributes and values have significant relationships between the two sets, i.e. $R_{uv} \geq \theta$, otherwise, "0" would be assigned.

	b_{Y1}	b_{Y2}	$ \dots$	$b_{Y(n-1)}$	b_{Yn}
a_{X1}	0	1	$ \dots$	0	1
a_{X2}	1	0	$ \dots$	1	0
$\vdots$	$\vdots$	$\vdots$	$ \dots$	$\vdots$	$\vdots$
$a_{X(m-1)}$	0	0	$ \dots$	1	0
a_{Xm}	0	1	$ \dots$	1	0

	v_{Y1}	v_{Y2}	$ \dots$	$v_{Y(n-1)}$	v_{Yn}
u_{X1}	1	0	$ \dots$	0	1
u_{X2}	0	1	$ \dots$	1	0
$\vdots$	$\vdots$	$\vdots$	$ \dots$	$\vdots$	$\vdots$
$u_{X(m-1)}$	1	0	$ \dots$	1	0
u_{Xm}	0	1	$ \dots$	1	0

(a) $P_{m \times n}$: attributes relationship

(b) $Q_{m \times n}$: values relationship

Figure 6.5 Sample relationship matrices for attributes and values

6.4.4 Generation of sub-clauses of IF and THEN

The purpose of the previous step is to select those related attributes and values that have significant relationships between the conditions set and conclusions set. Thus, the conditions set and conclusions set can be linked via those related attributes and values selected. The combinations of the selected attributes and values are expressed as Equation (6-6).

$$D_{AB} = \{ [a_{Xp_a}, u_{Xp_a} : b_{Yq_b}, v_{Yq_b}] | p_a \in p, q_b \in q \} \quad (6-6)$$

where D_{AB} represents the set of combinations of the selected attributes and values; p and q denote the set of array numbers in the conditions set S_{CD} and conclusions set S_{CC} respectively, $p=1,2,\dots, m, q=1,2,\dots, n$. Since the elements of D_{AB} are selected from D , D_{AB} is a subset of D , i.e. $D_{AB} \subseteq D$.

IF and THEN clauses of rules can be divided into a group of sub-clauses that are composed of the selected attributes and their corresponding values.

Let A_X denote the set of the combinations of the attributes and values used to represent the IF clauses

$$\begin{aligned} A_X &= \{ A_{Xp_a} \mid p_a \in p \} \\ &= \{ [a_{Xp_a}, u_{Xp_a}] \mid p_a \in p \} \end{aligned} \quad (6-7)$$

B_Y denotes the set of the combinations of the attributes and values used to represent the THEN clauses

$$\begin{aligned} B_Y &= \{ B_{Yq_b} \mid q_b \in q \} \\ &= \{ [b_{Yq_b}, v_{Yq_b}] \mid q_b \in q \} \end{aligned} \quad (6-8)$$

The elements of A_X and B_Y are those related attributes and values selected from the conditions set and conclusions set, hence, $A_X \subseteq S_{CD}$ and $B_Y \subseteq S_{CC}$ can be derived. According to Equations (6-7) and (6-8), the sub-clauses used to represent the IF clauses can be generated through the combinations of $[a_{Xp_a}, u_{Xp_a}]$. Similarly, the sub-clauses used to represent the THEN clauses can be generated through the combinations of $[b_{Yq_b}, v_{Yq_b}]$. Consequently, the representation of the sub-clauses of the IF and THEN clauses are defined as follows:

Let X denote the vector set of IF clauses for rule generation

$$X = \{ X_i = \{ A_{Xij_1} \mid j_1 = 1, 2, \dots, m_1 \} \mid i = 1, 2, \dots, n \} \quad (6-9)$$

Y denote the vector set of THEN clauses for rule generation

$$Y = \{ Y_i = \{ B_{Yij_2} \mid j_2 = 1, 2, \dots, m_2 \} \mid i = 1, 2, \dots, n \} \quad (6-10)$$

where X and Y denote the vector sets of IF and THEN clauses for rule generation, respectively; j_1 and j_2 are the number of sub-clauses in the IF and THEN clauses, $j_1=1,2,\dots,m_1$ and $j_2=1,2,\dots,m_2$; A_{Xij_1} and B_{Yij_2} denote the individual sub-clauses in the IF clauses and THEN clauses, respectively.

6.4.5 Identification of logical relationships between sub-clauses

The rules of a knowledge-based system are usually generated using IF and THEN clauses, which in turn can be produced by combining the sub-clauses according to certain logical relationships. Moreover, due to the uncertain and imprecise information that is often involved in knowledge-based systems, an uncertainty handling mechanism should be provided to deal with the confidence level of the generated rules. These issues are addressed in more detail below.

(i) *Mechanism for uncertainty handling.* A number of methods, such as MYCIN (an expert system for diagnosis of infectious diseases)-based certainty factors (CF) (Shortliffe 1976), Bayesian probability (Sen 1976, Guan et al. 1997), Dempster Shafer method (Shafer 1976), and Zadeh's fuzzy logic (Zadeh 1975) have been proposed to handle uncertainty in knowledge and data. The particular advantages and disadvantages with respect to these uncertainty handling methods were summarized in Gonzalez and Dankel (1993). Compared with other methods, the CF formalism is a popular tool for uncertainty handling in knowledge-based system with specific advantages. As pointed out by Gonzalez and Dankel (1993), firstly, the CF formalism permits the expression of belief and disbelief in each hypothesis, allowing the expression of the effect of multiple sources of evidence. Secondly, it allows knowledge to be captured in a rule representation while allowing the quantification of uncertainty. Thirdly, compared with other methods, the gathering of the CF values is significantly easier and no statistical base is required. These advantages are especially suitable for the problem in which many factors and complex knowledge are involved. Therefore, it appears that the CF formalism is a good choice for handling uncertainty in the MRM approach. Based on the reasoning mechanisms of MYCIN and PROSPECTOR (a consultation expert system for

mineral exploration, Duda et al. 1976), the following reasoning algorithms for handling uncertainty are employed in the proposed MRM approach.

As aforementioned, a rule consists of a condition part and a conclusion part, and is expressed using IF-THEN format. The IF and THEN respectively comprises a series of sub-clauses among which certain logical relationships exist. The certainty factor of a rule indicates the truth value of the conclusions when the conditions are true. According to the sub-clause representations in Equations (6-9) and (6-10), in a knowledge-based system using CF formalism, knowledge is expressed as a set of rules which has the following format:

$$\begin{aligned} \text{Rule } i: & \text{ IF } \{ A_{Xi1}, A_{Xi2}, A_{Xi3}, \dots, A_{Xim_1} \} \\ & \text{ THEN } \{ B_{Yi1}, B_{Yi2}, B_{Yi3}, \dots, B_{Yim_2} \} (CF_{ri}) \end{aligned}$$

where $\{ A_{Xi1}, A_{Xi2}, A_{Xi3}, \dots, A_{Xim_1} \}$ represents the different sub-clauses of IF clauses; $\{ B_{Yi1}, B_{Yi2}, B_{Yi3}, \dots, B_{Yim_2} \}$ represents the different sub-clauses of THEN clauses; and CF_{ri} denotes the certainty factor of the rule.

Let A_i denote the set of sub-clauses of IF part of rule i , and B_i denote the set of sub-clauses of THEN part of rule i , then $A_i = \{ A_{Xi1}, A_{Xi2}, A_{Xi3}, \dots, A_{Xim_1} \}$ and $B_i = \{ B_{Yi1}, B_{Yi2}, B_{Yi3}, \dots, B_{Yim_2} \}$. According to CF formalism,

$$CF_{ri} = CF_{A_i} * CF_{B_i} \quad (6-11)$$

where CF_{A_i} and CF_{B_i} denote the certainty factors of A_i and B_i respectively. Note that the CF formalism described here is for generic cases and CF_{A_i} is usually assigned the value “1” when the IF portion of a rule is regarded as absolutely certain and true.

To form IF and THEN clauses, the logical relationships among sub-clauses of IF and THEN are often represented using the connectives “and” and “or”. If there are only

conjunctions with “and” relationships existed among the sub-clauses of the set A_i and B_i , the certainty factors of A_i and B_i of rule i are respectively,

$$\begin{aligned} CF_{A_i} &= \text{MIN} (CF_{A_{Xi1}}, CF_{A_{Xi2}}, CF_{A_{Xi3}}, \dots, CF_{A_{Xim\eta}}) \\ CF_{B_i} &= \text{MIN} (CF_{B_{Yi1}}, CF_{B_{Yi2}}, CF_{B_{Yi3}}, \dots, CF_{B_{Yim2}}) \end{aligned} \quad (6-12)$$

where $CF_{A_{Xi1}}, CF_{A_{Xi2}}, CF_{A_{Xi3}}, \dots, CF_{A_{Xim\eta}}$ are the certainty factors of different sub-clauses in A_i ; and $CF_{B_{Yi1}}, CF_{B_{Yi2}}, CF_{B_{Yi3}}, \dots, CF_{B_{Yim2}}$ are the certainty factors of different sub-clauses in B_i . If there are only disjunctions with “or” relationships that exist among the sub-clauses of the set A_i and B_i , the certainty factors of A_i and B_i of rule i are respectively,

$$\begin{aligned} CF_{A_i} &= \text{MAX} (CF_{A_{Xi1}}, CF_{A_{Xi2}}, CF_{A_{Xi3}}, \dots, CF_{A_{Xim\eta}}) \\ CF_{B_i} &= \text{MAX} (CF_{B_{Yi1}}, CF_{B_{Yi2}}, CF_{B_{Yi3}}, \dots, CF_{B_{Yim2}}) \end{aligned} \quad (6-13)$$

However, the logical relationships among the sub-clauses often contain both “and” and “or”. In that case, it can be considered that there exist only the conjunctions with “and” relationships when combining the sub-clauses with “or” relationships into individual elements. Therefore, the certainty factors of A_i and B_i of rule i are respectively,

$$\begin{aligned} CF_{A_i} &= \text{MIN} (CF_{A_{Xi1}}, CF_{A_{Xi2}}, \text{MAX} (CF_{A_{Xi3}}, \dots, CF_{A_{Xik}}), \dots, CF_{A_{Xim\eta}}) \\ CF_{B_i} &= \text{MIN} (CF_{B_{Yi1}}, CF_{B_{Yi2}}, \text{MAX} (CF_{B_{Yi3}}, \dots, CF_{B_{Yik}}), \dots, CF_{B_{Yim2}}) \end{aligned} \quad (6-14)$$

where, as an instance, disjunctions with “or” relationships only exist in $(A_{Xi3}, \dots, A_{Xik})$ and $(B_{Yi3}, \dots, B_{Yik})$.

(ii) *Representation of logical relationships.* For simplification, two symbols “ $\wedge$ ” and “ $\vee$ ” are used to represent the connectives “and” and “or”, respectively. In addition, the symbol “-” is used to represent no direct logical relationships between the corresponding

elements. As a result, the logical relationships between different sub-clauses can be illustrated in two matrices, viz. $IR_{A_{Xi}}$ and $IR_{B_{Yi}}$, which are represented as two triangular tables shown in Figure 6.6. The relationships contained in each matrix are presented in a triangular relationship table similar to that employed in quality function deployment (QFD) (Bossert 1991). The elements of the triangular table represent the logical relationships between the sub-clauses. The certainty factors associated with each sub-clause are shown in parentheses. Subsequently, all the rules can be generated through combining different sub-clauses according to the logical relationships shown in the tables.

Figure 6.6 Sample logical relationships of sub-clauses in A_{Xi} and B_{Yi}

6.4.6 Generation of rules

Based on the two matrices $IR_{A_{Xi}}$ and $IR_{B_{Yi}}$ shown in Figure 6.6, the IF and THEN clauses can be determined and, consequently, the rules for building a knowledge-based system can be generated. For instance, a rule can be expressed as:

Rule i : IF $A_{Xi1}(CF_{A_{Xi1}})$ and $A_{Xi2}(CF_{A_{Xi2}})$...or ... $A_{Xim_1}(CF_{A_{Xim_1}})$

THEN $B_{Yi1}(CF_{B_{Yi1}})$ and $B_{Yi2}(CF_{B_{Yi2}})$...or $B_{Yim_2}(CF_{B_{Yim_2}})$

Thereafter, the CF_{ri} of rule i can be obtained using Equations (6-11), (6-12), (6-13) and (6-14).

As a simple example to illustrate the process of rule generation, based on Equations (6-9) and (6-10), assume that $i = 1, j_1 = 4, j_2 = 3$, X_1 and Y_1 represent the IF and THEN clauses of Rule 1, respectively. Thus, $A_{X11}, A_{X12}, A_{X13}, A_{X14}$ are the sub-clauses represent the IF clauses in Rule 1, and $B_{Y11}, B_{Y12}, B_{Y13}$ are the sub-clauses represent the THEN clauses in Rule 1. The logical relationships between the sub-clauses are shown in $IR_{A_{X1}}$ and $IR_{B_{Y1}}$ (Figure 6.7). Assume that the certainty factors associated with each sub-clause are given in the parentheses in the rectangles of the table.

(a) $IR_{A_{X1}}$: Logical interrelationships of sub-clauses in A_{X1} (b) $IR_{B_{Y1}}$: Logical interrelationships of sub-clauses in B_{Y1}

Figure 6.7 Logical relationships of sub-clauses in A_{X1} and B_{Y1}

According to the matrices $IR_{A_{X1}}$ and $IR_{B_{Y1}}$, the IF clauses and THEN clauses can be derived and hence Rule 1 is generated as follows.

Rule 1: IF A_{X11} (0.90) and A_{X12} (0.85) or A_{X13} (0.80) and A_{X14} (0.95)

THEN B_{Y11} (0.88) and B_{Y12} (0.90) or B_{Y13} (0.95)

The certainty factor of Rule 1 is calculated as follows:

$$CF_{A_1} = \text{MIN} (0.90, \text{MAX} (0.85, 0.80), 0.95) = 0.85$$

$$CF_{B_1} = \text{MIN} (0.88, \text{MAX} (0.90, 0.95)) = 0.88$$

$$CF_{r1} = CF_{A_1} * CF_{B_1} = 0.85 * 0.88 = 0.748$$

6.5 A Case Study

A case study on primarily diagnosing possible problems encountered in an automotive system is used to demonstrate the procedure of the proposed MRM approach. The knowledge involved in the case study is represented by a set of rules that identify the performance characteristics of an automotive system. The rule sets will be generated using the MRM approach proposed. Following the procedure described in Section 6.4, the case study is presented as follows:

Step 1. Sources of knowledge. In this case, relevant knowledge can be acquired from automotive domain experts, historical records, or related literature, etc., some of which can easily be obtained from the Internet.

Step 2. Taxonomic tree generation. After being collected from different knowledge sources, the acquired knowledge can be organized in a taxonomic manner. A variety of terms used to define an automotive system are elicited. These terms are further divided and categorized into different groups according to certain criteria. Figure 6.8 shows the

taxonomic tree generated after the general sorting process. As shown in the tree, the problems are most often found in the six components, namely sparking plugs, fuel tank, engine, battery, cables and starter motor.

Figure 6.8 The taxonomic tree generated from general sorting

Step 3. Identification of linkages between the conditions and conclusions sets. Based on the taxonomic tree shown in Figure 6.8, the attributes and values describing all

possible problems can be extracted. For instance, the attribute “Light” has two possible values, i.e. “come on” and “not come on”.

The attributes and their corresponding values can be classified into two categories, i.e. conditions set and conclusions set. The attributes and values in the conditions set are used to express the IF clauses in rule generation. By the same token, the attributes and values in the conclusions set are used to represent the THEN clauses in rule generation. Since the objective of this case is to diagnose possible problems with respect to the components of an automotive system, the attributes in the conclusions set can be identified as “problem” and the elements with potential problems can be regarded as corresponding values. Meanwhile, the terms describing the states of the automotive system are used to form the conditions set. Tables 6.2 and 6.3 list the attributes and values in the conditions set and conclusions set, respectively.

Table 6.2 The attributes and values in conditions set

Conditions set	
Attributes	Values
Lights	Come on Not come on
State of Engine	Getting gas Turn over Slack
Fuel tank	Gas existing Empty
Carburetor	Gas existing No gas existing
Gas smell	Not present Present
Result of trying starter	Car cranks normally Car cranks slowly

Table 6.3 The attributes and values in conclusions set

Conclusions set						
Attributes	Problem					
Values	Sparking plugs	Fuel tank	Engine	Battery	Cables	Starter motor

For generating rules, the relationship and linkage of attributes and values between the conditions set and conclusions set need to be identified. According to the relationship estimation procedure presented in Section 6.4.3, the relationships can be quantified in terms of Equation (6-5). Table 6.4 shows the linguistic values of the effects of the knowledge attributes on the system tasks, where W_k adopts numerical values 1-5 to estimate the scaled importance, namely 5=most importance and 1=least importance. According to the experience of domain expert(s), a threshold value $\theta=0.40$ is chose to identify the significant relationships in this case. Table 6.5 illustrates the results of the quantified relationships and the significant relationships identified for the attributes. The marked decimals which are greater than 0.40 indicate the significant relationships existing between each pair of attributes. Since the significant relationships exists between the attributes, the relationships between their corresponding values can also be considered significant. Hence, based on the results shown in Table 6.5, the significant relationships between the values can be determined by knowledge engineers. Table 6.6 shows the significant relationships between the values.

According to Tables 6.5 and 6.6, those attributes and values that have significant relationships between the conditions set and conclusions set will be selected for rule generation. For example, the attributes and values of the combination [Problem, battery] in the conclusions set are compared with the attributes and values in the conditions set to find the corresponding combination, which has significant relationship. As a result, three

attributes “Lights”, “State of engine” and “Result of trying starter” in the conditions set, and their corresponding values are selected. It is common knowledge that the engine and light will not work normally if there is something wrong with battery. Thus, the value “Not come on” is given to the attribute “Lights”, the value “Slack” is given to the attribute “State of engine” and the value “Car cranks slowly” is given to the attribute “Result of trying starter”.

	Diagnose sparkling plug (ST_1)	Diagnose fuel tank (ST_2)	Diagnose engine (ST_3)	Diagnose battery (ST_4)	Diagnose cables (ST_5)	Diagnose starter motor (ST_6)
Importance	2	2	3	1	3	4
Lights (KA_{u1})						
State of engine (KA_{u2})						
Fuel tank (KA_{u3})						
Carburetor (KA_{u4})						
Gas smell (KA_{u5})						
Result of trying starter (KA_{u6})						
Problem (sparkling plug) (KA_{v1})						
Problem (fuel tank) (KA_{v2})						
Problem (engine) (KA_{v3})						
Problem (battery) (KA_{v4})						
Problem (cables) (KA_{v5})						
Problem (starter motor) (KA_{v6})						

Table 6.4 Linguistic values of effects on knowledge attributes on system tasks

	Problem (Sparking plug) (KA_{v1})	Problem (Fuel tank) (KA_{v2})	Problem (Engine) (KA_{v3})	Problem (Battery) (KA_{v4})	Problem (Cables) (KA_{v5})	Problem (Starter motor) (KA_{v6})
Lights(KA_{u1})	0	0.308	0.308	0.438	0.423	0.417
State of engine (KA_{u2})	0.420	0.053	0.158	0.737	0.25	0.475
Fuel tank(KA_{u3})	0.188	0.350	0.450	0.348	0.238	0.225
Carburetor(KA_{u4})	0.295	0.250	0.450	0.300	0.272	0.225
Gas smell(KA_{u5})	0.214	0.500	0.363	0.382	0.329	0
Result of trying starter(KA_{u6})	0.312	0.456	0.375	0.552	0.324	0.225

	Problem (Sparking plug) (KA_{v1})	Problem (Fuel tank) (KA_{v2})	Problem (Engine) (KA_{v3})	Problem (Battery) (KA_{v4})	Problem (Cables) (KA_{v5})	Problem (Starter motor) (KA_{v6})
Lights(KA_{u1})	0	0	0	1	1	1
State of engine(KA_{u2})	1	0	0	1	0	1
Fuel tank(KA_{u3})	0	0	1	0	0	0
Carburetor(KA_{u4})	0	0	1	0	0	0
Gas smell(KA_{u5})	0	1	0	0	0	0
Result of trying starter(KA_{u6})	0	1	0	1	0	0

Table 6.5 Quantified relationships and significant relationships for attributes

	Sparkling plug(KV_{v1})	Fuel tank(KV_{v2})	Engine (KV_{v3})	Battery (KV_{v4})	Cables (KV_{v5})	Starter motor(KV_{v6})
(Lights) Come on(KV_{u1})	0	0	0	0	0	1
(Lights) Not come on(KV_{u2})	0	0	0	1	1	0
(State of engine) Getting gas(KV_{u3})	1	0	0	0	0	0
(State of engine) Turn over(KV_{u4})	1	0	0	0	0	0
(State of engine) Slack(KV_{u5})	0	0	0	1	0	1
(Fuel tank) Gas existing(KV_{u6})	0	0	1	0	0	0
(Fuel tank) Empty(KV_{u7})	0	0	0	0	0	0
(Carburetor) Gas existing(KV_{u8})	0	0	1	0	0	0
(Gas smell) Not present(KV_{u9})	0	1	0	0	0	0
(Starter) Car cranks normally(KV_{u10})	0	1	0	0	0	0
(Starter) Car cranks slowly(KV_{u11})	0	0	0	1	0	0

Table 6.6 The significant relationships between values

Step 4. Generation of sub-clauses of IF and THEN with certainty factors. After the relationships of the attributes and values are identified, the sub-clauses of IF and THEN can be expressed using the combinations of [Attributes, values]. Based on Equations (6-7) and (6-8), all the combinations of attributes and values for each rule are listed in Table 6.7. The certainty factor associated with each combination is given in a parenthesis next to it.

Table 6.7 Combinations of attributes and values for each rule

Rule No	IF	THEN
1	[State of engine, slack] (0.95) [Result of trying starter, car cranks slowly] (0.90) [Lights, not come on] (1.0)	[Problem, battery] (0.80)
2	[State of engine, getting gas] (0.90) [State of engine, turn over] (1.0)	[Problem, sparking plugs] (0.75)
3	[State of engine, slack] (0.95) [Lights, come on] (1.0)	[Problem, starter motor] (0.85)
4	[Lights, not come on] (1.0)	[Problem, cables] (0.70)
5	[Result of trying starter, car cranks normally] (0.87) [Gas smell, not present] (0.90)	[Problem, Fuel tank] (0.85)
6	[Fuel tank, gas existing] (0.95) [Carburetor, gas existing] (0.90)	[Problem, Engine] (0.94)

Subsequently, the IF and THEN clauses are generated via combining the attributes and values in Table 6.7. Continue with the example [Problem, battery], the sub-clauses of IF and THEN can be expressed as:

IF clause: [State of engine, slack] (0.95),

[Result of trying starter, car cranks slowly] (0.90),

[Lights, not come on] (1.0),

THEN clause: [Problem, battery] (0.80)

The above combinations may also be expressed as the following:

IF clause: 'State of engine' = 'slack' (0.95),

'Result of trying starter' = 'car cranks slowly' (0.90),

'Lights' = 'not come on' (1.0)

THEN clause: 'Problem' = 'battery' (0.80)

Step 5. Identification of logical relationships between sub-clauses. To complete IF and THEN clauses, the logical relationships among the sub-clauses need to be identified.

With the same example as in Step 4 and suppose it is Rule 1, in the IF clauses,

Let A_{x11} = “ ‘State of engine’ = ‘slack’ ”,

A_{x12} = “ ‘Result of trying starter’ = ‘car cranks slowly’ ”,

A_{x13} = “ ‘Lights’ = ‘not come on’ ”.

In the THEN clauses, B_Y is expressed as:

B_Y = “ ‘Problem’ = ‘battery’ ”

Based on Figure 6.6, the logical relationships among the sub-clauses of Rule 1 can be expressed as shown in Figure 6.9.

Figure 6.9 Logical relationship for IF portion of Rule 1

Step 6. Rule generation. The IF clauses and THEN clauses in Rule 1 can be expressed as:

IF A_{x11} (0.95) or A_{x12} (0.90) and A_{x13} (1.0)

THEN B_Y (0.80) (CF_{r1})

According to Equations (6-11), (6-12), (6-13) and (6-14), the CF_{r1} can be derived as follows.

$$CF_{r1} = \text{MIN} [\text{MAX} (0.95, 0.90), 1.0] * 0.80 = 0.76$$

Thus, Rule 1 is expressed as:

IF 'State of engine' = 'slack' or 'Result of trying starter' = 'car cranks slowly' and 'Lights' = 'not come on'
THEN 'Problem' = 'battery' ($CF_{r1} = 0.76$)

By the same token, six rules can be generated as listed in Table 6.5.

Table 6.8 All the rules generated in the case study

Rule 1:	IF 'State of engine' = 'slack' or 'Result of trying the starter' = 'car cranks slowly' and 'Lights' = 'not come on' THEN 'Problem' = 'battery' ($CF_{r1} = 0.76$)
Rule 2:	IF 'State of engine' = 'getting gas', and 'State of engine' = 'turn over', THEN 'Problem' = 'sparking plugs' ($CF_{r2} = 0.68$)
Rule 3:	IF 'State of engine' = 'slack', and 'Lights' = 'come on' THEN 'Problem' = 'starter motor' ($CF_{r3} = 0.81$)
Rule 4:	IF 'Lights' = 'not come on' THEN 'Problem' = 'cables' ($CF_{r4} = 0.70$)
Rule 5:	IF 'Result of trying starter' = 'car cranks normally' and 'Gas smell' = 'not present' THEN 'Problem' = 'Fuel tank' ($CF_{r5} = 0.74$)
Rule 6:	IF 'Fuel tank' = 'gas existing', and 'Carburetor' = 'gas existing' THEN 'Problem' = 'Engine' ($CF_{r6} = 0.85$)

The rule set shown in Table 6.8 has been tested using an expert system shell Exsys[®] Professional Version 4.0.0 under Microsoft Windows XP Professional operating system on a PC. Both backward and forward chaining inference processes were tested and the results were identical. This validates that the rule set generated by the proposed MRM approach is complete and consistent. Figure 6.10 shows a screenshot of the testing results. The constructed knowledge-based system has also been verified by sixteen experienced automobile drivers to see whether the results conformed to what was expected. In addition, a number of research students in mechanical engineering were invited to test the usability of the knowledge-based system. From the case study, the

proposed MRM approach appears to be an effective tool for facilitating knowledge acquisition and rule generation in constructing a knowledge-based system.

Compared with other rule induction methodologies, such as ID3, LERS, etc., the proposed MRM approach may possess the following advantages:

- (1) It is an integrated and easy-to-use process for knowledge acquisition.
- (2) It adopts a well-organized knowledge representation scheme and hence makes rule extraction easier and more effective.
- (3) A relationship estimation procedure is adopted to address the quantitative characteristics of information as well as the relationship strength between various knowledge elements, and thus facilitate the rule extraction.

Figure 6.10 A screenshot of the testing results

6.6 Summary

In this chapter, a mathematical model using matrix representation and mapping (MRM) techniques is developed to explore an integral knowledge acquisition process. In particular, the MRM approach for knowledge acquisition is proposed to facilitate the effectiveness of rule extraction in building a knowledge-based system. It begins by identifying the appropriate knowledge sources, followed by acquiring relevant knowledge and organizing the acquired knowledge in a taxonomic manner using a general sorting process. Based on the taxonomic trees generated, a series of matrix representation and mapping operations are performed to extract the rules. The activities and technological orientations involved in this approach are elaborated. These include:

(1) A well-established knowledge organization technique, such as general sorting, was used for the elicitation of the design knowledge from the knowledge sources, such as domain experts, historical records or related literature. The elicited knowledge is preliminarily classified and organized into a taxonomic tree structure, which will make the rule extraction much easier and more effective.

(2) A mathematical matrix representation and mapping process between the conditions set and conclusions set, which are composed of the knowledge attributes and values extracted from the taxonomic trees, was proposed.

(3) A relationship estimation procedure to quantify the relationships between the knowledge attributes/values was described. As a result, those attributes and values in the conditions set and conclusions set that have significant relationships can be selected to generate the sub-clauses of rules.

(4) An uncertainty handling mechanism that incorporates certainty factor formalism was presented. It is integrated into the MRM approach to deal with uncertain and imprecise information involved in the knowledge-based systems.

(5) A case study on primarily diagnosing automotive systems was used to illustrate how the proposed approach works. The result of the case study is rational and reveals the potential of the proposed MRM approach. As such, based on the proposed MRM approach, it would be worthwhile to develop and explore a full-scale, computerized process that facilitates and improves the efficiency of knowledge acquisition.

CHAPTER 7 CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This thesis investigates the strategies and implementation techniques of design evolution and knowledge handling for product conceptualization.

As already mentioned, product conceptualization plays an extremely important role in new product development (NPD). Many researchers have recognized the fact that product conceptualization stage deserves much more attention in a successful NPD process. Accordingly, much work has been contributed to it. However, from literature review, it appears that the issues such as design evolution and knowledge handling in product conceptualization have not been well addressed to date.

It has been recognized that a number of gaps exist in design evolution and knowledge handling for product conceptualization as follows:

- (1) *Information reuse in design evolution for product conceptualization.* Based on the literature review in Chapter 2, it has been established that design reuse is of crucial importance for companies to gain competitive advantages in NPD. Although some approaches or models have been developed to deal with this problem, design reuse still suffers many difficulties such as indexing, retrieval and adaptation of prior design knowledge. The potential of successful design reuse has not been fully exploited and limitations still exist in previous research.
- (2) *Indexing and Retrieval mechanism in CBR.* Among the various methodologies for design reuse, it has been identified that case-base reasoning (CBR) is an excellent one to deal with new design problem through reusing past design experiences. It has been well received that the most critical issues in a CBR

process are case indexing and retrieval. To deal with a complex design problem, an effective indexing mechanism and an efficient subsequent case retrieval process are extremely important for a successful CBR-based design system. However, the topics of case indexing and retrieval have not been well addressed in the literature.

(3) *Product Innovation in New Product Concept Development.* As aforementioned, product innovation is an essential factor for organizations to survive in today's competitive business environment. While CBR is promising in solving new design problems via effectively reusing historical design solutions, it is still plagued with difficulties in design innovation for new product concept development. However, few of previous research consider the innovation aspect in CBR-based design systems.

(4) *Knowledge Acquisition Bottleneck.* In order to effectively acquire the corresponding knowledge from various knowledge sources to build a knowledge-based system for design evaluation, an efficient knowledge acquisition process is highly desirable. However, it is well known that knowledge acquisition is a bottleneck that impedes the development of a knowledge-based system. As reported in Chapter 2, the approaches or models reported in the literature still have limitations in facilitating the process of knowledge acquisition.

As such, this work is devoted to bridge these gaps via proposing methodologies to facilitate the design evolution and knowledge handling for product conceptualization. Accordingly, a feature-based design evolution approach, a TRIZ enhanced innovative adaptation procedure and a matrix representation and mapping (MRM) approach for knowledge acquisition are proposed.

The contributions stemmed from this work (Rao and Chen 2005a, 2005b, 2005c, 2006) include:

1. *A feature-based design evolution approach via case-based reasoning.* Basically, this approach is proposed to facilitate the conceptual design evolution for product conceptualization in new product development. In order to effectively utilize the historical design experiences, the CBR methodology is employed and adapted in this work. Within a CBR process, due to the complex interrelationships between various design features, extra difficulties and inefficiency for case retrieval would be caused. As such, a feature relationship decomposition (FRD) procedure using graph decomposition method is proposed to separate the interrelated feature indexing structure in establishing a well-defined indexing mechanism. By decomposing the interrelationships between various features, FRD provides the indexing scheme with more flexibility for case retrieval and retention. Subsequently, AHP (analytical hierarchy process) is adopted to determine the feature priorities. Using the proposed FRD procedure, the number of pair-wise comparisons in AHP can be significantly reduced.

Furthermore, as already mentioned, to reduce the efforts in adapting the unsuitable retrieved attribute cases, a feature-based case retrieval process is proposed to select the most appropriate attribute cases corresponding to each design feature. The proposed process provides a new avenue for case retrieval focusing on selecting individual case attributes for each design feature. It enables the potential design solutions for a specific design problem to be generated by combining those selected appropriate cases. Consequently, the extra adaptation efforts, which is a challenging problem in CBR, can be saved. The capability of the proposed approach has been demonstrated using a case study on wood golf

club design. The result of the case study is analyzed and it reveals the effectiveness of the proposed approach by comparing with conventional CBR methodologies. It has been found that the proposed feature-based design evolution approach is rational and effective to facilitate product concept development.

2. *A TRIZ enhanced innovative adaptation procedure.* This procedure, so-called IAP, has been established to enhance the proposed CBR-based design systems with the capability of product innovation in new product development. It is concentrated on the adaptation stage of case-based reasoning. In order to deal with the contradictions between the new design requirements and derived potential solutions, the TRIZ methodology, which contains a set of powerful innovative problem solving tools, is incorporated into the IAP procedure. The TRIZ enhanced IAP can help designers come up with innovative solutions in resolving contradictions. Basically, the procedure consists of three major components, viz. conflicts checking, ordinary case reparation and resolve design contradictions with TRIZ.

Initially, the IAP performs as a constraint satisfaction process, in which an ordinary case reparation process might be applied if the derived potential solutions can be easily adapted to satisfy the new design requirements. Otherwise, an innovative solution will be put forward to solve the inconsistency between the potential case and new design specifications, especially when a contradiction exists. With the TRIZ enhanced IAP, it is expected the adaptation function, which is a challenging part of CBR process, can be improved. The IAP equips the CBR system with the capability of suggesting innovative solutions to

resolve the inconsistency between new design specifications with the derived potential solutions from the case repository. It is of great help for the CBR-based design evolution approach to generate innovative product concept despite basing on historical design information. A hard disk drive design case is employed as an example in this work to illustrate the procedure of IAP. It appears that the proposed IAP is able to put forward innovative solutions to handle the design contradictions emerged.

3. *A matrix representation and mapping (MRM) approach.* The approach is proposed to facilitate the process of rule extraction in knowledge acquisition. In this work, the rule generation is defined as the mapping process from one domain to another, namely a conditions domain as well as a conclusions domain. As a result, a mathematic model, so-called MRM, that employs the techniques of matrix representation and mapping has been put forward to express the mapping process. The proposed MRM approach for knowledge acquisition begins by identifying the appropriate knowledge sources. The acquired knowledge from various knowledge sources is then preliminarily organized in a taxonomic manner. To facilitate organization of the acquired knowledge into a taxonomic tree structure, a general sorting process is employed. Based on the taxonomic trees generated, a series of matrix representation and mapping operations are performed to extract the rules. A case study on primarily diagnosing automotive systems is used to illustrate how the proposed MRM approach works. The result of the case study is promising and reveals the potential of the proposed MRM approach. Basically, the proposed MRM approach provides a novel avenue for knowledge acquisition. It provides an integrated and complete process for

knowledge acquisition. In addition, it adopts a well-organized knowledge representation scheme and hence makes rule extraction easier and more effective. Based on the proposed MRM approach, it is worthwhile to investigate into a full-scale and computerized process to facilitate the efficiency of knowledge acquisition.

7.2 Limitations

The approaches proposed and implemented in this work present a novel solution to product conceptualization. While this work has made significant contributions in facilitating the process of product conceptualization, it contains some limitations or constraints as follows.

- The application of the proposed CBR-based design system is restricted to physical product design. The possibility of extending the current work to a broader scope, e.g. from physical product design to intangible product design such as service product design, needs to be further explored.
- TRIZ methodology is a tool that can be used to assist designers for product innovation. However, it can not be used to deal with all kinds of design innovation problems. Thus, in the IAP procedure, some aspects of design innovations may be left to the designer's own creativity for innovative adaptation.
- In the MRM knowledge acquisition model, the effectiveness of rule extraction depends largely on the quality of design knowledge elicited from the domain experts. Hence, the experience and competence of the domain experts and/or

knowledge engineers involved play an important role in building knowledge-based systems using the proposed MRM approach.

7.3 Future Work

In response to the limitations summarized in the previous section and to lead the methodologies or approaches proposed in this work to a broaden viewpoint, some future work can be conducted as follows:

- To further verify the robustness and/or sensitivity of the methodologies and approaches presented in this work, more case studies on various product concept development projects are required to be investigated. In addition, to broaden the application scope, more efforts should be placed on how to adapt these methodologies to other product design process, such as software product design. The proposed methodologies can also be applied to deal with the issues in other fields such as knowledge management in product design.
- To further evaluate the proposed CBR-based design system, usability study needs to be conducted when a ready-to-use CBR system has been developed. Furthermore, it would be desirable to validate the proposed approach using means other than examples in such a way that the potential computational savings in general case can be identified.
- In this work, the CBR-based design system is improved using the contradiction resolution principles, which is the core tool of TRIZ methodology. In future work, other TRIZ tools such as substance-field analysis, ideal final result and laws of engineering system evolution can possibly be incorporated into the system to provide supplementary functions. This could lead to a more comprehensive CBR-based innovative design system. Moreover, to further test

the TRIZ-enhanced innovative adaptation procedure, more case studies especially on designing new products should be investigated. It would be possible to solve the new design problem using the proposed approach as well as using the previously developed approach. Consequently, a study on comparing the solutions resulted from the two approaches and the computational efforts needed to generate the solutions can be conducted so as to further improve the proposed approach.

- To further improve the MRM approach proposed for knowledge acquisition, some effort can be devoted to the research on how to computerize and automate the process of knowledge acquisition and rule generation. In addition, sensitivity analysis of the strength of relationships between corresponding attributes and values in the conditions set and conclusions set of the relationship matrices should be carried out to test the reliability of the proposed approach.

References

1. Al-Mashari, M., Zairi, M. and Ginn, D. (2005). Key enablers for the effective implementation of QFD: a critical analysis. *Industrial Management + Data Systems*, 105(9), 1245-1260.
2. Al-Shihabi, T. and Zeid, I. (1998). A design-plan-oriented methodology for applying case-based adaptation to engineering design. (AI EDAM) *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 12(5), 463-478.
3. Altshuller, H. (1990). *And Suddenly the Inventor Appeared*, Worcester, MA: Technical Innovation center.
4. Altshuller, H. (1996). *And Suddenly the Inventor Appeared*, Worcester, MA: Technical Innovation center.
5. Atthirawong, W. (2002). *A Framework for International Location Decision-Making in Manufacturing using the Analytical Hierarchy Process Approach*. PhD Thesis, School of Mechanical, Materials, Manufacturing Engineering and Management, University of Nottingham, UK
6. Ayag, Z. (2005). An integrated approach to evaluating conceptual design alternatives in a new product development environment. *International Journal of Production Research*, 43(4), 687–713.
7. Ball, L.J., Lambell, N.J., Ormerod, T.C., Slavin, S. and Mariani, J.A. (2001). Representing design rationale to support innovative design reuse: A minimalist approach. *Automation in Construction*, 10(6), 663-674.
8. Bariani, P.F., Berti, G.A. and Lucchetta, G. (2004). A combined DFMA and TRIZ approach to the simplification of product structure. *Proceedings of the Institution of*

- Mechanical Engineers, Part B: Journal of Engineering Manufacture, 218(8), 1023-1027.
9. Baxter, M. (1995). Product design: A practical guide to systematic methods of new product development. Chapman & Hall, New York.
 10. Belaziz, M., Bouras, A. and Brun, J.M. (2000). Morphological analysis for product design. CAD Computer Aided Design, 32(5-6), CAD After 2000: Integrated, Intelligent, Collaborative. Part II, 377-388.
 11. Belecheanu, R., Pawar, K.S., Barson, R.J., Bredehorst, B. and Weber, F. (2003). The application of case based reasoning to decision support in new product development. Integrated Manufacturing Systems, 14(1), 36-45.
 12. Benfer, R.A., Brent, E.E.J. and Furbee, L. (1991). Expert Systems. Sage, Newbury Park.
 13. Benfer, R. A. and Furbee, L. (1989). Knowledge acquisition in the Peruvian Andes. AI Expert, 22-29.
 14. Berzal, F., Cubero, J.-C., Marin, N. and Serrano, J.-M. (2001). TBAR: An efficient method for association rule mining in relational databases. Data and Knowledge Engineering, 37(1), 47-64.
 15. Bing, C. J. and Liang, T.-P. (1995). Fuzzy indexing and retrieval in case-based systems. Expert Systems with Applications, 8(1), 135-142.
 16. Bohuslav, B., Darrell, M. and Pavel, J. (1999). Case Studies In TRIZ: A Novel Heat Exchanger (Use of Function Analysis Modelling to Find and Eliminate Contradictions). [Http://www.triz-journal.com/archives/1999/12/](http://www.triz-journal.com/archives/1999/12/), Accessed on 28/May/2006
 17. Bosak, J. (1990). Decomposition of graphs. Norwell, MA: Academic Publishers.

18. Bossert, J. L. (1991). *Quality Function Deployment: A Practitioner's Approach*. Milwaukee, WI: ASQC Quality Press.
19. Boyle, I.M., Rong, K. and Brown, D.C. (2006). CAFixD: a case-based reasoning fixture design method, Framework and indexing mechanisms. *Transactions of the ASME. Journal of Computing and Information Science in Engineering*, 6(1), 40-48.
20. Braha, D. (2001). *Data mining for design and manufacturing, methods and applications*. The Netherlands: Kluwer Academic Publishers.
21. Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984). *Classification and Regression Trees*. Chapman & Hall, New York
22. Cao, Y.D. and Jiang, N. (1998). Blackboard-based multi-agent cooperative expert system tool. *Journal of Beijing Institute of Technology (English Edition)*, 7(4), 406-411.
23. Carayannis, E. and Coleman, J. (2005). Creative system design methodologies: The case of complex technical systems. *Technovation*, 25(8), 831-840.
24. Carlson-skalak, S., White, M. D. and Teng, Y. (1998). Using an evolutionary algorithm for catalog design. *Research in Engineering Design*, 10, 63-83.
25. Carter, C. and Catlett, J. (1987). Assessing Credit card applications using machine learning. *IEEE Expert*, 2(3), 71-79.
26. Cascint, G. and Rissone, P. (2004). Plastics design: integrating TRIZ creativity and semantic knowledge portals. *Journal of Engineering Design*, 15(4), 405-424.
27. Cattral, R., Oppacher, F. and Deugo, D. (2002). Evolutionary data mining with automatic rule generalization. *Recent Advances in Computers, Computing and Communications*, 296-300.
28. Cercone, N., An, A. and Chan, C. (1999). Rule-induction and case-based reasoning: hybrid architectures appear advantageous. *IEEE Transactions on Knowledge and Data Engineering*, 11(1), 164–174.

29. Cestnik, G., Kononenko, I. and Bratko, I. (1987). Assistant 86: A knowledge acquisition tool for sophisticated users. In Bratko, I., & Lavrac, N. (Eds.), *Progress in Machine Learning*. Sigma Press.
30. Chang, H.-T. and Chen, J. L. (2004). The conflict-problem-solving CAD software integrating TRIZ into eco-innovation. *Advances in Engineering Software*, 35(8-9), 553-566.
31. Chau, K.W. and Albermani, F. (2004). Hybrid knowledge representation in a blackboard KBS for liquid retaining structure design. *Engineering Applications of Artificial Intelligence*, 17 (1), 11-18.
32. Chen, C.-H. (1996). Development of a product design evaluation system. Ph.D. Dissertation, University of Missouri-Columbia, Columbia, MO., December.
33. Chen, C.-H., Khoo, L. P. and Jiao, L.M. (2004). Information deduction approach through quality function deployment for the quantification of the dependency between design tasks. *International Journal of Production Research*, 42(21), 4623-4637.
34. Chen, C.-H., Khoo, L. P. and Yan, W. (2002). Web-enabled customer-oriented product concept formation via laddering technique and kohonen association. *Concurrent Engineering Research and Applications*, 10(4), 299-310.
35. Chen, C.-H., Khoo, L. P. and Yan, W. (2003). Evaluation of multicultural factors from elicited customer requirements for new product development. *Research in Engineering Design*, 14(3), 119-130.
36. Chen, C.-H., Khoo, L. P. and Yan, W. (2005). PDCS-a product definition and customisation system for product concept development. *Expert Systems with Applications*, 28(3), 591-602.

37. Chen, C.-H., Khoo, L.P. and Yan, W. (2006). An Investigation into Affective Design Using Sorting Technique and Kohonen Self-Organising Map. *Advances in Engineering Software*, 37(5), 334-349.
38. Chen, C.-H and Occeña, L.G. (1999). Knowledge sorting process for a product design expert system. *Expert Systems*, 16(3), 170-182.
39. Chen C.-H. and Occeña, L.G. (2000). Knowledge decomposition for a product design blackboard expert system. *Artificial Intelligence in Engineering* 14(1), 71-82.
40. Chen, C.-H, Occeña, L.G. and Fok, S. C. (2001). CONDENSE: A concurrent design evaluation system for product design. *International Journal of Production Research*, 39(3), 413-433.
41. Chen, C.-H., Wu, T. and Occeña, L.G. (2000). Linguistic evaluation with blackboard architecture for generating product design specification. *Proceedings of the 5th International Conference on Computer Integrated Manufacturing, ICCIM, Singapore, March, 28-30, pp.1096-1106.*
42. Chen, C.-H, Wu, T. and Occena, L. G. (2002b). Knowledge organization of product design blackboard systems via graph decomposition. *Knowledge-Based Systems*, 15(7), 423-435.
43. Chen, W.C., Tseng, S.-S., Chang, L.-P., Hong, T.-P. and Jiang, M.-F. (2002c). A parallelized indexing method for large-scale case-based reasoning. *Expert Systems with Applications*, 23(2), 95-102.
44. Chin, K. S. and Wong, T.N. (1996). Knowledge-based evaluation for the conceptual design development of injection molding parts. *Engineering Applications of Artificial Intelligence*, 9(4), 359-376.

45. Chougule, R.G. and Ravi, B. (2005). Variant process planning of castings using AHP-based nearest neighbour algorithm for case retrieval. *International Journal of Production Research* 43(6), 1255-1273.
46. Christensen, C. M. (1997). *The innovator's dilemma: when new technologies cause great firms to fail*. Harvard Business School Press.
47. Clark, P. and Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, 3, 261-283.
48. Cohen, E. (1996). Efficient parallel shortest-paths in diagraphs with a separator decomposition. *Journal of Algorithms*, 21, 331-357.
49. Cormican, K. and O'Sullivan, D. (2003). A collaborative knowledge management tool for product innovation management. *International Journal of Technology Management*, 26(1), p53-67.
50. Craig, I.D. (1991). *Formal Specification of Advanced AI Architectures*. Ellis Horwood, Chichester.
51. Cross, N. (2000). *Engineering Design Methods: Strategies for Product Design*, third edition. John Wiley & Sons, Ltd.
52. Das, S. K. and Gami, S. (2004). Expert rules for manufacturability analysis of misalignment defects during product design. *International Journal of Computer Integrated Manufacturing*, 17(1), pp. 58 - 68.
53. Dasarathy, B.V. (1991). *Nearest Neighbor Norms: NN Pattern Classification Techniques*. Los Alamitos, CA: IEEE Computer Society Press, 1991.
54. Devdas, S. (2002). *Design for Product Success*. Society of Manufacturing Engineering, Dearborn, Michigan.
55. Dieter, G.E. (2000). *Engineering Design: A Materials and Processing Approach*, 3rd edition. The McGraw-hill companies, Inc.

56. Dixon, J.R. and Duffy, M.R. (1990). The Neglect of Engineering Design. California Management Review, 32(2), pp.9-23.
57. Dubois, D. and Prade, H. (2000). Fundamentals of fuzzy sets. Kluwer Academic Publishers.
58. Duda, R.O., Hart, P.E. and Nilsson, N.J. (1976). Subjective Bayesian methods for rule-based inference systems. AFIPS Conference Proceedings of the 1976 National Computer Conference, vol. 45, pp. 1075 -- 1082.
59. Duffy, S.M., Duffy, A.H.B. and MacCallum, K.J. (1995). A design reuse model. International Conference on Engineering Design ICED'95 Praha, August, Vol.2, pp.490-495.
60. Eckhard, S.-H. and Ahrend, C. (2004). Applying the TRIZ Principles of Technological Evolution to Customer Requirement Based Vehicle Concepts-Experience Report. <http://www.triz-journal.com/archives/2004/03/2004-03-04.pdf>, Accessed on 28/May/2006.
61. French, M.J. (1985). Conceptual Design for Engineers, Design Council, London.
62. Gabow, H.N. (2000). Path-based depth-first search for strong and bi-connected components. Information Processing Letters, 74(3-4), 107-114.
63. Gao, C.Q., Huang K. Z. and Zhang, Y. (2005). Creative conceptual design ideas can be gotten with TRIZ methodology. <http://www.triz-journal.com/archives/2005/06>, Accessed on 28/May/2006.
64. Garza, A. G. D. S. and Maher, L. M. (1999). An Evolutionary Approach to Case Adaptation. Lecture Notes in Computer Science, 1650, p162.
65. Gonzalez, A. J. and Dankel, D.D. (1993). The Engineering of Knowledge-based Systems, Theory and Practice. Prentice-Hall, Inc.

66. Graham, I.J., Case, K. and Wood, R.L. (2001). Genetic algorithms in computer-aided design. *Journal of Materials Processing Technology*, 117(1-2), 216-221.
67. Green, G. (2000). Towards integrated design evaluation: validation of models. *Journal of Engineering design*, 11(2), 121-132.
68. Grzymala-Busse, D. M. and Grzymala-Busse, J. W. (1995). Usefulness of a machine learning approach to knowledge acquisition. *Computational Intelligence*, 11(2), 268-279.
69. Guan, J.W., Guan, Z. and Bell, D.A. (1997). Bayesian probability on Boolean algebras and applications to decision theory. *Information Sciences*, 97(3-4), 267-292.
70. Guan, Z. and Liu, Y. (1993). New decomposition method and partial stability of nonlinear large scale time-delay systems. *Control Theory and Applications*, 10(4), 392-398.
71. Guha, S., Rastogi, R. and Shim, K. (2001). CURE: An efficient clustering algorithm for large databases. *Information Systems* 26(1), 35-58.
72. Han, Y.-H. and Lee, K.W. (2006). A case-based framework for reuse of previous design concepts in conceptual synthesis of mechanisms. *Computers in Industry*, 57(4), 305-318.
73. Hsiao, S.-W. (1999). Integrated concurrent engineering based approach for electric-fan design. *Integrated Computer-Aided Engineering*, 6(3), 171-187.
74. Hsiao, S.-W. and Chou, J.-R. (2004). A creativity-based design process for innovative product design. *International Journal of Industrial Ergonomics*, 34, 421-443.
75. Huang, G.Q. and Mak, K.L. (1999). Web-based morphological charts for concept design in collaborative product development. *Journal of Intelligent Manufacturing*, 10, 267-278.

76. Huang, Y.M. (1999). On the general evaluation of customer requirements during conceptual design. *Journal of Mechanism Design, Transactions of the ASME*, 121, 92-97.
77. Hu, Y.-C. (2006). Determining membership functions and minimum fuzzy support in finding fuzzy association rules for classification problems. *Knowledge-Based Systems*, 19(1), 57-66.
78. Ignizio, J.P. (1991). *Introduction to Expert Systems-The Development and Implementation of Rule-based Expert systems*. McGraw-Hill, New York.
79. Ishii, K. and Barkan, P. (1987). Design Compatibility Analysis—a framework for expert systems in mechanical system design. *ASME Computers in Engineering* 1987. Vol. 1. LOC 82-72306. pp. 95-102.
80. Ishii, K. and Eubanks, F. (1993). Life-cycle evaluation of mechanical systems. In: *Proceedings of the 1993 NSF Design and manufacturing systems conference*, Vol.1, pp.575-579.
81. Ishii, K. (1996). Life-cycle Design, in *Mechanical Design: Theory & Methodology*. Springer-Verlag New York, Inc.
82. Islam, M.N. (2004). A methodology for extracting dimensional requirements for a product from customer needs. *International Journal of Advanced Manufacturing Technology*, 23(7-8), 489-494.
83. Jared, G.E.M., Limage, M.G., Sherrin, I.J. and Swift, K.G.(1994). Geometric reasoning and design for manufacture. *Computer-Aided Design*, 26(7), 528-536.
84. Jiang, Y.C., Zhang, S.Y. and Zhong, Y.P. (2005). Constructing agents blackboard communication architecture based on graph theory. *Computer Standards & Interfaces*, 27(3), 285-301.

85. Jiao, L.M. (2004). Design project management for new product development in a global collaborated environment. PhD Thesis, Nanyang Technological University, Singapore.
86. Kantardzic, M. (2003). Data mining: concepts, models, methods, and algorithms. IEEE Press, Wiley-Interscience.
87. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R. and Wu, A. Y. (2002). An efficient k-means clustering algorithms: Analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7), 881-892.
88. Kao, H.-P., Su, E. and Wang, B. (2002). I²QFD: A blackboard-based multi-agent system for supporting concurrent engineering projects. International Journal of Production Research, 40(5), 1235-1262.
89. Kessler, E. H. (2000). Lightening the belt: methods for reducing development costs associated with new product innovation. J. Eng. Technol. Manage, 17, 59–92.
90. Khoo, L.P. and Zhai, L.Y. (2001). A prototype genetic algorithm-enhanced rough set-based rule induction system. Computers in Industry, 46(1), 95-106.
91. Kobayashi, H. (2006). A systematic approach to eco-innovative product design based on life cycle planning. Advanced Engineering Informatics, 20(2), 113-125.
92. Kohonen, T. (2001). Self-organizing maps, 3rd Edition. New York: Springer-Verlag Berlin Heidelberg.
93. Kolodner, J.L. (1993). Case-based Reasoning, Morgan Kaufmann Publishers, Inc.
94. Korf, R.E. (1993). Linear-space best-first search. Artificial Intelligence, 62(1), 41-78.
95. Kraslawski, A. and Kudra, T. (2001). Case-based reasoning for design of drying equipment. Drying Technology, 19(5), 767-783.

96. Kusiak, A., Szczerbicki, E. and Park, K. (1991). A novel approach to decomposition of design specifications and search for solutions. *International Journal of Production Research*, 29(7), 1391-1406.
97. Kwong, C.K., Smith, G.F. and Lau, W.S. (1997). Blackboard-based approach to concurrent process design of injection moulding. *Journal of Materials Processing Technology*, 70(1-3), 258-263.
98. Lai, H.-H, Lin, Y.-C and Yeh, C.-H. (2005). Form design of product image using grey relational analysis and neural network models. *Computers & Operations Research*, 32(10), 2689-2711.
99. Lander, S. E. (1997). Issues in multiagent design systems. *IEEE Expert*, 12(2), 18-26.
100. Leake, D.B. (1996). *Case-Based Reasoning: Experiences, Lessons, & Future Directions*. AAAI Press/The MIT Press, Boston.
101. Lee, C.K.M., Lau, H.C.W. and Yu, K.M. (2005). An object-based knowledge integration system for product development: A case study. *Journal of Manufacturing Technology Management*, 16(2), 156-177.
102. Leeuwen, J. V. (1990). "Graph algorithms." in: Leeuwen J. van (Ed.), *Handbook of Theoretical Computer Science, Vol. A: Algorithms and Complexity*, Elsevier. The MIT Press, Amsterdam, pp. 527-631.
103. Lee, Y.-C., Hong, T.-P. and Lin, W.-Y. (2005). Mining association rules with multiple minimum supports using maximum constraints. *International Journal of Approximate Reasoning*, 40(1-2), 44-54.
104. Legardeur, J., Boujut, J.F. and Tiger, H. (2003). ID²: a new tool to foster innovation during the early phases of design projects. *Concurrent Engineering: Research and Applications*, 11(3), 235-244.

105. Levitt, R.E., Jin, Y. and Dym, C.L. (1991). Knowledge-based support for management of concurrent, multi-disciplinary design. *AI EDAM*, 5(2), 77-95.
106. Liang, G.-S. and Wang, M.-J. (1993). A Fuzzy Multi-Criteria Decision-Making Approach for Robot Selection. *Robotic & Computer-Integrated Manufacturing*, 10(4), 267-274.
107. Lian, W., Cheung, D. W. and Yiu, S.M. (2005). An efficient algorithm for finding dense regions for mining quantitative association rules. *Computers and Mathematics with Applications*, 50(3-4), 471-490.
108. Liao, T.W., Zhang, Z. M. and Mount, C.R. (1998). Similarity measures for retrieval in case-based reasoning systems. *Applied Artificial Intelligence*, 12(4), 267-288.
109. Liu, H. and Tang, M.X. (2006). Evolutionary design in a multi-agent design environment. *Applied Soft Computing*, 6(2), 207-220.
110. Li, Y.J., N, P., Wang, X. S. and Jajodia, S. (2003). Discovering calendar-based temporal association rules. *Data and Knowledge Engineering*, 44(2), 193-218.
111. Madhusudan, T., Zhao, J.L. and Marshall, B. (2004). A case-based reasoning framework for workflow model management. *Data & Knowledge Engineering*, 50(1), 87-115.
112. Maher, M.L., Balachandran, M.B. and Zhang, D.M. (1995). Case-based reasoning in design. Lawrence Erlbaum Associates, Inc., Mahwah, New Jersey.
113. Maher, L. M. and Pu, P. (1997). Issues and Applications of Case-based Reasoning in Design. Lawrence Erlbaum Associates, Inc.
114. Maltby, R.D. (1982). Golf club design, fitting, alteration, and repair. Newark, OH: Malby Enterprises

- 115.Maltby, R.D. (1995). Golf club design, fitting, alteration, and repair: the principles and procedures, Newark, OH: R. Maltby Enterprises: Book trade distribution by R.Maltby Enterprises.
- 116.Mangiameli, P., Chen, S.K. and West, D. (1996). A comparison of SOM neural network and hierarchical clustering methods. *European Journal of Operational Research*, 93(2), 402-417.
- 117.Mani, N. (1994). Towards modelling a multi-agent knowledge system in engineering design. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 1, 160-164.
- 118.McCreary, C.L. Chapman, R.O, and Shieh, F.-S. (1998). Using graph parsing for automatic graph drawing. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 28(5), 545-561.
- 119.McManus, J. W. and Bynum, W. L. (1996). Design and analysis techniques for concurrent blackboard systems. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, 26(6), 669-680.
- 120.Michalski, R. S., Mozetic, I., Hong, J. and Lavrac, N. (1986). The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. *Proc. 5th Nat. Conf. Artificial Intelligence*, pp. 1041-1045.
- 121.Mingers, J. (1987). Rule induction with statistical data- a comparison with multiple regression. *Journal of the Operational Research Society*, 38(4), 347-351.
- 122.Moulianitis, V.C., Aspragathos, N.A. and Dentsoras, A.J. (2004). A model for concept evaluation in design--an application to mechatronics design of robot grippers. *Mechatronics*, 14, 599 – 622.
- 123.Nilsson, N.J. (1980). *Principles of Artificial Intelligence*, Palo Alto, CA: Tioga Publishing company.

124. Norris, K.W. (1963). The morphological approach to engineering design. Conference on Design Methods, Pergamon, Oxford.
125. Ong, S. K. and Guo, D. O. (2004). Online design reuse tool for the support of the generation, embodiment and detailed design of products. *International Journal of Production Research*, 42(16), 3301-3331.
126. Otto, K. N. and Wood, K. L. (1998). Product evolution: A reverse engineering and redesign methodology. *Research in Engineering Design - Theory, Applications, and Concurrent Engineering*, 10(4), 226-243.
127. Otto, K. N. and Wood, K. L. (2001). *Product design: Techniques in Reverse Engineering and New Product Development*. Prentice Hall, Prentice-Hall, Inc, New Jersey, USA.
128. Owen, C.L. (1970). An algorithm for the decomposition of non-directed graphs. In: *Emerging methods in environmental design and planning* (GT, M., editor), pp. 133-146. Cambridge, MA: MIT Press.
129. Owen, C.L. (1986). Structured planning: a computer-supported process for the development of design concepts. In *Proceedings of the 1986 international symposium and workshop on Industrial Design, Section 2*, Tainan, Taiwan, pp. 1–19.
130. Pahl, G and Beitz, W. (1996). *Engineering Design: A Systematic Approach* Second Edition Springer-Verlag London Limited.
131. Pan, D., Zheng, Q.-L., Zeng, A. and Hu, J.-S. (2002). A novel self-optimizing approach for knowledge acquisition. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, 32(4), 505-514.
132. Park, J.-H. and Seo, K.-K. (2006). A knowledge-based approximate life cycle assessment system for evaluating environmental impacts of product design

- alternatives in a collaborative design environment. *Advanced Engineering Informatics*, 20(2), 147-154.
133. Prasad, B. (1994). Competitiveness Analysis of Early Production Introduction and Technology Insertion, PED-Vol.68-1, *Manufacturing Science and Engineering*, Volume I, ASME: pp.121-134. Proceedings of the 1994 Int'l Mechanical Engineering Congress and Exposition, Nov., Chicago, IL
134. Prerau, D.S. (1987). Knowledge acquisition in expert system development. *AI Magazine*, 8(2), 43-51.
135. Pugh, S. (1990). *Total Design*. Addison-Wesley Publishers INC., New York.
136. Pugh, S. (1996). *Creating innovative products using total design: the living legacy of Stuart Pugh*. Reading, Mass, Addison-Wesley Pub.Co.
137. Qin, X.L. and Regli, W. C. (2003). A study in applying case-based reasoning to engineering design: Mechanical bearing design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, 17(3), 235-252.
138. Quinlan, J.R. (1983). Learning efficient classification procedures and their application to chess end games, in *Machine Learning: An Artificial Intelligence Approach*, R.S. Michalski (ed), Palo, CA: Tioga Press.
139. Quinlan, J.R. (1988). Decision trees and multivalued attributes. In J.E. Hayes & D. Michie (eds), *Machine Intelligence*, Vol. 11, London: Oxford University Press, 305-318.
140. Quinlan, J.R. (1992). *C4.5: Programs for Machine learning*. San Mateo, CA: Morgan Kaufmann.
141. Rainey, D. (2005). *Product innovation: leading change through integrated product development*. Cambridge University Press, New York.

- 142.Rao, Z.M. and Chen, C.-H. (2005a). A Feature-based CBR Approach for Product Concept Development. Proc. The 2005 Pan-Pacific Vocational Design Conf. – Design Nexus, pp. 31-40, Yunlin, Taiwan, October.
- 143.Rao, Z.M. and Chen, C.-H. (2005b). A CBR-based Design Evolution Approach for New Product Development. Proc. The Int. Design Congress – IASDR 2005 (2005idc), 10p, Yunlin, Taiwan, November.
- 144.Rao, Z.M. and Chen, C.-H. (2005c). A matrix representation and mapping approach to knowledge acquisition for product design. Lecture Notes in Computer Science, Vol. 3681, 311-317.
- 145.Rao, Z.M. and Chen, C.-H. (2006). A feature-based case retrieval approach for conceptual design evolution. The 36th International Conference on Computers and Industrial Engineering: Challenges for Advanced Technology Industries, 2989-2997, June 20-23, Taipei, Taiwan.
- 146.Readle, J.C. and Henry, R.M. (1998). Intelligent on-line system identification using a blackboard system. Engineering Applications of Artificial Intelligence, 11(6), 759-770.
- 147.Riesbeck, C. and Schank, R. (1989). Inside case-based reasoning. Hillsdale, NJ: Erlbaum
- 148.Roh, T.H., Oh, K.J. and Han, I. (2003). The collaborative filtering recommendation based on SOM cluster-indexing CBR. Expert Systems with Applications, 25(3), 413-423.
- 149.Rosenman, M.A. (1990). Applications of expert systems to building design analysis and evaluation. Building and Environment, 25(3), 221-233.
- 150.Rosenman, M. (2000). Case-based evolutionary design. Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM, 14(1), 17-29.

151. Roy, R. and Riedel, J. C.K.H. (1997). Design and innovation in successful product competition. *Technovation*, 17(10), 537-548.
152. Roy, U. and Liao, J.M. (1998). Application of a blackboard framework to a cooperative fixture design system. *Computers in Industry*, 37(1), 67-81.
153. Royzen, Z. (1998). Case Study: TRIZ Solves a Hard Drive Reliability Problem. [Http://www.triz-journal.com/archives/1998/11/a/index.htm](http://www.triz-journal.com/archives/1998/11/a/index.htm). Accessed on 31/May/2006.
154. Rugg, G. and McGeorge, P. (1997). Sorting techniques: a tutorial paper on card sorts, picture sorts, and item sorts. *Expert Systems*, 14(2), 80-92.
155. Rugg, G. and McGeorge, P. (2005). The sorting techniques: a tutorial paper on card sorts, picture sorts and item sorts. *Expert Systems*, 22(3), 94-107.
156. Saaty, T. L. (2000). *Fundamentals of decision making and priority theory with the analytical hierarchy process*. Pittsburgh, PA, USA: RWS Publications.
157. Sarno, E., Kumar, V. and Li, W. (2005). A hybrid methodology for enhancing reliability of large systems in conceptual design and its application to the design of a multiphase flow station. *Research in Engineering Design*, 16(1-2), 27-41.
158. Sato, K. and Owen, C.L. (1981). A prestructuring model for system arrangement problems. *Design Studies*, 2, 67-76.
159. Savransky, S.D. (2000). *Engineering of creativity: Introduction to TRIZ methodology of inventive problem solving*. CRC Press, Washington, D.C.
160. Sen, Z. (1976). The joint prior probability distribution function of parameters in Bayesian estimation. *Bulletin of the Technical University of Istanbul*, 29(1), 102-110
161. Shafer G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.

162. Shao, X., Zhang, G., Li, P. and Chen, Y. (2001). Application of ID3 algorithm in knowledge acquisition for tolerance design. *Journal of Materials Processing Technology*, 117(1-2), 66-74.
163. Sharma, A. (2005). Collaborative product innovation: integrating elements of CPI via PLM framework. *Computer-Aided Design*, 37, 1425-1434.
164. Shortliffe, E.H. (1976). *Computer based medical consultations: MYCIN*. American Elsevier.
165. Singh, P. and Bettig, B. (2004). Port-Compatibility and Connectability-Based Assembly Design. *Journal of Computing and Information Science in Engineering*, 4(3), 197-205.
166. Steadman, S. and Pell, K. M. (1995). Expert systems in engineering design: An application for injection molding of plastic parts. *Journal of Intelligent Manufacturing*, 6(5), 347-353.
167. Stratton, R. and Mann, D. (2003). Systematic innovation and the underlying principles behind TRIZ and TOC. *Journal of Materials Processing Technology*, 139(1-3), 120-126.
168. Suh, M. S., Jhee, W. C., Ko, Y. K. and Lee, A. (1998). A case-based expert system approach for quality design. *Expert Systems with Applications*, 15(2), 181-190.
169. Surko, P. (1989). Tips for knowledge acquisition. *PC AI*, 14-18.
170. Tandon, N., Agrawal, V.P. and Rao, V.V.P. (2006). Vibration monitoring of computer hard disk drives. *Mechanical Systems and Signal Processing*, 20(4), 1008-1013.
171. Tay, F.E.H. and Gu, J. (2003). A methodology for evolutionary product design. *Engineering with Computers*, 19(2-3), 160-173.
172. Thompson, B. and Thompson, W. (1986). Finding rules in data. *Byte*, 149-158.

173. Tian, Y.Q., Thurston, D.L. and Carnahan, J.V. (1994). Incorporating End-User's Attitudes Towards Uncertainty into an Expert System. *Journal of Mechanical Design*, 116(2), 493-500.
174. Tor, S.B., Britton, G.A., Zhang, W.Y. and Deng, Y.-M. (2002). Guiding functional design of mechanical products through rule-based causal behavioural reasoning. *International Journal of Production Research*, 40(3), 667-682.
175. Tor, S.B., Britton, G.A. and Zhang, W.Y. (2005). A knowledge-based blackboard framework for stamping process planning in progressive die design. *International Journal of Advanced Manufacturing Technology*, 26(7-8), 774-783.
176. Troussier, N., Pourroy, F., Tollenaere, M. and Trebucq, B. (1999). Information structuring for use and reuse of mechanical analysis models in engineering design. *Journal of Intelligent Manufacturing*, 10(1), 61-71.
177. Tsai, C.-Y., Chiu, C.-C. and Chen, J.-S. (2005). A case-based reasoning system for PCB defect prediction. *Expert Systems with Applications*, 28 (4), 813-822.
178. Tsay, Y.-J. and Chang-Chien, Y.-W. (2004). An efficient cluster and decomposition algorithm for mining association rules. *Information Sciences*, 160(1-4), 161-171.
179. Tseng, H. -E, Chang, C.-C and Chang, S.-H. (2005). Applying case-based reasoning for product configuration in mass customization environments. *Expert Systems with Applications*, 29(4), 913-925.
180. Tseng, M.M. and Jiao, J.X. (1997). Case-based evolutionary design for mass customization. *Computers & Industrial Engineering*, 33(1-2), 319-323.
181. Ulrich, K.T. and Eppinger, S.D. (1995). *Product Design and Development*. McGraw-Hill, Inc., New York, USA.
182. Ulrich, K.T. and Eppinger, S.D. (2004). *Product Design and Development*, Third Edition. McGraw-Hill, Inc., New York, USA.

183. Vadde, S., Allen, J.K. and Misstree, F. (1992). Catalog Design: Design using Available Assets. ASME design automation conference, D. A. Hoeltzel, ed., Scottsdale, AZ, pp. 345-354.
184. Vadde, S., Allen, J.K. and Misstree, F. (1995). Catalog Design: Selection Using Available Assets. *Engineering Optimization*, 25, 45-64.
185. Vinoid, B, Gevins, J., Baudin, C., Mabogunje, A., Toye, G. and Leifer, L. (1992). An experimental study of design information reuse. Proceedings 4th international conference on design theory and Methodology, ASME, Scottsdale, AZ, pp.140-147.
186. Von, N.J. and Morgenstern, O. (1947). *Theory of Games and Economic Behaviour*. Princeton University Press.
187. Vong, C.M., Leung, T.P. and Wong, P.K. (2002). Case-based reasoning and adaptation in hydraulic production machine design. *Engineering Applications of Artificial Intelligence*, 15(6), 567-585.
188. Wagner, W. P., Otto, J. and Chung, Q. B. (2002). Knowledge acquisition for expert systems in accounting and financial problem domains. *Knowledge-Based Systems*, 15(8), 439-447.
189. Wang, H., Chen, G. L., Lin, Z.Q. and Wang, H.H. (2005). Algorithm of integrating QFD and TRIZ for the innovative design process. *International Journal of Computer Applications in Technology*, 23(1), 41-52.
190. Wang, Y.-X. and Yan, H. –S. (2002). Computerized rules-based regeneration method for conceptual design of mechanisms. *Mechanism and Machine Theory*, 37(9), 833-849.
191. Watson, I. (1997). *Applying case-based reasoning: Techniques for enterprise systems*. California: Morgan Kaufmann.

192. Weiss, M. and Stetter, F. (1992). Hierarchical blackboard architecture for distributed AI systems. *Proceedings of the 4th International Conference on Software Engineering and Knowledge Engineering*, 349-355.
193. Wong, M.L. (1998). An adaptive knowledge-acquisition system using generic genetic programming. *Expert Systems with Applications*, 15(1), 47-58.
194. Wu, C.-H., Kao, S.-C. and Srihari, K. (2000). GDKAT: A goal-driven knowledge acquisition tool for knowledge base development. *Expert Systems*, 17(2), 90-105.
195. Wu, C.-H. and Kao, S.-C. (2002). Induction-based approach to rule generation using membership function. *International Journal of Computer Integrated Manufacturing*, 15(1), 86-96.
196. Xing, H., Huang, S. H. and Shi, J. (2003). Rapid development of knowledge-based systems via integrated knowledge acquisition. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM, New Artificial Intelligence Paradigms for Manufacturing*, 17(3), 221-234.
197. Xu, W.L., Tso, S.K. and Tso, Y. (2000). Reuse of existing design information in the development of new electronic PTC devices via a neural network approach. *IEEE Transactions on Industrial Electronics*, 47(2), 454-467.
198. Yamashina, H., Ito, T. and Kawada, H. (2002). Innovative product development process by integrating QFD and TRIZ. *International Journal of Production Research*, 40(5), 1031-1050.
199. Yan, H.-T. and Lee, H.-S. (2006). Creative Design of a Straightline Intermittent Reciprocator. [Http://www.triz-journal.com/archives/2006/02/](http://www.triz-journal.com/archives/2006/02/), Accessed on 2/June/2006.
200. Yan, W. (2003). A Knowledge handling strategy for customer-oriented product conceptualization. PhD Thesis, Nanyang Technological University, Singapore.

201. Yan, W., Khoo, L. P. and Chen, C.-H. (2005). A QFD-enabled product conceptualization approach via design knowledge hierarchy and RCE neural network. *Knowledge-Based Systems*, 18(6), 279-293.
202. Yan, W., Pritchard, M. C., Chen, C.-H. and Khoo, L. P. (2006). A study of bidding-oriented collaborative product conceptualization. *International Journal of Computer Integrated Manufacturing*, 19(2), 161-177.
203. Zadeh, L.A. (1975). The concept of a Linguistic Variable and its Application to Approximate Reasoning. *Information Science*, 8, 199-249.
204. Zeid, I., Gupta, S. M. and Bardasz, T. (1997). Case-based reasoning approach to planning for disassembly. *Journal of Intelligent Manufacturing*, 8(2), 97-106.
205. Zha, X.F., Du, H.J. and Qiu, J.H. (2001). Knowledge-based approach and system for assembly oriented design, Part I: The approach. *Engineering Applications of Artificial Intelligence*, 14(1), 61-75.
206. Zha, X. F. and Du, H. (2006). Knowledge-intensive collaborative design modeling and support, Part I: Review, distributed models and framework. *Computers in Industry*, 57, 39–55.
207. Zha, X.F., Du, H.J. and Qiu, J.H. (2001). Knowledge-based approach and system for assembly-oriented design, Part II: The system implementation. *Engineering Applications of Artificial Intelligence*, 14(2), 239-254.
208. Zha, X.F., Sriram, R.D. and Lu, W.F. (2004). Evaluation and selection in product design for mass customization: a knowledge decision support approach. (AI EDAM) *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 18(1), 87-109.

209. Zhang, T., Ramakrishnan, R. and Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. SIGMOD Record (ACM Special Interest Group on Management of Data), 25(2), 103-114.
210. Zhang, W.Y., Tor, S.B., Britton, G.A. and Deng, Y.-M. (2001). EFDEX: A knowledge-based expert system for functional design of engineering systems. Engineering with Computers, 17(4), 339-353.
211. Zhang, W.Y., Tor, S.B. and Britton, G.A. (2002). Automated functional design of engineering systems. Journal of Intelligent Manufacturing, 13(2), 119-133.
212. Zhang, W. Z., Wang, G. X., Cheok, B. T. and Nee, A. Y. C. (2003). A functional approach for standard component reuse. International journal of Advanced Manufacturing Technology, 22, 141-149.
213. Zimmermann, H.-J. (2001). Fuzzy set theory--and its applications. Forurth Edition. Kluwer Academic Publishers.

APPENDIX A GRAPH DECOMPOSITION ALGORITHM

As mentioned in Chapter 4, a graph decomposition algorithm is employed to decompose the complicated relationships between the interrelated design features. The graph decomposition algorithm is a modified version of a non-directed graph decomposition algorithm developed by Owen (1970). For clarification, some of the terms used most frequently in the algorithm are defined in the following:

Vertex: a point at which the sides of an angle intersect in a graph.

Link : a relationship between two vertices in a graph. A link weight is used to indicate the importance level of a relationship labeled by a link.

Connection ratio (C. R.): the ratio of actual incident internal links to possible incident internal links for the vertex under consideration. If all the vertices of a graph have a C.R. of 1.00, then the graph is considered a complete graph.

Subgraph : a component graph less than or equal to the original graph and containing only links and vertices from the original graph.

Decomposition: the breakdown of a graph into component subgraphs.

Accordingly, the algorithm comprises thirteen consecutive steps (Chen and Occeña 2000), as shown below:

Step 1. Set the value for C.R. and each kind of relationship. Then according to the relationship between two vertices, assign a weight to all the links (a link is a relationship between design features):

$$W_j^l = C_j + \lambda_j \quad (\text{A-1})$$

where W_j^l is the link weight of the j th link, $j=1, 2, \dots, m$ (m is the total number of links), λ_j is the weight index for link j , $\lambda_j = 0, 1^2, 5, 10$, C_j is the number of three-link circuits in the j th link.

Step 2. Assign a weight to all the vertices:

$$W_i^v = \frac{1}{n_i} \sum_{j=1}^n W_j^l = \frac{1}{n_i} \sum_{j=1}^n (C_j + \lambda_j) \quad (\text{A-2})$$

where W_i^v is the vertex weight of the i th vertex, n_i is the number of links incident to the i th vertex, W_j^l is the link weight of the j th link.

Step 3. Sort vertices in the descending order by weight.

Step 4. Apply a strong-link decomposition pass. Construct sub-graph blocks by considering the vertices in order, under the following two conditions: that a vertex may join any block if (1) its connection ratio in the original uncut graph exceeds a threshold value, and (2) it is connected to any of the vertices present in the block by its strongest link (as determined from the link values of the current graph structure under the cutting passes). If a vertex does not meet the conditions for entry to any of the blocks already created, permit it to start a new sub-graph block. Ignore disconnected vertices.

Step 5. Apply follow-up decomposition passes. Augment already formed sub-graph blocks by reconsidering all vertices in order (including disconnected vertices), and allowing a vertex to join any block in which it meets condition (1) of step 4. Continue passes until no further additions are made in a single pass.

² Wherever a three-link circuit appears in a graph, a weight of 1 is added to each of its component links.

Step 6. Compare the subgraphs found in the present series of decomposition passes with those saved from previous passes. Save all sub-graphs which are not identical to previously saved sub-graphs.

Step 7. Compute the value of the cutting function, k_i , for a possible i th cutting pass.

$$k_i = \max [k_{i-1} + 1, \min (W'_j)] \quad (\text{A-3})$$

where $j = 1, 2, \dots, n$ (with n being the number of weighted links remaining at the beginning of the i th pass), $i = 1, 2, \dots, n-1$, $k_{i-1} = 0$, when $i = 1$ (no cuts have yet been made). Compare k_i with $\max(W'_j)$. If $k_i \geq \max(W'_j)$, go to Step 8; otherwise, apply a cutting pass, cutting all links with weights less than or equal to k_i and return to Step 1.

Step 8. Select the largest previously unselected subgraph (or subgraphs if there is more than one of the same size) from the list compiled in Step 6. Ignore all subgraphs that intersect previously selected subgraphs

Step 9. Remove the selected subgraph (or subgraphs) from the graph and make it a block of the final partition. If the selected sub-graph is larger than previously established sub-graph blocks, then remove the smaller blocks from the partition and restore them to the graph.

Step 10. Examine the remaining vertices in the graph. If only disconnected singleton or doubleton subgraphs remain, go to Step 11. If larger structures are still present, return to Step 1.

Step 11. Use the largest subgraph (or subgraphs if more than one of the same size exists) found in partition as the first block of the non-disjunctive decomposition.

Step 12. From the list of Step 6, select and remove the largest remaining subgraph (or subgraphs) covering previously uncovered vertices. Ignore all subgraphs that are proper subgraphs of those previously selected.

Step 13. Examine the remaining graph structure. If all vertices are covered, the decomposition is complete. Otherwise return to Step 12.

The graph decomposition algorithm described above is a computer-oriented technique. It is extremely tedious and error prone to decompose a graph manually, especially for complex graphs. For efficient and effective decomposition of graphs, an application software, so-called “GraphDec”, to implement the revised Owen’s algorithm was developed (Chen et al. 2002b). Figure A.1 shows the main window of the application program. In the middle of the main window is a relationship table that describes the relationship between each of the vertices in a graph. After decomposition, the resultant subgraphs are listed in a tree structure on the right side of the main window. Different decomposition results will be obtained by adjusting the values of C.R. and link weight. This tool makes further study on the relationships between various values of C.R. and/or link weight and the resultant subgraphs possible.

To decompose this graph by using the application program, several steps should be followed:

Step 1. Start the application program “GraphDec”. The main window of this program appears with an empty table as shown in Figure A.1.

Figure A.1 Main interface of graph decomposition software

Step 2. Click on the first button locating at the upper left of the main window to create a new graph. Follow the program's prompt to enter the item numbers and corresponding item names for the graph. In feature relationship decomposition, each item corresponds to a certain design feature. This is shown in Figure A.2.

Figure A.2 Inputted feature names

Step 3. Once the item names were entered, the program will return to the main window and display a table to be filled. This table only accepts five kinds of values, namely N.R., R, I.R, V.I.R and C.I.R. Each of these values represents No Relationship, Relationship, Important Relationship, Very Important Relationship and Critical Important Relationship, respectively. In the application program, the user can simply click on right mouse button to select one of the five values from a context menu. The completed table will be looked like the one shown in Figure A.3.

The relationship table is symmetric and hence each cell in it has a corresponding cell with the same value. To facilitate the efficiency and avoid mistakes of data entry, once the user changes the value of a cell, the program will update the corresponding cell's value automatically. In addition, the program allows users to save incomplete or complete table to a file for later editing.

	Material	Weight	Volume	Sole Radius
Material	-----	V.I.F	N.R	N.R
Weight	V.I.R	-----	V.I.R	N.R
Volume	N.R	V.I.F	-----	R
Sole Radius	N.R	N.R	R	-----
Face Bulge	N.R	N.R	V.I.R	N.R
Face Roll	N.R	N.R	N.R	N.R
ce Progressi	N.R	N.R	N.R	I.R
Lie	N.R	N.R	N.R	I.R
Loft	N.R	N.R	R	R
Neck Length	N.R	R	I.R	N.R
Flex	C.I.R	R	N.R	N.R
Price	V.I.R	R	R	N.R

Figure A.3 Relationship table for design features

Step 4. As described previously, in the revised Owen's algorithm, two parameter values, connection ratio (C.R.) and link weight, will strongly affect the decomposition result. To customize the parameter values, the users only need to click on the button "Advanced Setting" to trigger a dialog box for setting preferable parameter values. Figure A.4 shows a set of sample parameter configuration values.

Figure A.4 Parameters configuration dialog box

Step 5. To decompose the graph, simply click on the "Launch Decomposition" button, the decomposition result in terms of a sub-graphs set will be displayed as a tree structure as shown on the right window in Figure A.5. As can be seen, two resultant sub-graph sets are obtained, one is disjunctive sub-graph set and the other one is non-disjunctive sub-graph set. It is not allowed for the sub-graphs in the former set to have overlapped vertices, while it is on the contrary for the latter set.

Figure A.5 Graph decomposition with application program

To help users using this application program, an online help is integrated in the program (as shown in Figure A.6).

Figure A.6 Online help

APPENDIX B RESULTS OF CLUSTERING ANALYSIS AND FEATURE-BASED CASE RETRIEVAL

This appendix consists of the screenshots illustrating the results of clustering analysis and case retrieval for the rest of features which are not shown in Chapter 4. These features are lie, price, sole radius, face bulge, roll, face progression, neck length, volume, weight and flex.

B.1 Results of clustering analysis and case retrieval for lie

B.2 Results of clustering analysis and case retrieval for price

B.3 Results of clustering analysis and case retrieval for sole radius

B.4 Results of clustering analysis and case retrieval for face bulge

B.5 Results of clustering analysis and case retrieval for roll

B.6 Results of clustering analysis and case retrieval for face progression

B.7 Results of clustering analysis and case retrieval for neck length

B.8 Results of clustering analysis and case retrieval for volume

B.9 Results of clustering analysis and case retrieval for weight

Notes: Since there are only five generally used ratings for shaft flex, namely Extra Stiff, Stiff, Regular (ReFlex), Senior and Ladies, the number of cases are not sufficient for the clustering analysis. The similarity reasoning for feature flex can be determined according to the Euclidean distance instead.

B.10 Results of case retrieval for flex