A Descriptive Design Methodology to Support Designers

A thesis submitted to the
Cardiff University
for the degree of

Doctor of Philosophy

by

Kok Weng Ng

Manufacturing Engineering Centre

Cardiff University

United Kingdom

September 2011

Abstract

An engineering design methodology helps designers to design in a systematic way. Based on the findings from a literature review, engineering design methodologies can be categorised into three types: prescriptive, descriptive and normative. Most established design methodologies are of the prescriptive type and they are based on step-oriented models.

However, designers in industry are not found to be too keen on using any of these design methodologies. Among the reasons for not adopting these methodologies are that the prescriptive and normative design methodologies were found to be influencing the design strategies and approaches of a designer while the descriptive types were mostly used to study the design process. Though designers have their own design strategies and approaches, they also need design support. The descriptive type will not interfere with the designer's strategies but they do suffer from a lack of structure in supporting designers. The goal of this research is to derive a design methodology framework to support designers without influencing their design approaches and strategies.

A descriptive design methodology framework to support designers is proposed in this research work. This framework was derived based on four aspects: a descriptive type based on a function-oriented model, the types of support facilities that can be provided, identification of critical design factors as design parameters for the framework and lastly, the adaptation of the Ishikawa fishbone diagram to represent the framework.

The novel descriptive design methodology was applied in two case studies: the first with an experienced designer without using any design methods and second, with a novice designer adopting a design approach based on the step-oriented model. The second case study included an additional design tool based on TRIZ to verify the effectiveness of the novel descriptive design methodology working with other tools. The designers' feedback and observations from these both case studies showed that the novel descriptive design methodology was able to support designers in many ways. In particular it was able to accommodate different design approaches and strategies without influencing the designer, providing both methodology-related and computational-platform related support facilities as well as working in a complementary way with other design tools.

Acknowledgements

I would like to express my utmost gratitude to my supervisor Prof. D. T. Pham for his splendid supervision, continuous advice, encouragement and support throughout my work. His support, advice, comments, opinions and profound insights were invaluable. Without his support, I could not have completed my research work. I would also like to convey my gratitude to my co-supervisor Prof. Stefan Dimov for his advice and support throughout my research.

Special thanks to Dr. Wang Xiang Wei, Mr. Min Htet Saw for their efforts and contributions to my case studies and to my wife, Dr. Mei Choo Ang who supported and encouraged me.

I also want to thank the Manufacturing Engineering Centre of the Cardiff University for the facilitation of my research and their staff for their kind help and assistance whenever I was in need.

I would also like to thank Sirim Berhad and the Ministry of Science, Technology and Innovation for their funding and support of my research.

I would also like to thank Prof. Jami Shah of Arizona University, United States of America for his interest in my work as the core framework of his research work on collaborative design in his application of the NSF (National Science Fund) funding.

Finally, I am deeply indebted to my wife, my parents, my sisters, my brother and my children, for their continuous support and understanding. Their continuous love, support and patience throughout my research have contributed to the completion of my study in Cardiff.

Declaration and statements

Declaration

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed...(Kok Weng Ng - Candidate)

Date..............................

Statement 1

This thesis is the result of the candidate's own investigations, except where otherwise stated. A list of references is appended.

Signed...(Kok Weng Ng - Candidate)

Date..............................

Statement 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan and for the title and the synopsis to be available to outside organisations.

Signed.. (Kok Weng Ng - Candidate)

Date..............................

Signed.. (Professor D.T. Pham - Supervisor)

Date..............................

Table of Contents

Abstract ii
Acknowledgements iii
Declaration and statements iv
Table of Contents v
List of Figures x
List of Tables xiv
Abbreviations xvii
Chapter 1 Introduction 1
1.1 Background of Research 1
1.2 Research Motivation 1
1.3 Research Aims and Objectives 2
1.4 Research Methodology 3
1.5 Outline of the Thesis 6
Chapter 2 Engineering Design Methodology 9
2.1 Background of Engineering Design Methodology 9
2.2 Normative Design Models 10
2.2.1 Probability-based Normative Model 11
2.2.2 Utility-based Normative Model – Decision-based Design 12
2.2.3 Axiomatic Design Model 13
2.3 Descriptive Design Models 16
2.3.1 Protocol Study 16
2.3.2 McDonnell Descriptive Design Model for Interpreting Design 18
2.3.3 Question-based Approach 21
2.3.4 Reflective Descriptive Design Model 23
2.3.5 Design Logbook Model 24
2.4 Prescriptive Design Models 26
2.4.1 Step-oriented Model 26

2.4.2 Issue-based Model 29
2.4.3 Function-based Model 32
2.4.4 Design Reflection-based Model 34
2.4.5 Collaborative Design Model 37
2.4.5.1 Methodology for Group Interaction and Co-operation 39
2.4.5.2 Methodology for Conflict Management 40
2.4.5.3 Methodology for Resource Sharing 40
2.5 The Deficiencies and Strengths of the Established Design Methodologies 42
2.5.1 Deficiencies and Strength of Normative Design Models 42
2.5.2 Deficiencies and Strength of Descriptive Design Models 44
2.5.3 Deficiencies and Strength of Prescriptive Design Models 45
2.6 Analysis of Established Design Methodologies Models 49
2.7 Desired Features of Engineering Design Methodology 54
2.8 Summary 60
Chapter 3 Design Support Facilities for a Design Methodology 61
3.1 Overview of Design Support 61
3.2 Design methodology-related support 62
3.3 Computational-Platform-Related Support 66
3.4 Concept Selection Support – Decision Analysis Techniques 67
3.5 Concept Ideation Support 74
3.5.1 Cognitive-based design ideation approach 75
3.5.1.1 TRIZ (Theory of Inventive Problem Solving) 76
3.5.1.2 Brainstorming 84
3.5.1.3 Lateral Thinking 85
3.5.1.4 Morphological Analysis 86
3.5.1.5 Delphi Method 87
3.5.1.6 Synectics 88
3.5.1.7 Creativity Templates 89

3.5.2 Generative design ideation approach 90

3.6 Analysis of Design Support Facilities for a Design Methodology 91

3.7 Summary 96

Chapter 4 A Descriptive Design Methodology to Support Designers – Conceptualisation and Implementation 98

4.1 Overview of the need for a descriptive design methodology to support designers 98

4.2 Conceptualisation and Derivation of a Descriptive Design Methodology to support Designers 99

4.2.1 Factors affecting the common characteristics of design tasks and the aims of a designer 100

4.2.2 The conceptualisation the descriptive design framework to support designers 103

4.2.3 Detailed description of the descriptive design framework to support designers 106

4.3 Development of the Descriptive Design Methodology Tool to support Designers 108

4.3.1 The architecture of the descriptive design methodology tool 109

4.3.2 Design of user interface for designers 110

4.3.2.1 Text Input User interface for Given Functional Requirements (GFRs) 110

4.3.2.2 Text Input User Interface for Introduced Functional Requirements (IFR)s 112

4.3.2.3 Graphical User interface for designers 115

4.3.2.4 Database design 117

4.4 Summary 121

Chapter 5 Case Study 1: Designing A Device/Method to Support Concrete Loading In Between Beams 122

5.1 Overview of Case Study 1 (Phase 1 and Phase 2) 123

5.2 Implementation of the Descriptive Design Methodology in Case Study 1 (Phase 1) 124

5.2.1 The aims of the Case Study 1 (Phase 1) 124

5.2.2 The details of implementation for the Case Study 1 (Phase 1) ... 125

5.3 Results from the Implementation of the Descriptive Design Methodology in Case Study 1 (Phase 1) ... 128

5.3.1 The database information captured in Case Study 1 (Phase 1) ... 129

5.4 Implementation of the Descriptive Design Methodology in Case Study 1 (Phase 2) ... 135

5.4.1 The aims of the Case Study 1 (Phase 2) ... 137

5.4.2 The details of implementation for Case Study 1 (Phase 2) ... 137

5.5 Results from the Implementation of the Descriptive Design Methodology in Case Study 1 (Phase 2) ... 138

5.5.1 The database information captured in Case Study 1 (Phase 2) ... 138

5.6 Results from the Interview with the Designer on the Descriptive Design Methodology in Case Study 1 (Phase 1 and Phase 2) ... 142

5.7 Analysis and Discussions of the Implementation of the Descriptive Design Methodology and the Results Obtained in Case Study 1 (Phase 1 and Phase 2) – Design of concrete filling support device for gap between beam slabs case study ... 143

5.7.1 Analysis and discussion of the results obtained in Case Study 1 (Phase 1) ... 146

5.7.2 Analysis and discussion of the results obtained in Case Study 1 (Phase 2) ... 153

5.8 Summary ... 160

Chapter 6 Case Study 2: Designing A Conceptual End-Effector to Manipulate and Handle Human Segments for a First Aid Robot System (FAROS) ... 161

6.1 Overview of Case Study ... 161

6.2 Implementation of the Descriptive Design Methodology with TRIZ-based Ideation Tool in Case Study 2 ... 164

6.2.1 The aims of Case Study 2 ... 165

6.2.2 The Ideation Tool based on TRIZ ... 166

6.2.3 The Integration between the Descriptive Design Methodology Tool and the TRIZ-based Ideation Tool based on TRIZ ... 170

6.2.4 The details of implementation for Case Study 2 172

6.3 Results from Implementation of the Descriptive Design Methodology Software Tool with TRIZ-based Ideation Tool in Case Study 2..... 180

6.3.1 The database information captured in Case Study 2............. 180

6.3.2 Results from the Interview with the Designer on the Descriptive Design Methodology in Case Study 2.. 195

6.4 Analysis and discussion of the Implementation Design Methodology with TRIZ-based Ideation Tool and the Results Obtained in Case Study 2... 196

6.5 Summary 207

Chapter 7 Conclusions, Contributions and Further Work............................... 208

7.1 Conclusions 208

7.2 Contributions 209

7.3 Future work 215

Appendix 1: Feedback form for the descriptive design software tool............. 218

Appendix 2: Feedback form for the descriptive design software tool............. 219

References 220

List of Figures

Figure 1.1 Action research approach for establishing the descriptive design framework 4

Figure 1.2 The flow of chapter of this thesis 8

Figure 2.1 An overview of the Northrop Grumman Commercial Aircraft Division (NGCAD) Probabilistic Design Model (reproduced from (Long and Narciso 1999)) 12

Figure 2.2 A Decision-based Design Model by Hazelrigg (reproduced from (Hazelrigg 1998)) 14

Figure 2.3 Suh's axiomatic design methodology (reproduced from (Suh 1990)) 14

Figure 2.4 Distinction between coupled, decoupled and uncoupled design 15

Figure 2.5 How protocol study is carried out and a portion of the data collected (reproduced from (Cross et al. 1996)) 18

Figure 2.6 Function Protocol Input (reproduced from (Bender et al. 2002b)).. 19

Figure 2.7 Notation for links in a systemic grammar network (reproduced from (McDonnell 1997)) 19

Figure 2.8 A portion of the system grammar network that represents a design proposal (reproduced from (McDonnell 1997)) 20

Figure 2.9 The question-based model that transform design requirement to design decision via generative design questions and deep reasoning questions (reproduced from (Eris 2004)) 21

Figure 2.10 The conceptual framework of questions-based on Lehnert's taxonomy, 4 of Graesser's taxonomy and an additional 5 categories added by Eris (reproduced from (Eris 2004)) 22

Figure 2.11 An example of how to use generic design questions (reproduced from (Grebici et al. 2009)) 22

Figure 2.12 The reflection of a design process (reproduced from (Reymen et al. 2006)) 24

Figure 2.13 The classification schema elements used in EEL (reproduced from (McAlpine et al. 2009)) 25

Figure 2.14 The drop-down menu tagging and the design notes used in EEL (reproduced from (McAlpine et al. 2009)) 25

Figure 2.15 A Generalised Model of Step-Oriented Prescriptive Design Methodologies 28

Figure 2.16 An example of a Lahti issue-based concept model (reproduced from (Lahti et al. 1996)) 29

Figure 2.17 The rationale structure work plane for the mobile arm support project (reproduced from Bracewell (2004)) 31

Figure 2.18 The functional labels for a cordless screwdriver and a vehicle seat (reproduced from (Hirtz et al. 2002)). 33

Figure 2.19 The basic function groups of taxonomies (reproduced from (Kirschman et al. 1996)) 33

Figure 2.20 The user interface of the functional design tool developed by Kirschman (reproduced from (Kirschman et al. 1996))........................ 34

Figure 2.21 The main interface of the checklist description for design situation (reproduced from (Reymen and Melby 2001))...................... 35

Figure 2.22 The user interface for designers to provide detail information on a particular design factor in a design situation (reproduced from (Reymen and Melby 2001)). 36

Figure 2.23 The text –based user interface of the design relation in which designers have to provide their input for the Reymen's reflection based model (reproduced from (Reymen and Melby 2001)). 36

Figure 2.24 The architecture of IPDE (reproduced from (Urban et al. 1999)) 41

Figure 2.25 The "logical incrementalism" is similar to the root and branch approach which is shown here (Quinn 1980) 54

Figure 3.1 The differences between a descriptive question-based design methodology and a prescriptive question-based design methodology . 64

Figure 3.2 The differences of scope between a design methodology and a decision analysis models. 68

Figure 3.3 The brainstorming method with its characteristics (reproduced from (Orloff 2006)) 85

Figure 3.4 The morphological analysis method with its characteristics (reproduced from (Orloff 2006)) 86

Figure 3.5 The Delphi method flow chart (reproduced from (Slocum 2005))... 87

Figure 3.6 The Synectics method with its characteristics (reproduced from (Orloff 2006))... 88

Figure 3.7 The operators involved in the templates (reproduced from (Goldenberg et al. 1999b) 89

Figure 3.8 The other four templates: component control, replacement, displacement and division templates (reproduced from Goldenberg et. al. (1999b) 90

Figure 3.9 The schematic representation of a generative design ideation system (reproduced from (Eckert et al. 1999))....................................... 91

Figure 4.1 Critical factors affecting the common characteristics of a design task and the aims of a designer... 102

Figure 4.2 The product design process... 103

Figure 4.3 Framework of the Descriptive Design Methodology that allows a designer to design according to his preference and approach 105

Figure 4.4 Architecture of the Descriptive Design Tool 110

Figure 4.5 The text-based interface for "Given functional requirements" 111

Figure 4.6 Text-based interface for "Introduced Functional Requirements" with the link menu launched 114

Figure 4.7 Dialogue box for designers to provide information on reminder for delaying design decisions 115

Figure 4.8 The graphical interface of the descriptive design tool (a zoom-in view) 116

Figure 4.9 The schematic structure of the repository system for the descriptive product design tool 118

Figure 4.10 A screen capture of the database system for the prototype descriptive product design tool for Case Study 1 (Phase 1 and Phase 2) 119

Figure 4.11 A screen capture of the improved database system for the prototype descriptive product design tool for case study 2................ 120

Figure 5.1 The existing method to support concrete loading in between beam slabs with steel bars, steel cables and a wood planks 124

Figure 5.2 The design process of the descriptive design tool............................ 127

Figure 5.3 The CAD model of the device to support the concrete filling between beams.. 134

Figure 5.4 The concrete filling support device in operation (noticed the use of cable tie to pull the wood plank) 135

Figure 5.5 The graphical user interface of the descriptive design prototype software tool when Phase 1 of the case study 1 was completed with a zoom in view 136

Figure 5.6 The improved concrete filling support device................................. 141

Figure 5.7 The initial idea of using a triangular device with two cable ties suggested by the customer.. 149

Figure 5.8 The different orientation of the newly designed concrete loading support device allows it to accommodate different sizes of gaps in between beams (viewed from the top).. 149

Figure 5.9 The snapshot of the graphical user interface of the descriptive design prototype tool when Phase 2 of Case Study 1 was completed with a zoom in view of the corners. 155

Figure 5.10 Design change involved when the number of cable ties used was reduced from two to one.. 156

Figure 5.11 One of the CAE analysis results in the documentation captured by the descriptive design framework 157

Figure 6.1 Difference between the linkages of ideation tool with descriptive design framework.. 164

Figure 6.2 The user interface of ideation tool based on TRIZ that allows assignment of weightage.. 167

Figure 6.3 The demarcation of descriptive design system and the TRIZ-based ideation system within the entire database system.................... 171

Figure 6.4 A snapshot of the modified text-based user interface of GFR for the descriptive design prototype tool that was integrated with the TRIZ-based ideation tool 174

Figure 6.5 A snapshot of the modified text-based user interface of IFR for the descriptive design prototype tool that was integrated with the TRIZ-based ideation tool 175

Figure 6.6 A snapshot of the TRIZ-based ideation tool with the input data completed and the results shown for the IFR 1.7................................ 177

Figure 6.7 A snapshot of the TRIZ-based ideation tool with the input data completed and the results shown for the IFR 3.1................................ 178

Figure 6.8 A snapshot of the TRIZ-based ideation tool with the input data completed and the results shown for the IFR 1.6.1.1.2....................... 179

Figure 6.9 A screen shot of the junction table IFR/LinkIFR (top) and the LinkIFR table (bottom) that stored the information about the links between IFR and IFR or GFR for the design of a conceptual end-effector in Case Study 2.. 185

Figure 6.10 The translation of IFR 1.7 (Need to avoid injuring the patient) to improving and worsening features and the recommended inventive principles from the TRIZ-based ideation tool...................................... 201

Figure 6.11 The translation of IFR 1.6.1.1.1 (Need to determine the preliminary action of an end-effector to deal with possible positions) to improving and worsening features and the recommended inventive principles from the TRIZ-based ideation tool 203

Figure 6.12 The translation of IFR 3.1 (Need to determine the shape of end-effector to take max load) to improving and worsening features and the recommended inventive principles from the TRIZ-based ideation tool 204

Figure 6.13 The snapshot of the graphical user interface of the descriptive design prototype tool when Case study 2 was completed with zoom in views.. 206

List of Tables

Table 2.1 Summary and comparisons of several step-oriented prescriptive design methodology models 30

Table 2.2 Summary of several weaknesses of group decision making (reproduced from (Baron et al. 2003)) 38

Table 2.3 Summary of differences between normative, descriptive and prescriptive design methodologies 56

Table 2.4 Results of analysis on current design methodologies and desired design methodology 59

Table 3.1 Summary of the design methodology-related support facilities provided by normative, descriptive and prescriptive design methodologies 65

Table 3.2 Summary of the concept selection support facilities provided by decision analysis models 69

Table 3.3 Brief description of several decision analysis models 70

Table 3.3 Brief description of several decision analysis models (continued) 71

Table 3.3 Brief description of several decision analysis models (continued) 72

Table 3.3 Brief description of several decision analysis models (continued) 73

Table 3.4 Schematic table of the classical TRIZ contradiction matrix (the numbers in italic are numbers representing inventive principles adapted from Mann (2002)) 77

Table 3.5 The 40 inventive principles of TRIZ (Altshuller 1997) 78

Table 3.6 The 48 improving and worsening features of TRIZ contradiction matrix by Mann (reproduced from (Mann et al. 2003)) 82

Table 3.7 The 50 improving and worsening features of TRIZ contradiction matrix by Mann (reproduced from (Mann 2009)) 83

Table 3.8 A summary of comparisons between different design tools and their support facilities to help the designer. 97

Table 5.1 A screen shot of the data captured in ProductInfo table by the descriptive design prototype software tool 130

Table 5.2 A screen shot of the data captured in ComponentInfo table in by the descriptive design prototype software tool 130

Table 5.3 A screen shot of some data captured in the GFRInfo table by the descriptive design prototype software tool 130

Table 5.4 A screen shot of a section of the IFRInfo table before Phase 1 was completed (noticed the WIP status) 132

Table 5.5 A screen shot of the CADInfo table ... 133

Table 5.6 A screen shot of the PicDetail table ... 133

Table 5.7 A screen shot of the data captured in LinkIFR table of the database ... 133

Table 5.8 A screen shot of the data captured in ParticipatingDesigner table of the database ... 133

Table 5.9 A screen shot of the IFRInfo table with highlighted additional IFRs to improve the device in the Phase 2 of the case study 1 ... 139

Table 5.10 A screen shot of the CADInfo table with an additional CAD file in the Phase 2 of the case study ... 140

Table 5.11 A screen shot of the PicDetail table with an additional doc file of the analysis report in the Phase 2 of the case study ... 140

Table 5.12 A screen shot of the LinkIFR table with added links in Phase 2 ... 140

Table 5.13 A screen shot of the data captured for the thoughts about the changing the design from a device with triangular shape to a device with telescopic features highlighted in IFRInfo table ... 148

Table 6.1 An example of how TRIZ contradiction matrix with trade-offs is used ... 169

Table 6.2 A screen shot of the data captured in the ProductInfo table for the design of FAROS end-effector ... 181

Table 6.3 A screen shot of the data captured in the ComponentInfo table for the design of FAROS end-effector ... 181

Table 6.4 A screen shot of the data captured in the GFRInfo table for the design of FAROS end-effector ... 182

Table 6.5 A screen shot of the data captured in the GFRInfo table for the design of FAROS end-effector with the links to IFRinfo expanded ... 183

Table 6.6 A screen shot of the IFRInfo table for the design of a conceptual end-effector in the Case Study 2 ... 184

Table 6.7 A screen shot of the PicDetail table that stored the information about the documentation for the design of a conceptual end-effector in the Case Study 2 ... 185

Table 6.8 A screen shot of the ParticipatingDesigner table that stored the information about the designer for the design of a conceptual end-effector in the Case Study 2 ... 186

Table 6.9 Sections of the IFRInfo table that indicate the utilisation of the TRIZ-based ideation tool (note the highlighted field "IFRSupportID") to assist the designer for the design of a conceptual end-effector in the Case Study 2 ... 188

Table 6.10 The IFRTrendSupport table that stored the information about the details of the TRIZ-based ideation tool utilisation by designer for the design of a conceptual end-effector in the Case Study 2......... 189

Table 6.11 The IFRTrendSupport/ImpFeaInfo table that stored the information about the improving features selected by designer for the design of a conceptual end-effector in the Case Study 2............... 189

Table 6.12 The IFRTrendSupport/WorseFeaInfo table that stored the information about the worsening features selected by the designer for the design of a conceptual end-effector in the Case Study 2......... 190

Table 6.13 The ImpFeaInfo table that stores all the improving features listed in the contradiction matrix updated by Mann (2003) 191

Table 6.14 The WorseFeaInfo table that stores all the worsening features of the updated TRIZ contradiction matrix by Mann (2003).................. 192

Table 6.15 The IFRTrendSupport/InventivePrincipleSol table stores all the links between IFRProposeSolID and the InventivePrincipleID......... 193

Table 6.16 The InventivePrincipleSol table that stores all 40 inventive principles based on the TRIZ contradiction table by Altshuller (1997).. 194

Table 6.17 The four IFRs that were studied by the designer to see whether their technology can be adapted for the FAROS end-effector 197

Table 6.18 The captured data on how the need to grasp various sizes of human segments decompose into the determination of the average size of human segments for grasping 198

Table 6.19 Another example of the decomposition from the need to grasp various shapes of human segments to the determination of the average shape of human segments for grasping 198

Table 6.20 Summary of analysis of Case Study 2.. 205

Abbreviations

AHP	**–**	**Analytic Hierarchy Process**
CAD	**–**	**Computer-aided Design**
DP	**–**	**Design Parameter**
DRQ	**–**	**Deep Reasoning Question**
EEL	**–**	**Engineering Electronic Logbook**
FAROS	**–**	**First Aid Robot System**
FR	**–**	**Functional Requirement**
GDQ	**–**	**Generative Design Question**
GFR	**–**	**Given Functional Requirement**
IFR	**–**	**Introduced Functional Requirement**
IPDE	**–**	**Integrated Product Data Environment**
NGCAD	**–**	**Northrop Grumman Commercial Aircraft Division**
SMART	**–**	**Simple Multi Attribute Rating Technique**
SMARTER	**–**	**Simple Multi Attribute Rating Technique Extended to Ranking**
STEP	**–**	**Standard for the Exchange of Product Model Data (ISO 10303)**
TOPSIS	**–**	**Technique for Order Preference by Similarity to Ideal Situation**
TRIZ	**–**	**Teoriya Resheniya Izobretatelskikh Zadatch (The theory of inventive problem solving)**
UML	**–**	**Unified Modelling Language**

Chapter 1
Introduction

1.1 Background of Research

In this challenging global market, every organisation has to be competitive to survive. The ability to derive better new products with more and better features is critical to enhancing competitiveness. The design process plays a crucial role in deriving new products with better features. As the design task becomes more challenging, there is a need to design in a systematic and better way to ensure design errors are minimised and design lead time is reduced. The conventional design approach by tinkering and trial-and-error is becoming infeasible in this challenging market. To differentiate this conventional design methodology and a new design methodology that helps designers to design in a systematic way, an engineering design methodology is introduced. An engineering design methodology provides a systematic approach to design a better product. However, for this research work, the engineering design methodology will at times be referred to as design methodology.

1.2 Research Motivation

Engineering design methodology was established about two decades ago and one of the key aims of such a methodology is to support designers to enable the design process to be carried out in a systematic manner. Within these two decades, huge amounts of literature and research work on design methodologies were published. Current established design methodologies are mostly guidelines that provide general guidance and advice on the management of the design process in phases. Some of the design methodologies are derived based on mathematical models and axioms. Such design methodologies have interfered and influenced the designers' own design approaches and strategies that may differ from those recommended by the methodologies.

Design methodologies are proposed to help designers to design better in a systematic manner. However, the support provided by these design methodologies is insufficient and most of the established design methodologies, which are prescriptive types and guidelines, provide little support to the designers. Design support is critical to designers. Experienced designers and novice designers have different support need. Most established design methodologies have little computational support for designers. Hence, there is a need for a design methodology that can support designers better and can work with other existing design tools to help the designer.

1.3 Research Aims and Objectives

The aims of this research were to explore and derive a design methodology that could support designers without influencing their design approach and strategy. Hence, based on the aims of this research within the limitation of the research scope the research question for this thesis is:

> "Can designers be supported by a design methodology that does not influence their design approaches and strategies?"

With the research question defined, the following research objectives were pursued to resolve this research question:

1. To explore, review and compare established design methodologies from the perspective of supporting designers and their influence on the designer's approach and strategy.
2. To identify and evaluate the crucial design support facilities, namely the design methodology-related support and the computational-platform-related support, the concept selection support and the ideation support that can be applied in a design methodology to help designers.
3. To determine and link common characteristics of design tasks with the aims of a designer to determine the critical design parameters for the purpose of deriving a novel design methodology.
4. To propose and derive a novel design methodology that can provide these key design support facilities without influencing the designer's approach and strategy.
5. To integrate an optional design tool based on TRIZ with the novel descriptive design methodology to evaluate its ability to work with established design tools.
6. To assess and verify the effectiveness of the novel design methodology in two case studies.

1.4 Research Methodology

In any successful research work, a research methodology plays a crucial role in providing the platform, enabling the focus and showing the directions to allow a researcher to work systematically throughout the duration of research to achieve the aims of the proposed research. Therefore, it is very important to determine the appropriate research methodology before carrying out any research to ensure the research endeavour will have the highest chance of success.

In this research, the characteristics of the domain of research play a significant role in determining the selection of an appropriate research methodology. In the domain of engineering design research, the difficulties and the complexities confronted in validating any research outcomes are obvious. It is difficult to validate the contributions of any research work in the engineering design domain because of the subjective nature of the research outcomes in this domain (Pedersen et al. 2000). Hence, the research methodology adopted in this research work has to deal with the subjective nature of the research domain. Action-research methodology, a qualitative-based research methodology, was adopted for this research work. The approach of this action-research methodology is shown in Figure 1.1.

.

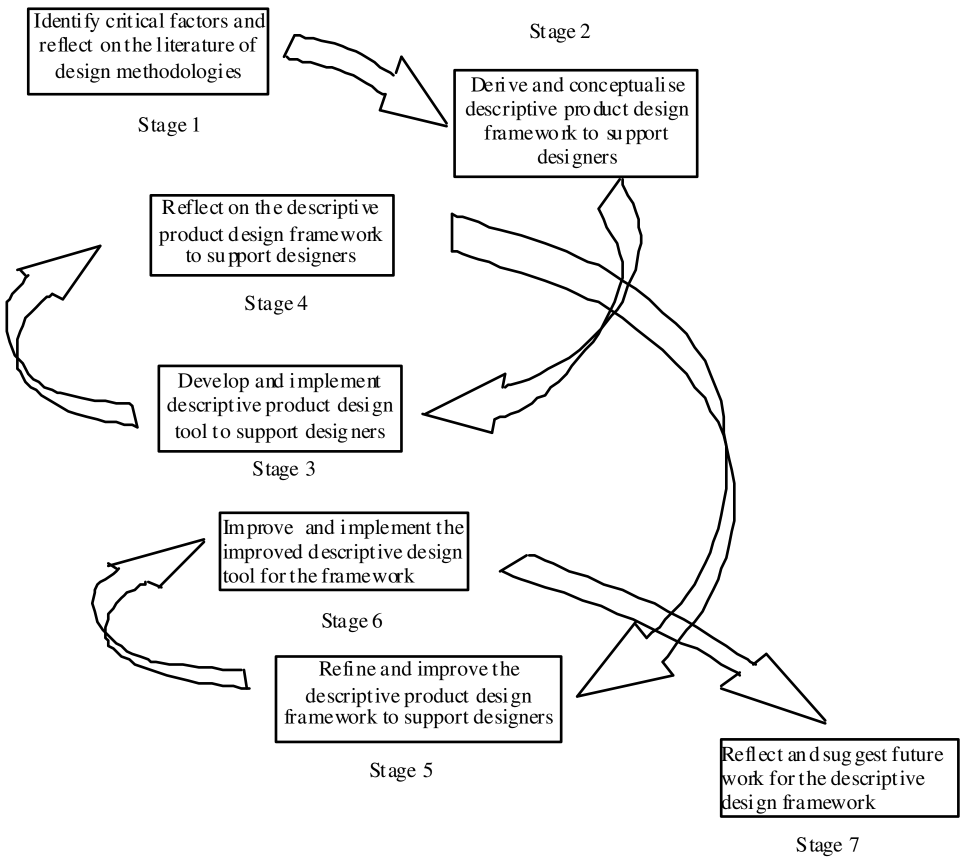


Figure 1.1 Action research approach for establishing the descriptive design framework

One of the reasons for using an action research approach is because it is a participatory approach where the research work carried out involved the participation of subjects, in this case, designers in validating the research outcomes.

The action research approaches may be split into seven stages where the stages are spiralling downwards as the descriptive design framework is refined via feedback from the case study in each spiral. As the action research approach spirals downwards, the case studies also verify and validate the contributions of the proposed framework.

The seven stages of the action research approach are as below:

Stage 1- Identify critical factors and reflect on the literature of design methodologies.

Stage 2 - Derive and conceptualise descriptive product design framework to support designers.

Stage 3 - Develop and implement descriptive product design tool to support designers.

Stage 4 - Reflect on the descriptive product design framework to support designers.

Stage 5 - Refine and improve the descriptive product design framework to support designers.

Stage 6 - Improve and implement the improved descriptive design tool for the framework.

Stage 7 - Reflect and suggest future work for the descriptive design framework.

The details of the research work carried out in each stage will be elaborated in the coming chapters where research findings from Stage 1 will lead to the decision to develop and conceptualise a descriptive product design framework to support designers. Further findings in Stage 4 later contribute to the need to include an integrated approach into the framework to conceptualise solution concepts to improve the framework.

1.5 Outline of the Thesis

This thesis is organised in seven chapters. The details of each chapter are briefly described as follows:

Chapter 2: In this chapter, an extensive literature review of all the established engineering design methodologies ranging from normative, descriptive and prescriptive is briefly presented and they are critically compared from the perspective of their strengths, weaknesses and characteristics. The findings from these extensive reviews defined the research gaps and highlighted the prospective research work that will contribute to the understanding of the need for a new design methodology.

Chapter 3: This chapter describes additional investigations of the literature related to the type of support facilities that are provided by different available design methodologies and design tools. The outcome of this investigation provides important information on the type of support facilities that can be provided, the way designers were supported and the crucial role of these facilities to a designer. The investigations also determine the advantages and deficiencies of different design support facilities and how integrated these facilities are with the design methodologies and design tools.

Chapter 4: This is the chapter that will elaborate on the derivation of the descriptive design methodology framework. In this chapter, the origin of the descriptive design framework and the conceptualisation details of the descriptive design methodology framework are presented. This is then followed by the derivation of the descriptive design methodology framework and finally, the development of the prototype tool.

Chapter 5: A chapter that presents the application of the descriptive design methodology framework in a commercial design project case study to design a device for supporting concrete loading in between beams. This case study demonstrates and verifies how the framework computational-platform-related and the methodology-related support facilities help a designer in designing. The case study has two phases: Phase 1 is to design the device and Phase 2 is to improve the design. This case study involved an experienced designer and a designer from a customer from the construction industry.

Chapter 6: In this chapter, a case study based on a student project to design a conceptual end-effector for first aid robot (FAROS) is presented. The descriptive design methodology framework is applied along with an optional design tool based on TRIZ. The designer was a student and novice designer. The aim of the case study is to see how well the descriptive design methodology works with other design tools and how well it supports a novice designer who designs in a step-oriented design environment.

Chapter 7: This chapter reflects and concludes the findings of this research work. Future work based on the findings from the research work is also discussed and highlighted in this chapter.

Figure 1.2 illustrates the flow of the chapters and the structure of this thesis.

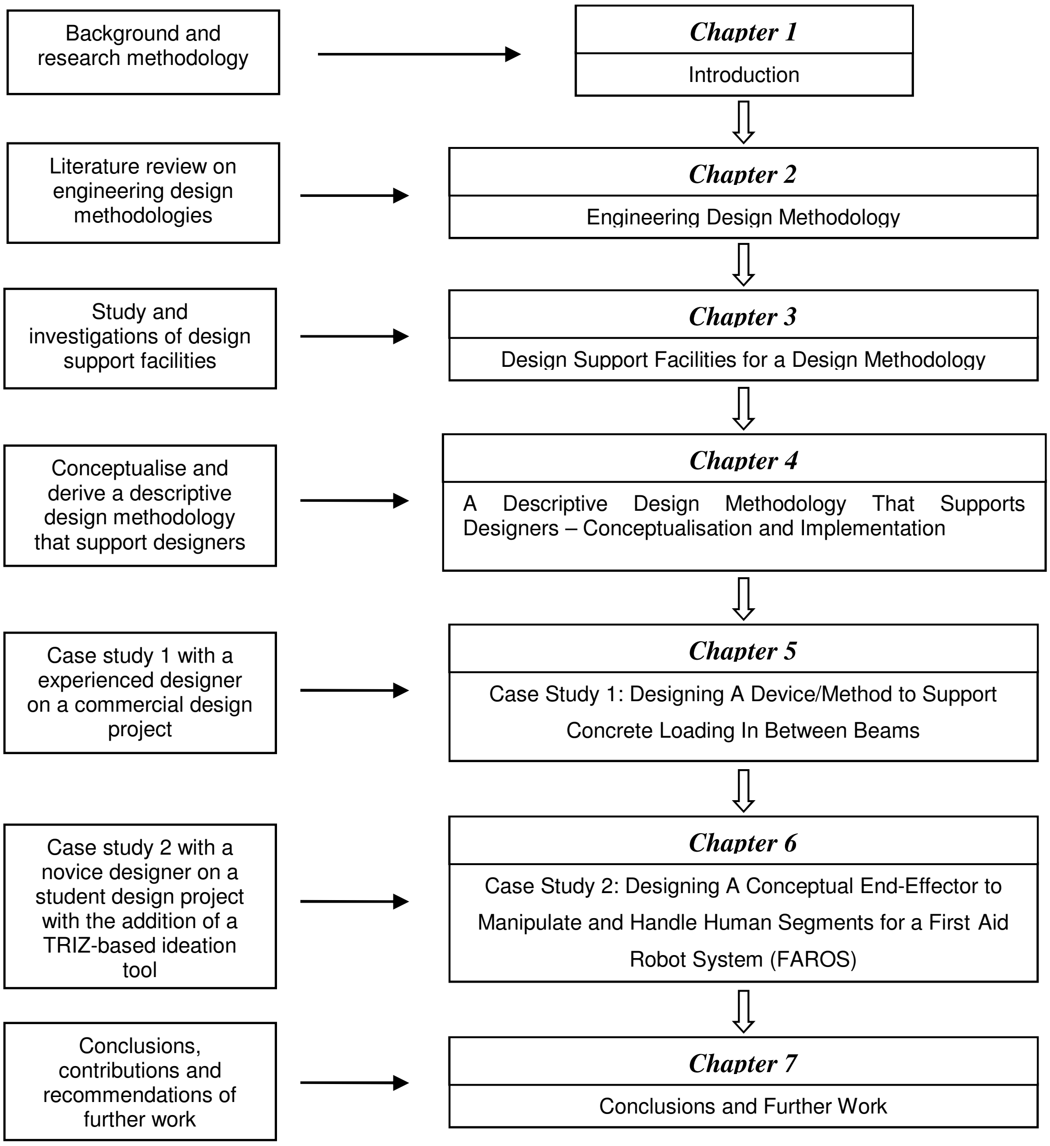


Figure 1.2 The flow of chapter of this thesis

Chapter 2
Engineering Design Methodology

2.1 Background of Engineering Design Methodology

An engineering design methodology is defined by Cross (1994) as

"any procedure, technique, aid or tool for designing with intention of bringing rational procedures into the design process."

Engineering design methodology is here defined as the process of transforming the requirements of human needs into a technical artefact that satisfies these requirements in a rational and systematic manner. As this scientific basis is presented, the question of whether design is an art or a science is raised. If design is an art i.e. a process that does not have a rational or systematic basis, then the design process is known to be a "bricolage" (Louridas 1999) or a tinkering process. A number of researchers such as Eder (1995), Finkelstein (1983), and Louridas (1999) considered design both an art and a science.

Several reviews and surveys have been conducted on the contribution of several engineering design methodology models and the findings acknowledged their positive contributions to the process of designing (Eder 2009; Evbuomwan et al. 1996; Finger and Dixon 1989a, b; Finkelstein and Finkelstein 1983). A majority of the engineering design methodology models in the literature are based on managing design phases or stages of design and utilised an analysis-synthesis-evaluation procedure approach (Cross 1994; Dym and Little 2000; French 1971; Hubka and Eder 1995; Jones 1970; Pahl and Beitz 1995; Pugh 1991; Roozenburg and Eekels

1995; Ullman 1997; Ulrich and Eppinger 2000). These methodologies have core similarities. Such engineering design methodologies models are also known as step-oriented from the strategy perspective and are categorised as prescriptive design models by Finger and Dixon (1989a). In addition to the prescriptive design models, from the literature studies, there are two more design methodology models: the normative design models and the descriptive design models (Finger and Dixon 1989a). These methodologies are supposed to provide support and guidance to designers during the design process. However, this is not the case for descriptive design methodologies as is shown in the section 2.3.

The distinction between these three engineering design methodology models can best be explained by Buchanan (1999) although he applied them to the decision-making models. From the perspective of engineering design methodology models, normative design models describe how design should be carried out while descriptive design models describe how design is carried out and prescriptive models describe how design should and can be carried out. Each of these models individually has advantages and disadvantages (Evbuomwan et al. 1996; Finger and Dixon 1989a). The next few sections will review these three models, explore these models in more detail and critically analyse the current established engineering design methodologies as well as their strengths and deficiencies. The chapter will conclude with a summary of the findings and the possible research gaps for the future direction of research on engineering design methodology.

2.2 Normative Design Models

Normative design models are rational mathematical models applied in engineering design which utilise probabilities (Siddall 1972), statistical, single or multi-attributes utility-based approaches (Thurston 1993) which include decision-based design methods (Hazelrigg 1996; Hazelrigg 1998; Thompson and Paredis 2009) and axiomatic methods (Suh 1990, 2001) to solve design problems. Normative models were proposed with the assumption that all designers will make decisions on a rational basis and that there is therefore a rational basis for all design decisions. However, there are challenges in that people do not always make rational decisions but rather make decisions within their own constraints – its bounded rationality (Chase et al. 1998). The next few sections explore some of the common normative design models utilised by designers.

2.2.1 Probability-based Normative Model

Normative models that utilise probabilities and statistical methods provide the designer with rational means to determine the design parameters. From these probabilities, they provide a mathematical basis to assist the designer in solving a design problem. Such normative models are of the compensatory type (allowing trade off) and have been proposed since the early 70s. For example, the probabilistic structural analysis methods provide a means to quantify the inherent risk of a design and assess the sensitivities of design variables. With this information, designers can make better design decisions. Figure 2.1 illustrates an example of a high-level depiction of a probabilistic design model of the Northrop Grumman Commercial Aircraft Division (NGCAD) probabilistic design methodology (Long and Narciso 1999).

The Northrop Grumman probabilistic design methodology employs numerical integration with Monte Carlo simulation to determine probability of failure of a structural component and/or system of structural components. Using this method also implies that the design solution is an optimum solution.

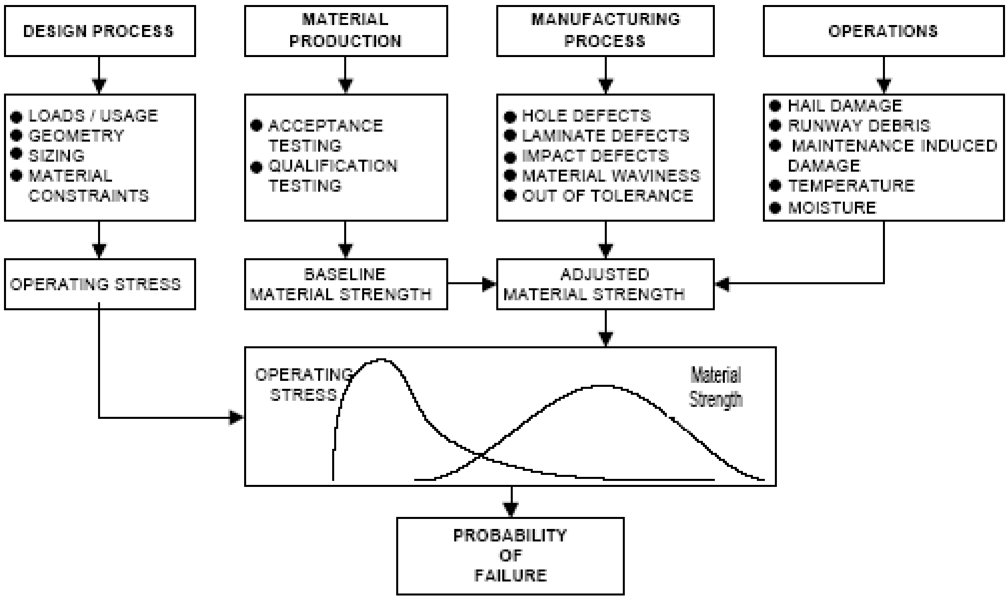


Figure 2.1 An overview of the Northrop Grumman Commercial Aircraft Division (NGCAD) Probabilistic Design Model (reproduced from (Long and Narciso 1999))

2.2.2 Utility-based Normative Model – Decision-based Design

The single or multi-attributes utility-based normative models are also of a compensatory type, which encourage the process of quantifying design attributes into a utility function. Among the multi-attributes utility-based normative models is the decision-based design model. This model enable designers to make better design decisions based on the analysis of the subjective utility function assigned to design attributes within a design environment with uncertainties. Based on these analyses they are able to assist designers to make better design decisions that will lead to a design solution with the highest utility value. The framework for decision-based design proposed by Hazelrigg (1998) is as shown in Figure 2.2. All the design alternatives are assigned a value of von Neumann utility (von Neumann and Morgenstern 1953). For each design iteration, the designers are expected to seek a better design or a design with a higher utility using this framework. Higher utility means higher profit, so the designers are expected to select a design configuration along with its dimensions and manufacturing processes or design variables which give the highest utility. In the framework, exogenous variables are random variables that the designer has no control over i.e. the weather, the future labour cost, and other costs. This also means that the design solution is usually an optimum solution. There is another type of normative design model, which works based on axioms. The next section will explore the axiomatic design model developed by Suh (1990).

2.2.3 Axiomatic Design Model

The axiomatic-based normative model is proposed by Suh (1990) and Suh's axiomatic design methodology is a function-oriented methodology from the perspective of strategy (Von der Weth 1999).

Suh's axiomatic design model (Suh 1990) is initiated by a list of identified functional requirements[1] and these functional requirements are decomposed and mapped directly to a hierarchy of design parameters from top down in a zigzagging manner as shown in Figure 2.3. The functional requirements and the design parameters are decomposed into hierarchies. The lower level functional requirements cannot be determined without first determining the design parameters at the level above. Although Suh's axiomatic design model (Suh 1990) is a normative design methodology, it has a framing effect and this forces designers to develop a design solution that complies with two main axioms i.e. the independent axiom and the information axiom. Suh's axiomatic design methodology is a non-compensatory model i.e. a model that does not allow trade-offs. A model that is non-compensatory is a model that will not allow a compromise of priorities on the specified design requirements and hence, the design outcome has to satisfy all the design requirements with the same degree of importance.

---

[1] Suh's design methodology is also a normative design methodology that defines desired outputs as functional requirements.

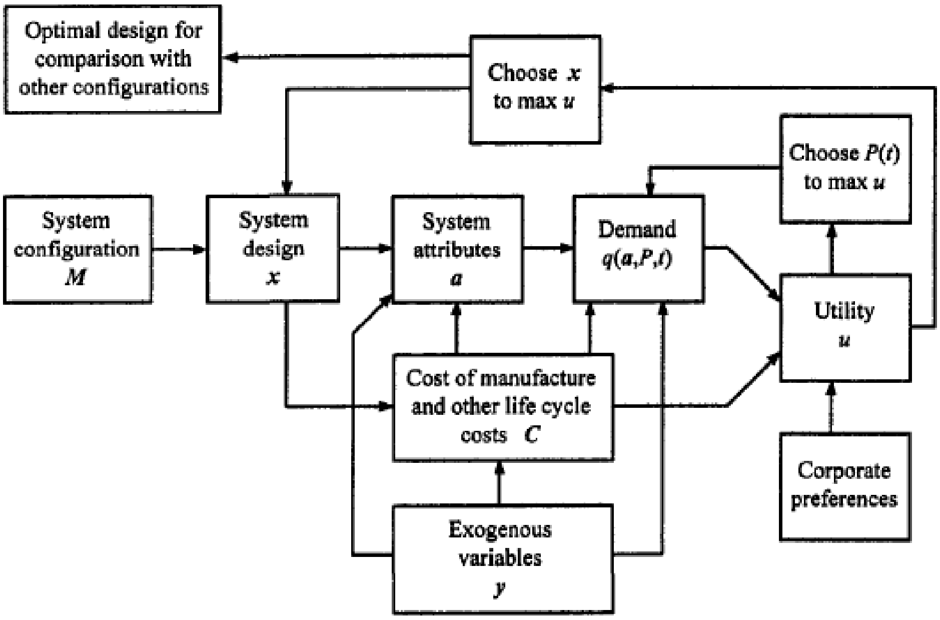


Figure 2.2 A Decision-based Design Model by Hazelrigg (reproduced from (Hazelrigg 1998))

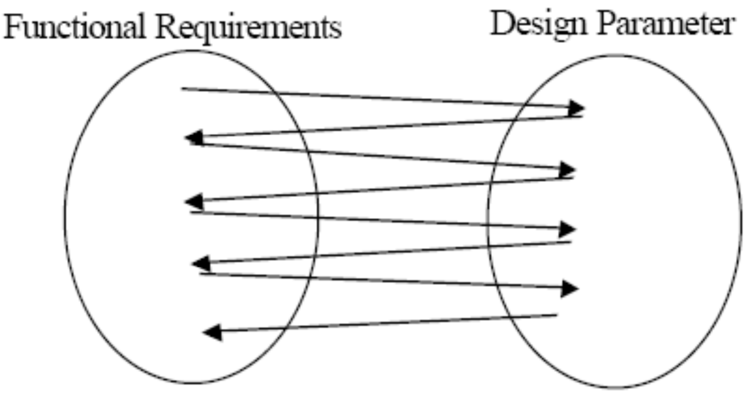


Figure 2.3 Suh's axiomatic design methodology (reproduced from (Suh 1990))

For Suh's axiomatic model (Suh 1990), the designer must derive design parameters in a zigzag manner from a higher level to a lower level that satisfy the two axioms adopted, which lead to either an uncoupled or a decoupled design solution. This means in a complex design, the designer will have to identify all the functional requirements and arrange them as a design equation matrix for the first level. From each functional requirement, the designer will then derive design parameters related to it. If the first level of functional requirements are found to be further decomposable then each of first level functional requirements will be further decomposed into second level functional requirements and their respective design parameters would be derived to form a second level of the design equation matrix. This process is repeated until the functional requirements cannot be decomposed further and the design parameters provide the design solutions. Throughout this process, every design equation matrix must satisfy the two axioms, namely the independent and the information axiom. In order for the design equation matrix to satisfy the independent axiom, it has to be a decoupled or an uncoupled design solution. Suh's axiomatic design model does not accept a coupled design solution at all costs. The design solution is also a "satisficing"[2] solution rather than being an optimum solution. Figure 2.4 illustrates the distinction between coupled, decoupled and uncoupled design where FR is a functional requirement and DP is a design parameter.

$$\begin{bmatrix} FR_1 \\ FR_2 \\ FR_3 \\ FR_4 \end{bmatrix} = \begin{bmatrix} x & 0 & x & x \\ x & x & x & x \\ x & x & x & x \\ 0 & x & x & x \end{bmatrix} \begin{bmatrix} DP_1 \\ DP_2 \\ DP_3 \\ DP_4 \end{bmatrix} \qquad \begin{bmatrix} FR_1 \\ FR_2 \\ FR_3 \\ FR_4 \end{bmatrix} = \begin{bmatrix} x & 0 & 0 & 0 \\ x & x & 0 & 0 \\ x & x & x & 0 \\ 0 & x & x & x \end{bmatrix} \begin{bmatrix} DP_1 \\ DP_2 \\ DP_3 \\ DP_4 \end{bmatrix} \qquad \begin{bmatrix} FR_1 \\ FR_2 \\ FR_3 \\ FR_4 \end{bmatrix} = \begin{bmatrix} x & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & x \end{bmatrix} \begin{bmatrix} DP_1 \\ DP_2 \\ DP_3 \\ DP_4 \end{bmatrix}$$

Coupled Design   Decoupled Design   Uncoupled Design

Figure 2.4 Distinction between coupled, decoupled and uncoupled design

[2] "Satisficing" is an action to find a design solution that is not optimum but merely satisfies the design requirements qualitatively. Optimum design solution means the design solution is the best solution in meeting design requirements.

2.3 Descriptive Design Models

Descriptive design models are engineering design methodologies that attempt to model designs from the perspective of the actual or natural[3] design process (from a cognitive point of view). These descriptive models work on the mental or cognitive processes of a designer by describing, simulating and emulating their cognition during design (Finger and Dixon 1989a). Descriptive models focus on determining the nature of design problems before attempting to solve them via cognitive techniques.

Among the popular descriptive design models found in the literature are the protocol study, question-based model, reflective design model, and design logbook model. Depending upon the way the question-based design model and reflective design model are applied, they can be descriptive models. When the question-based design and the reflective design models are applied in accordance with the actual way a designer carries out his design, they are classified as descriptive design models.

2.3.1 Protocol Study

In a protocol study model, designers are required to talk aloud about what they are thinking during design and what they say will be recorded. There will be a third party who will also take notes of what the designer is doing and saying. Most of these models are utilised for analysis to carry out research on engineering design. One of the most famous and comprehensive protocol studies was performed at Delft University of Technology in 1996 where a broad range of design studies and analysis were performed by a team of design researchers from all around the world (Cross et al. 1996). The recorded results were analysed from various perspectives and were very useful in enabling researchers to understand better how design

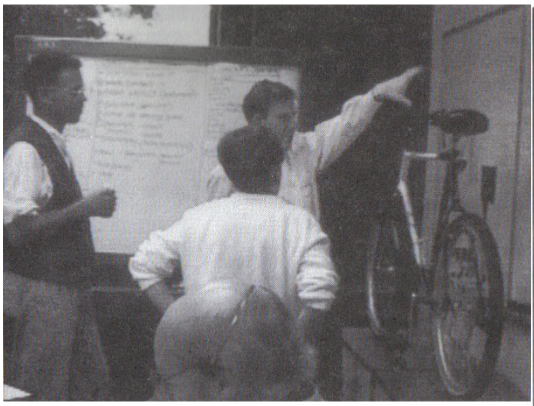
---

[3] The term "natural" means a designer is allowed to design according to his or her preference without rules, procedures, etc.

processes were carried out by designers. For example, from the protocol study, Akin and Lin (1996) managed to link the data from the protocol study to novel design decisions, Hideaki et al. (1996) tried to model the design process from the functional evolution perspective using the data from the protocol study, and Ullman et al. (1996) found out that some sub-requirements raised by designers during the design process were not addressed. Figure 2.5 shows a snapshot of a scene during the protocol study and the data collected. The protocol study is still widely used to find out and understand more about engineering design.

Protocol study is also used to find out more about how a novice designer designs and what are their differences when compared with experienced designers (Ahmed et al. 2003; Ho 2001; Kavakli and Gero 2002; Liikkanen and Perttula 2009). The findings from these studies are important and will contribute to the justifications and input for some of the work conducted in this research.

Similarly, Bender has proposed a systematic observation, analysis and categorisation methodology which utilises photo-documentation, non-participative observation using protocol function and cognitive strain test to obtain data (Bender et al. 2001, 2002a; Bender et al. 2002b) to collect data for the advanced analysis of the design process. Protocol study classifies the broad range of possible operations performed in design into groups of basic operations. By selecting the basic operation via pressing its related button on the software tool, the designer can be informed of what he is doing throughout the design process. This allows designers to avoid talking aloud while designing. Figure 2.6 shows the function protocol input of Bender's software tool. Protocol study is also utilised in investigations and studies on the collaborative design environment. The aim of these studies is to learn more about collaborative design (Brereton et al. 1996; Rosenthal and Finger 2006). Usually, design solutions proposed in a protocol analysis are "satisficing" in nature.



(K: puts on backpack)
(J: is 'riding' bike)
J so I I keep thinking that there's all this weight in this area between the seat but that could be used but I wonder if that would really work when you're like pumping really hard pedalling up a hill it sounds like
K between the seat where?
J I mean between the in the triangle
K well your knees work (inaudible)
J that was a pull on the brake-lever (laugh)
I can can you un
J un-?
I there's a way to un-.. un it
J oh to unhook it right there so you can
I I guess we don't need to (inaudible)
J that's OK so
00:27:00
K so we want to put it in there but I let's see if you got (inaudible)
J yeah you'd never really be able to
K you wouldn't be able to get your knees pedalling
(K: puts pack over rear wheel, horizontally)
K OK now what about
(K: puts pack over rear wheel, vertically)
K They've gotta a prototype that kinda has it this way
J is a facing forew
I yeah that's right facing forward OK
J would it be too funky to have on the like projecting from the front wheel?
I handlebars? yeah try that
(K: carries pack to front of bike)
J or off this handlebar stem even because that's fixed but if it's off the handlebars you know it's like an old bike basket that way like the Wizard of Oz (laugh)
(K: puts pack over handlebars)
K OK now steer tends to
J you could turn it long ways
I or if you could get it down low where the
K (inaudible)
I centre of gravity is low
(K: puts pack between handlebars and seat)
J oh yeah like a motorcycle gas tank
I oh yeah the gas tank there we go
J you know like on your on your Harley (laugh)
K except on a Harley you don't pedal so you don't get your knees cranking
(J: moves legs in reverse pedal, hits pack with knees)
J maybe they're gonna have to redesign the top of their backpack

Figure 2.5 How protocol study is carried out and a portion of the data collected (reproduced from (Cross et al. 1996))

2.3.2 McDonnell Descriptive Design Model for Interpreting Design

McDonnell (1997) has suggested the application of a descriptive model to interpret design using a systemic grammar network. The systemic grammar network utilises five notations of links as shown in Figure 2.7 for a design description. A design proposal to install a new transformer at a sub-station and how the systemic grammar network description provides an interpretation of the qualitative data is presented in Figure 2.8.

By applying the systemic grammar network, McDonnell (1997) model hoped to provide a better understanding of design via representation and this representation can capture that understanding.

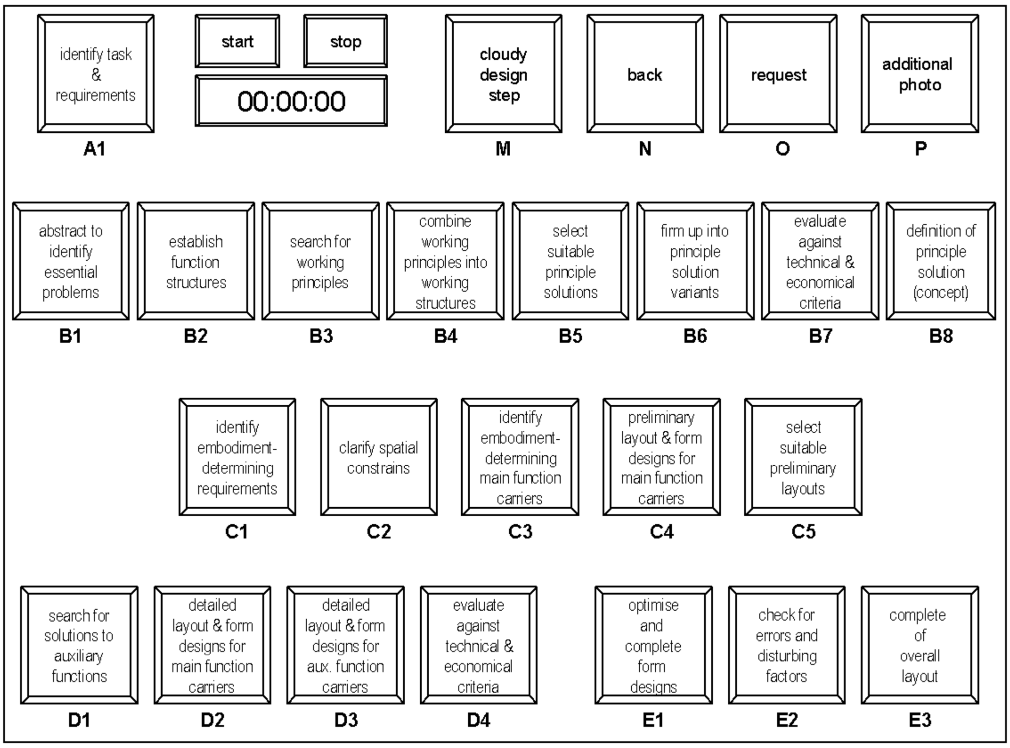


Figure 2.6 Function Protocol Input (reproduced from (Bender et al. 2002b))

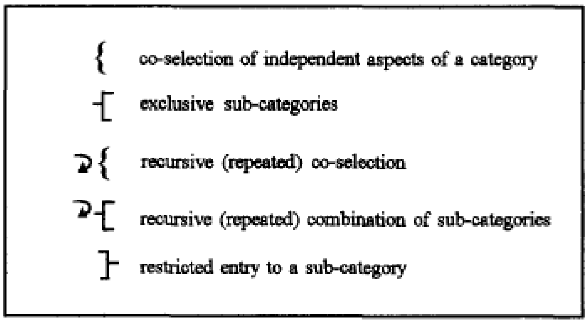


Figure 2.7 Notation for links in a systemic grammar network (reproduced from (McDonnell 1997))

Design Recommendation	SGN Path
"This paper sets out proposals to install the fourth 15 MVA 33/11 kV transformer at S1 substation	The design alternative proposed (*a*) is that the network be reinforced (*b*) by extending (*c*) S1 (location *d*) by installing (*e*) a transformer (*f*) as specified (*g*). [Connections information is specified later in the proposal - it completes this bracket.]
to enable S2 substation to be relieved of load by means of secondary distribution load transfers from S2 to S3,	Load is to be transferred [further selection from terms to the right of *a*] - (*h*) from S2 (*l*) to S3 (*j*). 12.3 MVA is the amount to be transferred (*k*) at secondary voltage (*l*, *m*). It can be inferred from the next sentence of the design recommendation that this is a transfer for normal operations (*n*, *o*).
and from S3 to S1.	Load is to be transferred also (*h*) [repeated entry to bracket to right of the term 'transfer load'] from S3 (*l*) to S1 (*j*). The amount to be transferred is 12.3 MVA (*k*) at secondary voltage (*l*, *m*), and again it is a transfer for normal operations (*n*, *o*).
The proposals given are short term measures until the CityS substation is commissioned in the Central Square area programmed for 1991/92.	This with a later statement [see * later in this column] relates to limiting conditions of project life (terms at the top of the bracket to the right of 'design proposal' *x*, *y*).
The transformer would be connected into an existing idle 33 kV oil filled cable installed between S4 and S1.	Connection to the new transformer, i.e. connection information, (*p*) [completing the bracket to the right of the term 'reinforce network' entered above for the substation extension] supply is from S4 (*q*) to S1 (*s*) for normal operation (*r*, *t*).
A one panel extension would be required to the 33 kV switchboard at S4,	More network reinforcement in the form of substation extension is to be effected (*b*) [repeated entry to bracket to right of 'reinforce network']. Substation extension (*c*) at S4 (*d*) by installing (*e*) a switch panel (*u*) as specified - 33kV, etc. (*v*) is to take place. Connections for the switch panel can be inferred (*p*), infeed will be from S4 (*q*, *r*) and will be supplying S1.
and a seven panel 11kV extension required at S1.	There is further network reinforcement in the form of substation extension (*b*) [repeated entry to the bracket to the right of the term 'reinforce network']. Substation extension (*c*) at S1 (*d*) is to take place by installing (*e*) a switch panel (*u*) as specified - 11kV, etc. (*v*). Connections for the switch panel (p) can be inferred [from the original comments about load transfer], infeed from S3 (*q*, *r*), supplying S1 (*s*, *t*).
The transformer and 33 kV switchgear would be taken from spare stock. The 11 kV switchgear extension would be new equipment.	These two sentences complete the brackets to the right of the term 'install equipment' for the three substation extensions giving a path through terms of increased delicacy to the right of the terms 'transformer' or 'switchgear' as appropriate.
Five new 11 kV interconnectors would be required between S1 and S3 to facilitate the 11 kV load transfers between these substations.	Further entries are made to the brackets to the right of the term 'reinforce network', connections are expressed by statements about load transfers as before, locations are S1 and S3 respectively and the path will include the term 'install circuit' (*w*) - not shown explicitly.
The S2 to S3 load transfer would be effected using existing interconnection.	
Project completion is programmed for October 1987.	*This with information earlier referring to CityS substation (*y*), gives start date (*z*) and limiting conditions (*y*) and with statement of cost (*A*) completes the leftmost bracket of the part of the SGN which represents the design proposal.
The total estimated cost is £xxx,xxx."	

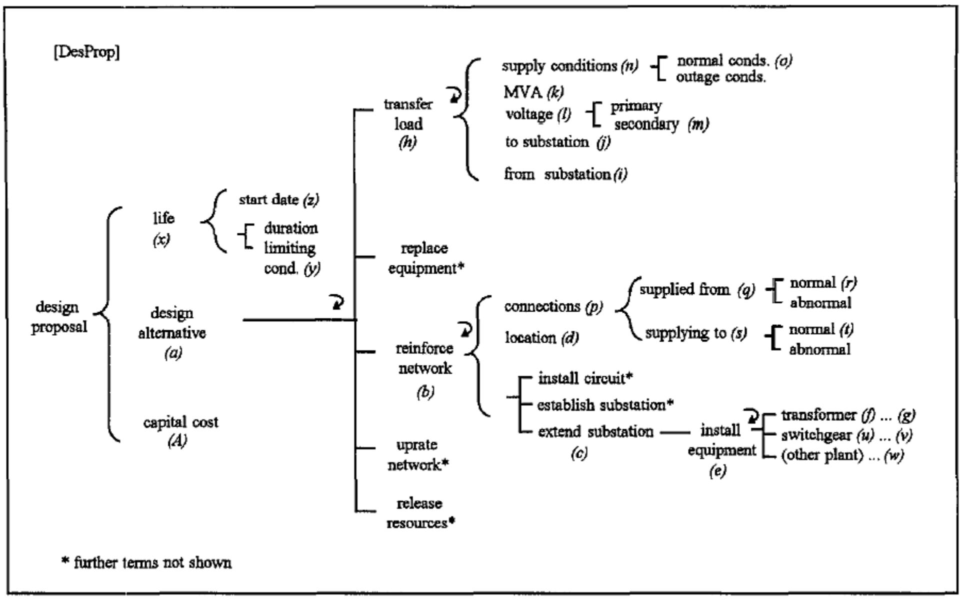


Figure 2.8 A portion of the system grammar network that represents a design proposal (reproduced from (McDonnell 1997))

2.3.3 Question-based Approach

The question-based approach was proposed by Eris (2004) around the notion that engineering design is a question-driven process. This approach attempts to create a question-decision centric theory that is able to promote a convergent-divergent thinking paradigm during design. The question-based approach is developed based on empirical experiments that use protocol study on several design teams. The question-based approach promotes structured questioning to improve design performance. By encouraging designers to engage in divergent thinking, the question-based approach helps designers to expand the design requirements into design concepts before going into convergent thinking to transform them into design decisions. In order to engage in divergent thinking, designers are encouraged to ask generative design questions (GDQ) and for the convergent thinking, deep reasoning questions (DRQ) are asked during the design process. The question-based design model is as shown in Figure 2.9. Figure 2.10 shows some examples of generative design questions (GRQ) and deep reasoning questions (DRQ). The categorisation of GRQ and DRQ are based on several taxonomies of questions proposed by Lehnert, Graesser and Eris (2004).

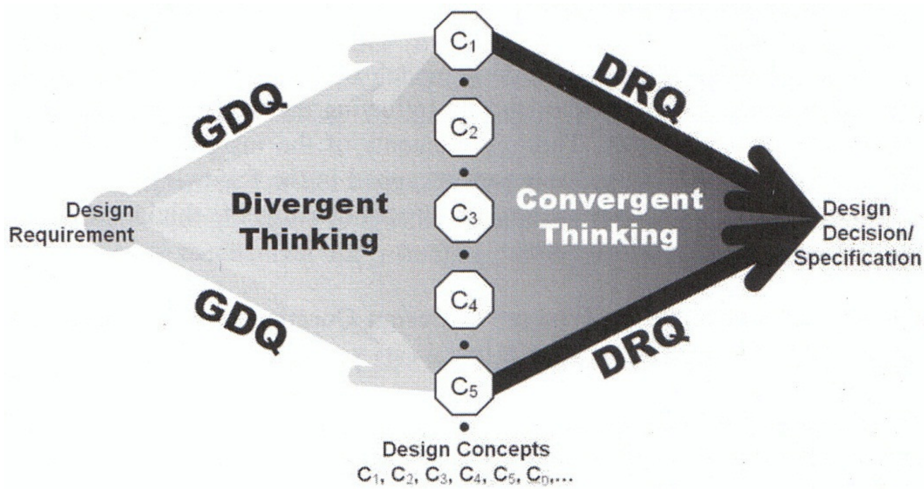


Figure 2.9 The question-based model that transform design requirement to design decision via generative design questions and deep reasoning questions (reproduced from (Eris 2004))

	Category	Example	
	Request	Can you hand me the wheel?	
	Verification	Did John leave?	
	Disjunctive	Was John or Mary here?	
	Concept Completion	What did Mary eat?	
	Feature Specification	What material is the wheel made of?	
	Quantification	How many wheels do we have?	
	Definition	What is a pneumatic robot?	
	Example	What are some flying insects?	
	Comparison	Does the small wheel spin faster?	
	Judgemental	Which design do you want to use?	
Deep Reasoning Question (DRQ)	Interpretation	Will it slip a lot?	Convergent Thinking
	Procedural	How does a clock work?	
	Causal Antecedent	Why is it spinning faster?	
	Causal Consequence	What happened when you pressed it?	
	Rationale/Function	What are the magnets used for?	
	Expectational	Why is the wheel not spinning?	
	Enablement	What did they need to attach the wheel?	
Generative Design Question (GDQ)	Enablement	What allows you to measure distance?	Divergent Thinking
	Method Generation	How can we keep it from slipping?	
	Proposal/Negotiation	Can we use a wheel instead of a pulley?	
	Scenario Creation	What if the device was used on a child?	
	Ideation	What can we do with magnets?	

Figure 2.10 The conceptual framework of questions-based on Lehnert's taxonomy, 4 of Graesser's taxonomy and an additional 5 categories added by Eris (reproduced from (Eris 2004))

The question-based model is also explored by Grebici (2009) to guide designers into their design inquiries by consulting the generic questions raised by their research work on design description taxonomies. Grebici (2009) developed a new design description taxonomy for his research work and Figure 2.11 shows an example of how generic design questions are used in their research.

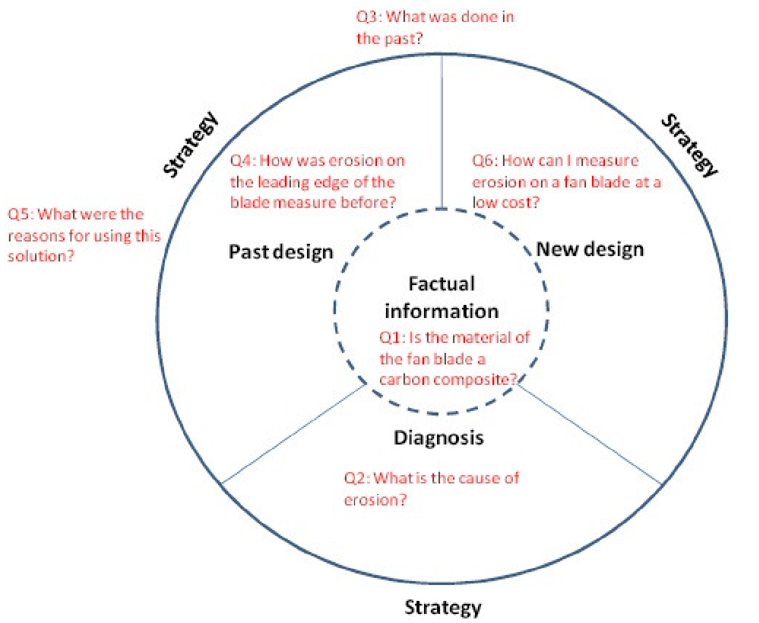


Figure 2.11 An example of how to use generic design questions (reproduced from (Grebici et al. 2009))

2.3.4 Reflective Descriptive Design Model

Reflective descriptive design is a model based on the design philosophy of Schön (1983). This descriptive design model was proposed by Reymen (2006) and was developed as a basis for a domain independent prescriptive model which helps a designer to reflect his design decisions better. Reymen's descriptive model reflects the actual process of design where design requirements are evolved throughout the design process time line from one state to another until it reaches the final state, which is the design output. Based on the notion that a design process is a reflective process (Schön 1983; Valkenburg and Dorst 1998), she then utilised this descriptive model as a basis to develop a model to assist a designer to reflect his design decisions. Hence, the proposed new model that assists designers with a structured reflection is a prescriptive model because a designer has to provide the properties, factors, and relations of the design situations and the design activities performed during the design process for every state of design. This is to help the designers to deal with three main activities of design as a process of reflections i.e. preparation, image forming, and conclusion drawing. With this in view, this section will only discuss her descriptive model but not her prescriptive model of structured reflection.

Figure 2.12 shows the Reymen's descriptive model that represents a design process as a sequence of reflections on a sequence of design situations. In Reymen's model, a design process consisted of a sequence of design situations where each situation is a snapshot or a state of a design process in time. Hence, Reymen's model represents a transition of states or design situations during a design process. A reflection on a design process occurs between the current and the past design situation. The reflection on a design process is aimed at answering essential questions such as "Am I solving the essential problems or am I busy with sub-optimisations?", "Does the result feel satisfactory or are further iterations necessary?", "Is my way of designing effective and efficient?", "Is my design process appropriate for the problem?" (Reymen and Hammer 2002)

This descriptive model is then used as a basis for her to develop a prescriptive design model that supports designers in reflecting on design situations during the design process. This prescriptive design model will be explained in the section 2.4.

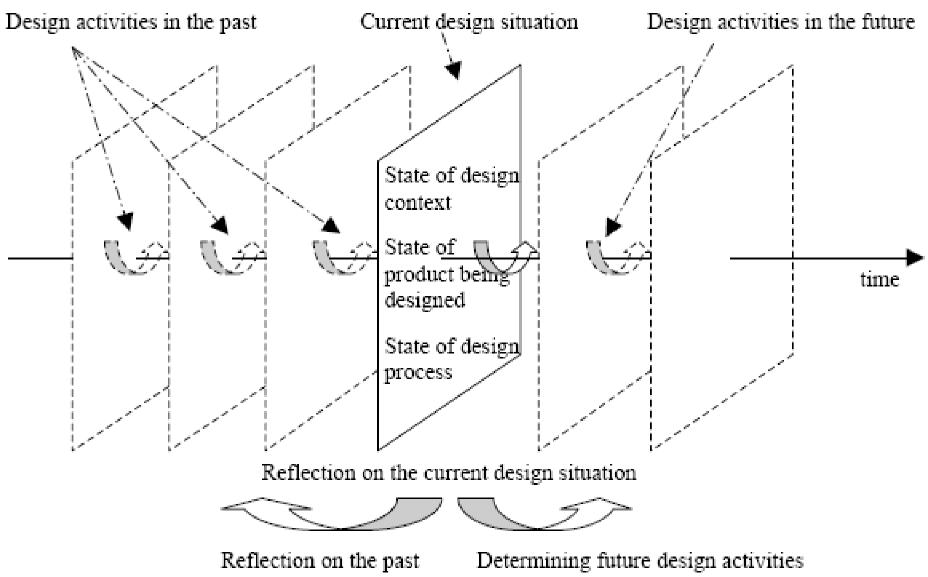


Figure 2.12 The reflection of a design process (reproduced from (Reymen et al. 2006))

2.3.5 Design Logbook Model

A design logbook is commonly used by designers to record information related to their daily thoughts, ideas, design sketches and design activities. This information is referred by the designers from time to time throughout the design process and serves as crucial evidence for any faults should any design failures and errors occur. Though the information in a design logbook is usually poorly structured and in the form of scribbling and full of annotations. Nevertheless, it is still a very important recording medium for a designer. Pedgley (2007) proposed the use of a logbook in the form of a diary to record and analyse the designer's own design activity. McAlpine (2009) proposed the use of an engineering electronic logbook (EEL) for designers to re-use information. However, the acceptance of designers in replacing their paper-based logbook is still low. The EEL proposed by McAlpine (2009) utilised the activity/object-based classification schema as shown in Figure 2.13. Figure 2.14 shows an example of an information input window of the EEL.

Activity templates	*Tags*	*Associated Metadata*
Design note Meeting note Pro-forma (customised)	Contact detail	Date, company, email, telephone, address, description
	Calculation	Date, subject, project, description
	Graph/chart	
	Sketch	
	Diagram	
	Table	
	External doc	Date, subject, project, description, source
	CAD	
	Annotated external doc	Date, subject, project, description, source, annotation author
	Annotated CAD	
	'To-do' item	Due date, project, description
	'Important' item	Due date, description
	Custom	*Custom*
	...	...

Figure 2.13 The classification schema elements used in EEL (reproduced from (McAlpine et al. 2009))

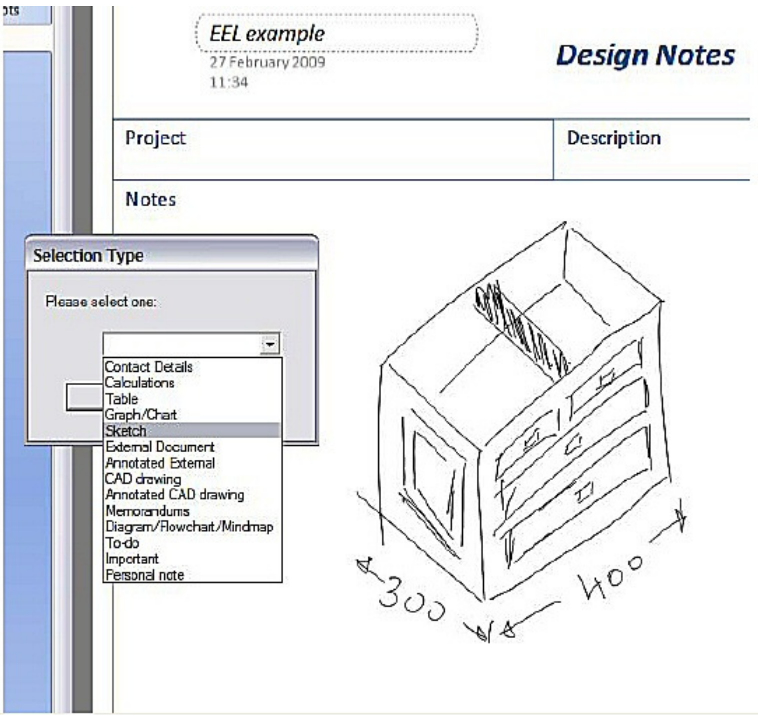


Figure 2.14 The drop-down menu tagging and the design notes used in EEL (reproduced from (McAlpine et al. 2009))

2.4 Prescriptive Design Models

Prescriptive design models are design methodologies that advise or prescribe techniques and methods to assist designers in designing. Most of these models are step-oriented models, which solve design problems step-by-step from phase to phase. Among the established step-oriented prescriptive design models are models proposed by Pahl and Beitz (1995), Pugh (1991), Roozenburg and Eekels (1995), Ullman (1997), Cross (2000), Hubka and Eder (1995), French (1971) and Ulrich and Eppinger (2000). These models possess a combination of normative and descriptive model characteristics. This is because one must know the nature of the design problems before trying to correct them and the way to correct them is usually based on rational mathematical techniques. Hence, prescriptive design models inherit both positive aspects of normative and descriptive design models by providing a more systematic way to design. In addition to that, a good prescriptive model also needs a good descriptive model (Baron 2004). This generic systematic approach of prescriptive design models has created immense interest among the research community.

Even though a number of step-oriented prescriptive models have core similarities, as shown in Table 2.1, there are also prescriptive models, which attempt to assist designers from the perspective of function, issue and collaboration. These prescriptive models are known as issue-based models, function-based models and collaboration design models or group design models respectively.

2.4.1 Step-oriented Model

Most established engineering design methodologies that represent design as a phase-based process are step-oriented methodologies (Von Der Weth and Frankenberger 1995). Step-oriented design methodologies aim to improve the design process by allowing designers to design in a systematic framework, which is based on design phases. A design phase[4] (represented by a small rectangle in Figure 2.15) contains a list of recommended generic design activities, rules and guidelines for a designer to carry out his or her design activities following a systematic design process. In general, step-oriented design methodologies can be simplified as shown in Figure 2.15. A step-oriented prescriptive methodology starts with a requirement planning phase. The outcome of this phase is the design requirement specifications or product design specifications (for product design). When the design requirement specifications are determined, the next phase is the conceptual design phase. This is the most crucial phase when compared with all other phases (French 1971). This is because the conceptual design phase is where solution concepts were derived and it is also the starting phase of the design process. This means that a lot of important design decisions are made in this phase and any error will incur the highest redesign cost. A solution concept is usually a design with a pre-determined configuration of required functional components.

Deriving solution concepts to produce design outcomes that meet the design requirements is important. There are several established methods to assist the designers to derive and generate solution concepts, namely, brain-storming, the Delphi method, Method 635, Synectics, TRIZ (Theory of Inventive Principles), morphological analysis, lateral thinking and creativity templates. These methods are ideation methods which support the designers and will be explored in depth in the next chapter. The design solution concepts are usually derived and evaluated based on the "satisficing" method i.e. qualitative evaluation on how well the concepts meet the design requirements.

After deriving the solution concepts, the next phase is the embodiment design phase. The designers need to select an appropriate solution concept which will best meet

---

[4] Conceptual design, embodiment design, and detail design are design phases. Some design methodologies do not have an embodiment design phase, for example Pugh model (1991) and the Ulrich and Eppinger model (2000).

the specified design requirements before they proceed with the embodiment of the design concept. Hence, the decisions made in selecting the appropriate solution concept are crucial before the selected solution concept can proceed into the embodiment phase. Most step-oriented design methodologies recommend decision analysis tools such as decision matrix (Pahl and Beitz 1995), multi-attribute/criteria decision analysis tools such as SMART (Goodwin and Wright 2004), SMARTER (Barron and Barrett 1996; Edwards and Barron 1994), and AHP (Saaty 1994). These tools can assist designers to make decisions in selecting the “best” solution concept method to proceed to the embodiment design phase. In decision analysis, it is vital to select a design concept that meets all the design requirements. However, there are cases where designers are unable to derive design concepts that meet all the design requirements. In such cases, trade-offs are required.

For some prescriptive design models that do not have an embodiment phase, the next phase is the detail design phase. With the embodiment phase, designers start to develop the selected design concept into working structures and the design layouts of a product. These working structures are in the preliminary form designs and will be further developed in the detail design phase with the design layouts finalised. Therefore, in the detail design phase, the designers will complete the definition of the dimensions, final layout, forms, material used, and relevant properties of the product.

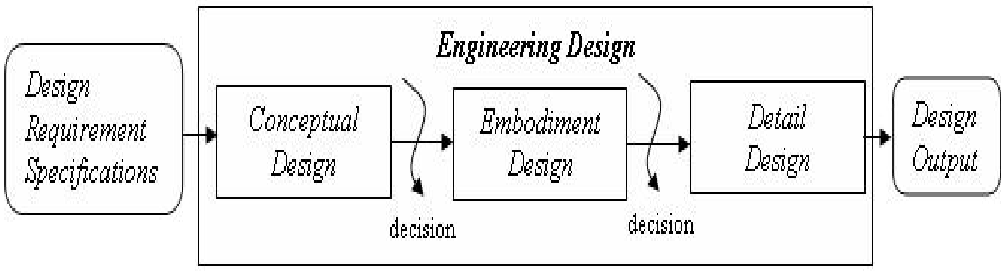


Figure 2.15 A Generalised Model of Step-Oriented Prescriptive Design Methodologies

These prescriptive methodologies assist designers to design via a framework that manages their design activities in a systematic manner and does not interfere with the cognition or the mental process of creating a design solution. At the end of each design phase, the designer is required to decide (refer to Figure 2.15 and the curvy arrow with the word "decision") on which design alternative to select to proceed to the next design phase. Table 2.1 illustrates the comparison between the different step-oriented prescriptive design methodologies.

2.4.2 Issue-based Model

Issue-based models provide a representation of a design process as an issue-solving process in a tree or graph with nodes. It attempts to capture the design rationale behind the option taken as well as the associated arguments in solving issues encountered throughout a design process. One of the well-known issue-based models is the issue-based information systems (IBIS) concept for capturing complex design decision (Bracewell et al. 2004) though there have been a lot of variations of IBIS concepts. Figure 2.16 shows an example of an issue-based model proposed by Lahti (1996). The importance of capturing the design rationale and the way to solve issues in these models is because it allows the tracing of the root source of each design decision, which then enables the reuse of design solutions and for the detection of design errors.

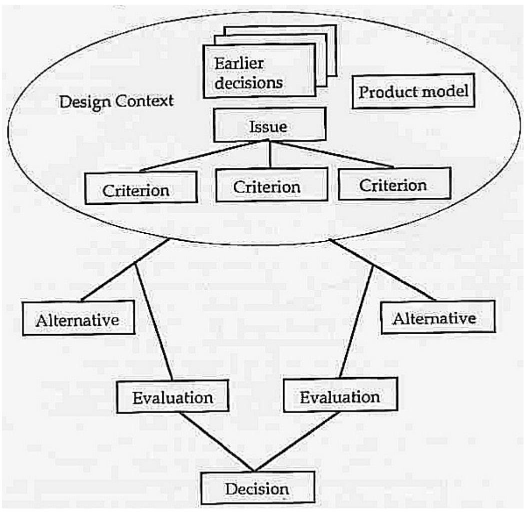
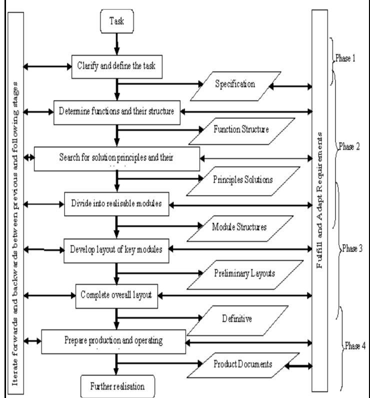
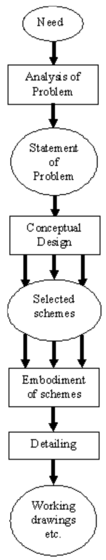
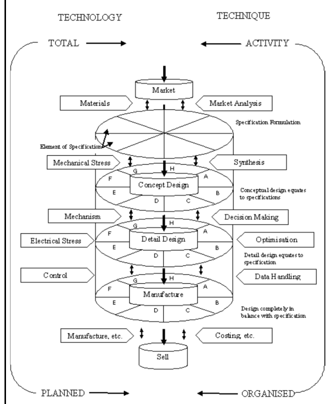
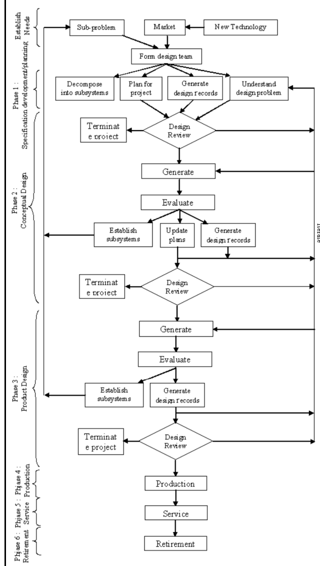
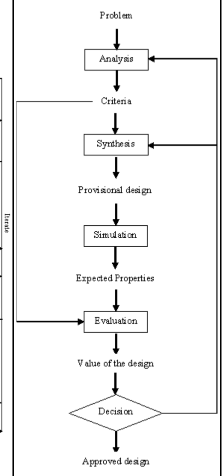
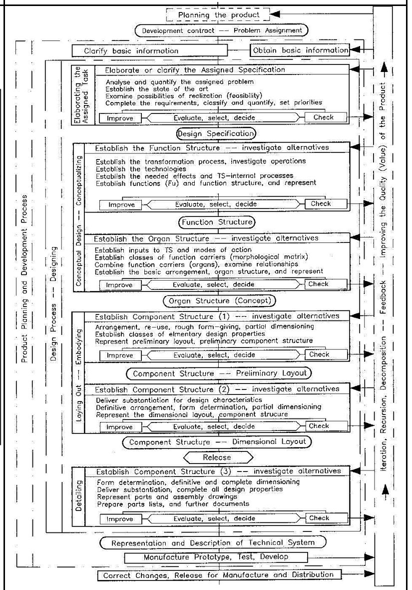


Figure 2.16 An example of a Lahti issue-based concept model (reproduced from (Lahti et al. 1996))

Table 2.1 Summary and comparisons of several step-oriented prescriptive design methodology models

Comparison Criteria	**Pahl and Beitz's Design Model (1995)**	**M.J. French Design Model (French 1971)**	**Stuart Pugh Design Model (1991)**	**David Ullman Design Model (1997)**	**Roozenburg & Eekels Design Model (1995)**	**Hubka & Eder Engineering Design Model (1995)**
Flow Chart						
Basis of Approach	Task based	Problem based	Product based	Product based	Problem based	Product based
Type of Model	Prescriptive	Prescriptive	Prescriptive	Prescriptive	Prescriptive	Prescriptive
Technique of assistance	Guidelines and tools	Guidelines and tools	Guidelines and tools	Guidelines and tools	Guidelines and tools	Guidelines and tools
Target of Model	Meeting requirements	Meeting requirements	Meeting requirements	Meeting requirements	Meeting requirements	Meeting requirements

Lahti's (1996) issue-based prescriptive model, like all other issue-based models, requires the designer to identify and record the issue encountered throughout the design process. The designer is also required to provide alternatives i.e. the possible solution concepts based on the "satisficing" method, the criterion that a product has to fulfil, the evaluation method, the decision on the selected solution concept and the respondent, which is the person who made the decision on the solution concept selection. Bracewell (2004) also developed a software tool known as DRED based on the issue-based model to capture the design rationale for a design process. His model utilises symbolic elements to represent different design issues for the design process. Figure 2.17 illustrates the application of DRED to the design of a mobile arm support.

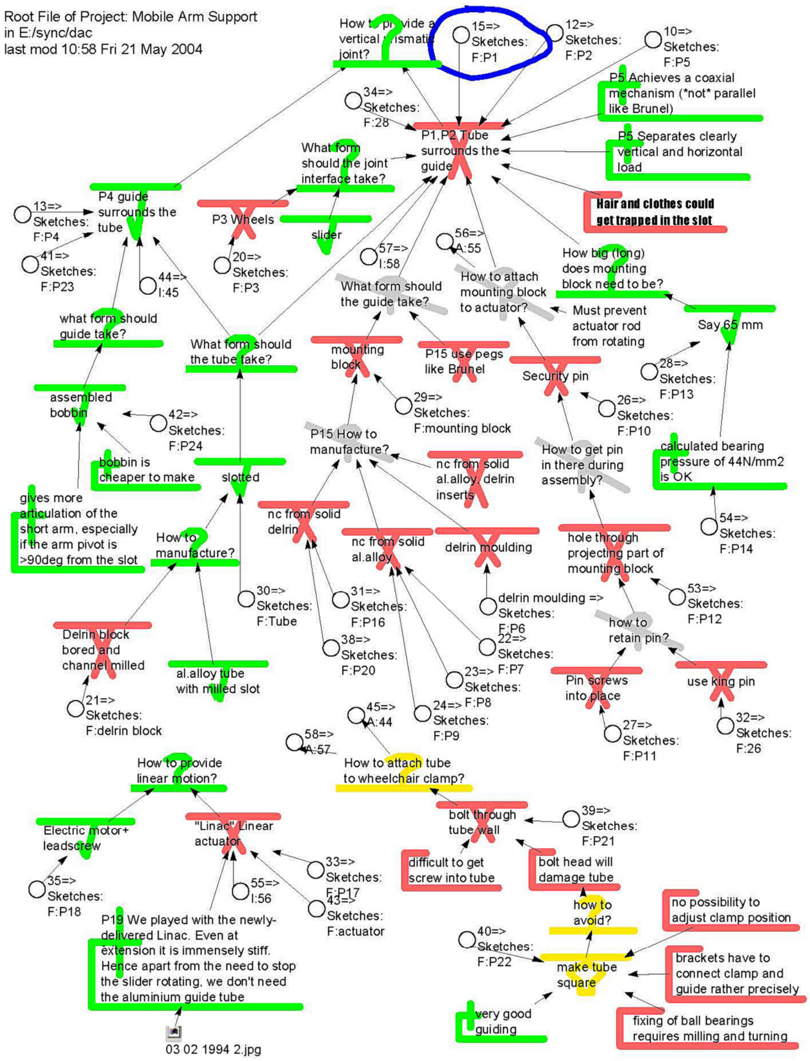


Figure 2.17 The rationale structure work plane for the mobile arm support project (reproduced from Bracewell (2004))

2.4.3 Function-based Model

Function-based design models are among the well-known models that try to represent a design process from the perspective of functions. There are several definitions of functions and different researchers define function differently. Due to this, there are inconsistencies in the definition of functions and this is one of the core issues in function-based models. A function can be defined from three perspectives according to Shah et al. (2001). These three perspectives are as

1. a purpose or intended use of a feature, component, or product
2. an abstract formulation of a task that is independent of any particular solution
3. a description of a task necessary to describe what an artefact is expected to do.

It is also important to differentiate clearly between function and behaviour. Van Wie et al. (2005) offers "behaviours are the physical events associated with a physical artefact or hypothesized concept over time or simulated time as perceived by an observer". A function is described as "the physical effect imposed on an energy or material flow by a design entity without regard for the working principles or physical solutions used to accomplish this effect differently" (Van Wie et al. 2005). In short, "a function is what an artefact does, a behaviour is how the artefact actually does it" (Scott and Antonsson 1996).

Function-based model is an active research domain (Bryant et al. 2005; Chakrabarti and Bligh 2001; Deng 2002; Fernandes et al. 2011; Goel et al. 2009; Hirtz et al. 2002; Johnson 1991; Kirschman et al. 1996; Stone and Wood 2000; Szykman et al. 1999; Thomas et al. 2009). Function-based models usually attempt to develop function taxonomies and ontologies. By utilising combinations of function taxonomies, the function-based models are able to capture the causal knowledge about the design of an artefact. With the causal knowledge captured, the function-based models are able to support designers via reuse of the knowledge captured. A function-behaviour-structure model is one of the established function-based model which tries to link the functions of a product to the behaviour and the structure (physical form) of a product (Regli et al. 2000).

Figure 2.18 shows an example of the taxonomy of functions applied to represent the design process of a cordless screwdriver and vehicle car seat (Hirtz et al. 2002). While Figure 2.19 illustrates the basic function taxonomy used by Kirschman et al. (1996) and Figure 2.20 shows the user interface of the tool developed by Kirschman (1996) for a functional design for a cordless electric drill.

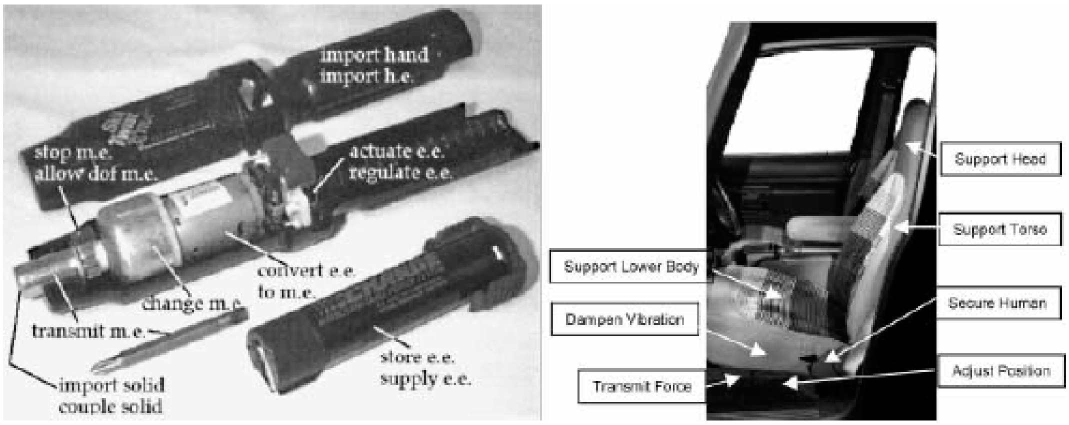


Figure 2.18 The functional labels for a cordless screwdriver and a vehicle seat (reproduced from (Hirtz et al. 2002)).

Motion	• Rotary, Linear, Oscillatory, Other • Create, Convert, Modify, Dissipate, Transmit • Flexible, Rigid
Control	• Power, Motion, Information • Continuous, Discrete • Modification, Indication • User–supplied, Internal Feedback
Power / Matter	• Store, intake, Expel, Modify, Transmit, Dissipate • Electrical, Mechanical, Other
Enclose	• Cover, View, Protect • Removable, Permanent • Support, Attach, Connect, Guide, Limit

Figure 2.19 The basic function groups of taxonomies (reproduced from (Kirschman et al. 1996))

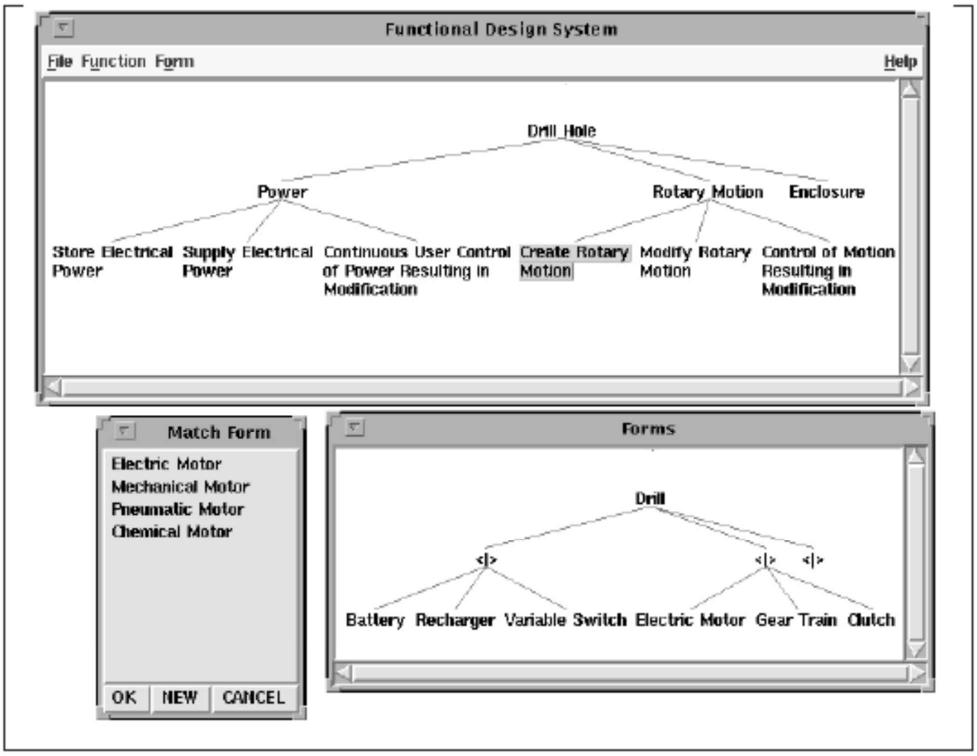


Figure 2.20 The user interface of the functional design tool developed by Kirschman (reproduced from (Kirschman et al. 1996))

2.4.4 Design Reflection-based Model

As mentioned earlier in the descriptive design model section, Reymen (2006) developed a prescriptive design model based on a transition of design situation model which is descriptive. In this prescriptive model, the designers are assisted to reflect on their past design situations to deal with the current design situation. However, designers are required to provide a certain amount of information that is related to a design situation, namely the description of its properties and factors related to the design task with values for their attributes. Hence, the designer must provide the basic attributes for the properties and factors for a design situation. The basic attributes are the label, the text, the value, the sources, the reference, and the rationale for a design situation. The proposed model utilised a text-based form to represent the design situation to be used for design reflection and thought. A design process is a series of transition process from a state to the other. Designers are not used to describe these states (Reymen and Hammer 2000). In order to ensure that the designers provide the relevant properties and factors at a certain moment in time for a design situation, a checklist has been developed by Reymen (2001) as a guide to the designer. The design reflection-based model encourages the designer to reflect on the design situation as frequently as possible throughout the design process. This is because design reflection does not necessarily occur in every design activity (Reymen and Melby 2001).

The main menu for the reflective-based design software tool is shown in Figure 2.21. The main menu of the tool provides a checklist description of a design situation which can be used to define specific attributes such as properties, factors and relations. These specific attributes of properties, factors and relations can be changed according to the designers' preferences for a particular design situation.

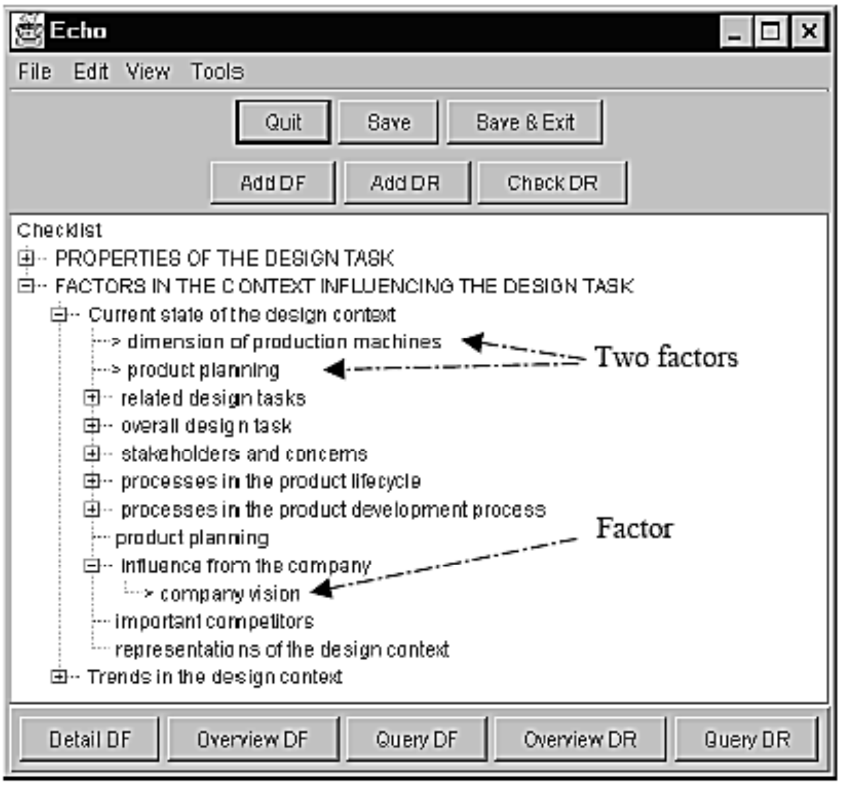


Figure 2.21 The main interface of the checklist description for design situation (reproduced from (Reymen and Melby 2001)).

Figure 2.22 shows the menu for defining design factor for designers to define the attributes for a particular factor, where in this case is about a kind of a coating.

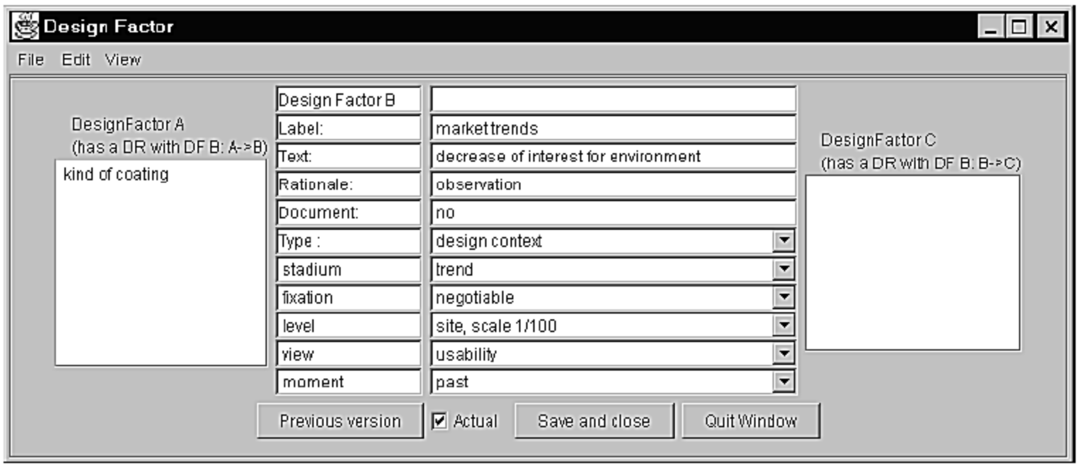


Figure 2.22 The user interface for designers to provide detail information on a particular design factor in a design situation (reproduced from (Reymen and Melby 2001)).

Figure 2.23 shows the text-based use interface for the design factors and design relations from the software tool developed to assist the designer to reflect on their design situation and the information the designer needs to provide for the tool (Reymen and Melby 2001).

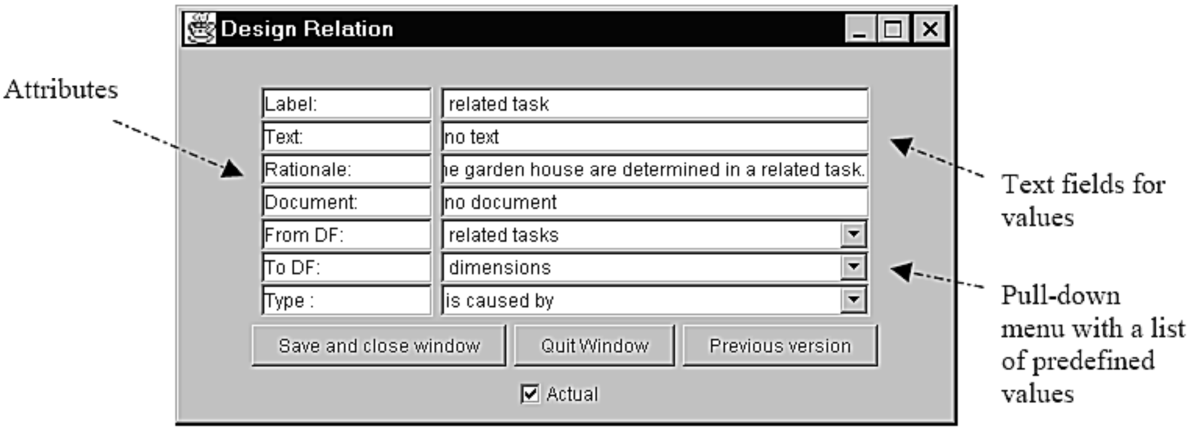


Figure 2.23 The text –based user interface of the design relation in which designers have to provide their input for the Reymen's reflection based model (reproduced from (Reymen and Melby 2001)).

2.4.5 Collaborative Design Model

Collaborative design models are prescriptive methodologies that focus on modelling a design process that involves collaboration between two or more designers and are an active on-going research domain. In a design collaboration environment, the focus of research is more related to social and organisation aspects of design, which can significantly influence collaborative decision making. Hence, it is a model that is based on social orientation.

According to Baron et al. (2003), group decision making may involve a team of people who will make decisions on certain issues or problems but people in the group may not be working together or linked to each individual in any aspect except to make a decision on a certain issue. Collaborative decision-making involves a group of people who are linked in a particular task and each individual decision will have effects on another individual task. However, collaborative decision-making has many similarities to group decision making as both environments involve many participants in making decisions. Among these similarities are the five features that group decision making is dependent upon (Baron et al. 2003). These features are size of group, composition of group, cohesiveness of group, communication and leadership.

Features such as the size of the group, composition of the design group, and leadership can be improved with the appropriate social, management and cognitive models and they are actively being researched. From the engineering design perspective, most of the design collaboration research focuses on two factors, improving communications and enhancing the cohesiveness of the design groups. In order to improve communications and cohesiveness of design groups, most research work on collaborative decision-making mostly concentrates on group interaction (communications while working together) and co-operation, conflict management, and resource and information sharing.

So, why collaborative design? There are general perceptions that many heads are better than one. If this is true then design judgments and design decisions made by a group of people should be better than an individual. Once again, the question arises how would we know whether a design decision is better? The answer to the question can be evaluated based on process or outcome. This is always debatable but group decisions have additional weaknesses as shown in Table 2.2 (Baron et al. 2003) and these weaknesses may also manifest in the collaborative design environment. The weaknesses listed in Table 2.2 are perceived from the social perspective. With current design projects getting more complicated and larger, it is impossible for a single designer to solve all the design problems in a large design project. Hence, collaboration in design is inevitable in a large design project.

Table 2.2 Summary of several weaknesses of group decision making (reproduced from (Baron et al. 2003))

No.	**Weakness Feature**	**Weakness Explanation**
1.	Group Size	Solution maybe more likely to achieve with bigger groups but it will be more difficult to manage.
2.	Conformity	The group members may shift their preferences to avoid being the odd one.
3.	Group member characteristics	Lower status members will be less confident and dominant group member is the downfall of many group decision processes.
4.	Social loafing	Some members maybe lacking in effort.
5.	Free riding	Commitment of some members may be less if others are performing sufficiently.
6.	Inequity based loss	If some members are performing insufficiently, it may lower others motivation.
7.	Production blocking	This weakness is common in face-to-face group where those who cannot verbalise their ideas will be soon forgotten.
8.	Evaluation apprehension	This event is due to group being pressured to fear of making non-positive contribution because of the involvement of external judging or being judged by outsiders. Hence, self-censor may occur that leads to removal of possible constructive contribution.
9.	Cognitive inertia	Formation of a mental representation of a problem which causes difficulty of people in changing their perception.
10.	Biased information pooling	Groups discuss and share information that is available to all but fail to share those information available to individual.
11.	Confirmation	Identification of a promising alternative and the group selectively focus on it.

With this in context, the designers may be located at different geographical locations and the ways design decisions are made are different. Design decisions maybe made based on negotiation, consensus and compromise. Therefore, most of the collaborative design models aspire to improve these three aspects. These models are derived as a basis for developing technological tools to help designers to bridge the problems caused by these two factors so that they can make better design decisions. Any issues encountered in a collaborative design environment can cause delay and mistakes in design decisions, which will lead to project delays and costly errors. The next three sections will elaborate the type of methodologies that were researched to support collaborative design.

2.4.5.1 Methodology for Group Interaction and Co-operation

The importance of group interaction (which is related to communication and social interaction) in a design collaboration environment is also actively investigated by researchers (Brereton et al. 1996; Crilly et al. 2008; Parent 1997; Simoff and Maher 2000). These interactions may involve two or more designers or between designers and the consumers or experts. However, most design research in group interactions are carried out for the purpose of trying to understand how designers interact and communicate. Among the methods proposed to improve the design collaboration between designers particularly from the communication and interaction perspective is the agent-based approach (Huang 2004; Jin and Zhou 1999; Liu et al. 2005). The agent approach utilises active programmed entities of codes with some extent of randomness that are able to actively adapt to the changes in the environment and work autonomously to assist communications and co-operation between designers.

Other than the agent-based approach, researchers also actively explore the prescriptive model for group interaction and communications in a collaborative design environment by taking advantage of the internet or World Wide Web (Huang 2002). Huang and Mak (2002) introduced a web-based design review framework using the systematic theory of axiomatic design review in a collaborative design environment after developing a web-based collaborative conceptual design tool (Huang and Mak 1999). Roy and Kodkani (2000) also proposed a web-based tool that searches the internet to assist concept development and a conferencing tool to allow communication among designers.

2.4.5.2 Methodology for Conflict Management

Conflict management is another research area of collaborative design that has been modelled by researchers. Differences in opinion and conflicting preferences in a collaborative design environment are expected and researchers have been proposing ways to solve such differences and conflicts faced by designers that work in groups. Resolving design conflicts has been of interest to researchers decades ago. Klein and Lu (1989) developed an explicit hierarchical representation of conflict resolution expertise. See and Lewis (2006) expanded and further developed the hypothetical equivalents and in equivalents method (HEIM) to support group decision-making for designers. His model attempted to address some significant issues in aggregating group member preferences. Lu and Cai (2000) used petri nets as a systematic representation method for the collaboration process and hence, they are able to express design state transformation, task dependencies and decomposition. With this representation, design conflicts can be detected and handled to support designers from the perspective of co-ordination.

2.4.5.3 Methodology for Resource Sharing

Resource sharing in a design collaboration environment is an area that is more related to information technology than social sciences. Among the resources that can be shared in a collaborative design environment is design data sharing (Davis et al. 2001). Most of the resource sharing research for the collaborative design environment is related to creating a kind of repository or database to facilitate the sharing process. Noel and Brissaud (2003) developed a knowledge-based tool using

Unified Modelling Language (UML) for dynamic data sharing within a collaborative design environment. Urban et. al. (1999) introduced the integrated product data environment (IPDE) as a database or repository approach based on STEP to share data between engineering design and an analysis tool. IPDE consist of three main components as shown in Figure 2.24: the integrated product database (IPDB), the shared data manager (SDM), and a set of domain access interfaces (DAIs). The IPDE adopted modified STEP concepts of Units of Functionality (UoFs) to support its functionality. With that, the users are able to manage the product data of different versions, accountability and maintaining the necessary relationships information.

It can be summarised that collaborative design model is geared towards assisting designers to design by a providing an information technology architecture to improve and support the issues related to social and group organisation.

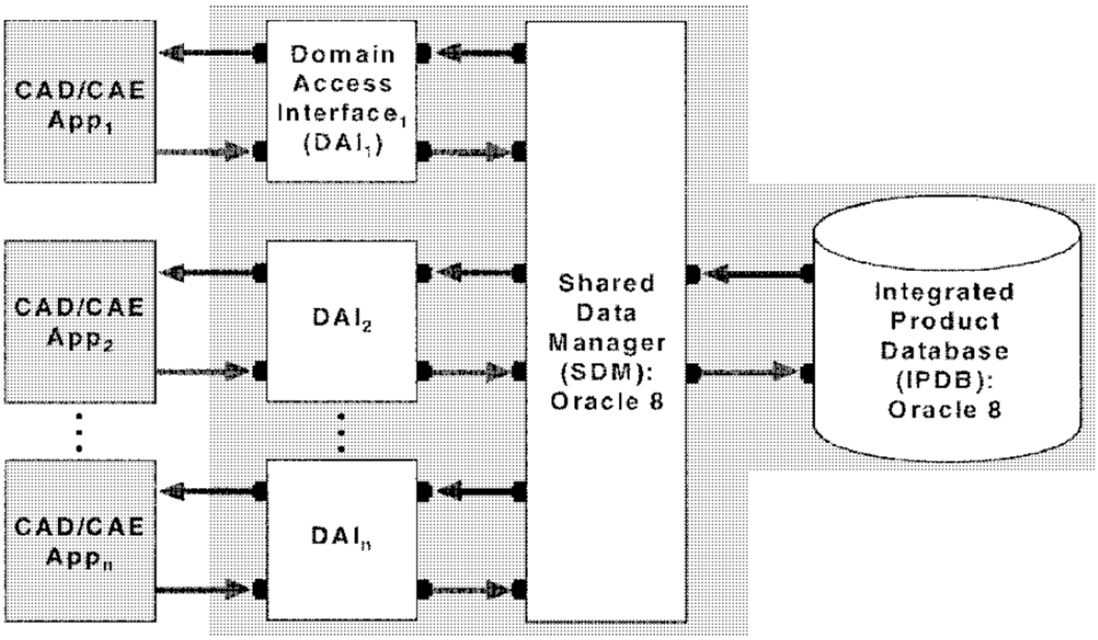


Figure 2.24 The architecture of IPDE (reproduced from (Urban et al. 1999))

2.5 The Deficiencies and Strengths of the Established Design Methodologies

From the literature review, the differences between models for a single designer environment and collaborative design environment are distinct. The model for a single designer environment is focused on solving a particular design problem from the perspective of managing the design (for prescriptive design models) and how to make a rational design decision (normative design models). The models for a collaborative design environment focus on social and management aspects between designers such as interaction, co-operation, and conflict management.

In addition to those distinct differences, every engineering design methodology model has its deficiencies and strengths. Some of them are inherent in the model, from which they are conceptualised while some are due to the way they were implemented. The next three sections explore the deficiencies and the strengths of the three normative, descriptive and prescriptive design methodologies. The deficiencies and the strengths of engineering design methodologies will be analysed against the findings from the empirical studies on designers during the design process in reality.

2.5.1 Deficiencies and Strength of Normative Design Models

For normative design models, assigning subjective utility functions in decision-based design models in a consistent and rational manner, and free from cognitive bias, is not an easy task (Thurston 2001). In the case of the probabilistic model, statistical data is crucial in determining the probabilities of a design parameter successfully. Hence, a sufficiently large amount of data is crucial for a probabilistic design model to work well but this is not always the case. When lack of sufficient data is encountered, parametric modelling methods that involve the fitting of parametric functions to data may be applied. However, as with the use of all parametric models, the designer should be critically aware of their shortcomings. As mentioned by Long and Narciso (1999), a famous statistician Breiman said that "all models are wrong but some are useful." For Suh's axiomatic model (Suh 1990, 2001), it is crucial to derive design solutions that will meet the two axioms, the independent axiom and the information axiom which will lead to uncoupled design solutions but this is a difficult task. The study conducted by Hirschi and Frey (2002) showed that it is difficult to derive a design solution to satisfy the two main axioms required by the model. Hirschi and Frey (2002) also found that the critical requirement of Suh's axiomatic model to reject a coupled design solution at all costs was flawed. In addition to that, Olewnik and Lewis (2005) also proved that Suh's methodology is fundamentally flawed as his method forces designers to conform to a particular preference structure. Thurston (2001) also found Suh's method (Suh 1990, 2001) most likely to be impractical if the design problems are complex and it is difficult to put into practice if the design project has a severe time constraint because a design solution that complies with the two axioms may not be found during that duration. Suh's method also does not consider design criteria and constraints. It only considers functional requirements.

From the strength perspectives, the normative design methodologies such as the decision-based design and probabilistic design models provide a rational basis for designers to decide on which design alternatives have the best utility or the best statistical chance of being successful in solving a design problem. For Suh's axiomatic design methodology, the framing structure of Suh's axiomatic design methodology (Suh 1990) provides some level of traceability and it is also a function-oriented methodology which is found to have a better focus in meeting design requirements. Traceability in design is important and is considered an advantage for any methodology because empirical studies conducted by Cooke et. al. (2003) showed that designers have difficulty in identifying and detecting the source of a design error when it happens. Cooke's studies concluded that the source of a design error is not due to a particular source but to a sequence of minor design decisions that individually may seems correct but collectively lead to a design error. Thus, identifying and determining the source of error is not as easy as it seems.

2.5.2 Deficiencies and Strength of Descriptive Design Models

Descriptive design models clearly focus on capturing or recording the design process as they try to describe the actual design process carried out by the designers. Most of these descriptive models are utilised for study purposes. By capturing the activities of designers during a design process, a huge amount of information and data can be gathered. Unfortunately, this also means a lot of the information and data gathered are not relevant to the design problem. Information and data gathered is also poorly structured and may not have any relational link, which is important in ensuring that the design process is a systematic process. These descriptive models are usually non-compensatory and may be resource intensive, not generic, and may be difficult to implement particularly for protocol studies. These characteristics are apparent as descriptive design models are basically psychological models that reflect the nature of humans (or designers). There are also inherent issues related to these descriptive models that may lead to inconsistent or irrational outcomes as designers are prone to making mistakes and errors. Hence, it is obvious by capturing all information and data during the design process without structuring it is ineffective and cannot assist the designers. Therefore, it is not surprising that most descriptive design models are applied to study and analyse design activities or to find out more about the actual design process. In the reflective design case, it is used as a basis of developing a prescriptive design model based on supporting design reflection.

For the question-based approach, it is considered to be a descriptive design model in this research because in the actual design process, the designers do raise a lot of question implicitly and explicitly regardless of the strategy, the approach or the methods used. The classification of deep reasoning questions and generative design questions as well as creating guidelines of the appropriate questions to be raised at different design activities are considered as a kind of support. It is obvious that such support is very abstract and is heavily dependent on the knowledge of the designer. In addition to that, the question-based approach also clearly lacks structure and systematic characteristics to support designers.

The strength of descriptive design models is evident. These models are able to accommodate different styles of designers and the different approach that may be utilised by the designers. The ability to capture the design information and data during design also provides a variety of opportunities. The information can be re-used, analysed and improved upon if the irrelevant information and data on the design process can be filtered out. Similarly, with the capturing ability of the descriptive design models, the models are inherently able to provide a traceability feature to a certain extent on the data captured. Re-use of information and data enhances design efficiency as designers does not have to design from scratch again (Ong et al. 2008). It also enables the utilisation of past information to be used to improve existing design and for solving new design problems. However, this traceability feature is limited by the current poorly structured nature of the established descriptive design models. Hence, some degree of engineering organisation and structuring of information and data is essential to enable an effective re-use of the information and data captured.

2.5.3 Deficiencies and Strength of Prescriptive Design Models

Prescriptive step-oriented design models are ones that look into the activities of the design process flow from the conceptual to the detail design phase. Hence these models are just providing guidelines and advice to designers on each design phase assuming the designers design in a way that moves from one phase to another. This assumption is found to be inaccurate for experienced designers from the empirical studies conducted by Fricke was quoted by Von der Weth (1999). Experienced designers were found to often use the function-oriented design approaches, which are more focused, time saving but are also more risky. This is because the approach encourages the designers to make important decisions early without carefully analysing the task, solution principles and concepts (Von der Weth 1999). Such function-oriented approaches contradict the principles of the step-oriented models, which require the designers to explore for design solutions and concepts, and to analyse them before selecting them.

From the utilisation of design support tools perspective, step-oriented models are merely guidelines and advice. They encourage the utilisation of various tools such as a decision matrix, morphological analysis and others during the design phases to assist designers on various activities along these design flow processes. However, these tools are utilised in isolation to deal with specific issues encountered throughout the design phases.

Therefore, it is not surprising that step-oriented design methodologies are rarely followed by practical designers (Stempfle and Badke-Schaub 2002) and do not even work under ideal laboratory conditions. The work of Stempfle and Badke-Schaub (2002) also showed that one of the design teams that used a step-oriented design methodology failed to solve the design problem posed in his experiment because the designers did not refer back to the design requirements consistently throughout the design process. This failure may be attributed to the nature of these step-oriented design methodologies, which advocate searching for a design solution that meets design requirements rather than deriving a design solution from design requirements as in function-oriented methodologies. With these deficiencies, it is not surprising that experienced designers are less interested in adopting these methodologies as the success of a design outcome in satisfying the requirements is the ultimate aim of any design process.

Further empirical work conducted by Von der Weth (1999) also found that experienced practical designers (without utilising step-oriented methodology) are actually practicing some form of function-oriented methodology that is more time-saving and is still able to produce successful design solutions. Additional empirical studies (Chakrabarti et al. 2004; Ullman et al. 1996) also showed that designers have a tendency to forget, ignore, misinterpret or lose track of the design requirement specifications during the design. Akin and Lin (1996) also demonstrated that minor design decisions[5] are made throughout the design process within a design phase before leading to the design solution. Current step-oriented design methodologies did not consider these minor design decisions sufficiently which can inevitably cause the design problems described by Cooke (2003). The deficiencies highlighted so far are just for the prescriptive design methodologies that are of a step-oriented type.

For function-based prescriptive design models, they provide designers a function-oriented approach in a systematic manner with the intention of capturing the causal knowledge of the design decisions made. A function-oriented design approach basically means an approach where designers design with the required functions of a component of the product or the end product in mind. While the design phases are not emphasised by function-based models, they concentrate on using specific behaviour and function terminology and the capturing of the links from behaviour to the structure of the final design. However the function-based models proposed by researchers (Hirtz et al. 2002; Johnson 1991; Kirschman et al. 1996; Stone and Wood 2000; Szykman et al. 1999) have severe limitations. These models may not be sufficient in representing the actual function-oriented design approach without becoming complicated (due to crisscrossing of links). The model has severe deficiencies in creating comprehensive but distinct taxonomical and the ontological terms for functions used in design. These deficiencies can cause confusions and uncertainties. These models also do not consider other factors besides behaviour or function that are important in the process of design such as size, weight, strength, shape and others.

The issue-based model has similar problems to function-based models in its inability to avoid messy and complex representation for design. This type of model has a structure that is also lacking in direction and spread exponentially, hence it is unable

[5] Akin's study considers a design decision to be any and all intentional declarations of action/information for the design problem at hand and represents it as a "novel design decision" which is known as a minor design decision in this report.

well to represent the design process if the design task gets bigger and more complicated. Issue-based models also have been found to be not very practical, confusing and rarely applied successfully in industry (Bracewell et al. 2004).

Reflective actions in design were first described by Schön (1983) and later Valkenburg and Dorst (1998) studied reflective practice in design teams before Reymen (2006) proposed a structured reflective design model to help designers. The reflective design model only focus on the supporting a designer to reflect on the current design situation from the last design situation. The model also required a designer to describe and analyse design situations and design activities throughout the design process by means of using checklists and reflecting at the beginning and at the end of each design session via forms. The needs and the requirements to provide information on properties and attributes of forms, which include design relation form for reflection purposes, are cumbersome. Designers are also required to determine the basic attributes for properties and factors such as labels, value, source, reference and rationale as well as attributes for relations and others. It is obvious that Reymen's (2006) reflective design model is investigative in nature and does not sufficiently represent relations among attributes within the same design situation and among multiple design situations. The current design reflective model is only able to support text-based descriptive attributes without the image forming and conclusion drawing that are key factors in a reflection process (Reymen and Melby 2001).

As for collaborative design methodologies, the literature review has shown that these model focusing mainly on social perspectives of design such as communication, interactions, team conflicts and others. The key need for an individual designer to derive successful design concepts or ideas on how to solve a design problem effectively is not addressed directly by the collaborative design methodologies. These methodologies support designers utilised information and communication technology such as agents to assist in solving conflicts and to finalise the design solution based on compromises and negotiations.

Even with some of the deficiencies of the prescriptive design elaborated, the prescriptive design methodologies, particularly the step-oriented methodologies, are still widely taught and are incorporated in the syllabus of design education. The importance step-oriented design methodology in helping designers to conceptualise design solutions by deriving useful and insightful function structures is acknowledged (Chamberlain et al. 2001). The step-oriented design methodologies also provide a useful systematic framework for structuring and management of the design process, generation of design concepts, and tools for evaluation and decision in design (Finkelstein and Finkelstein 1983). A systematic and structured framework for engineering is important to improve the design process. The other prescriptive design methodologies such as function-based, issue-based and reflective design seek to introduce a structure to capture and reuse of design information to assist designers. So far the prescriptive frameworks have been for a single designer, which is the core of any design process. However, most of the current design projects are complex and involve multi-disciplinary design teams in different geographical locations. Such design projects pose an additional challenge in communications, interaction and social-based issues and the collaborative design framework is actively researched to deal with these issues but the core of deriving ideas and design solutions is dependent on each individual designer in a design team.

2.6 Analysis of Established Design Methodologies Models

The findings from the literature review showed that prescriptive (Pahl and Beitz 1995) and normative design methodologies such as Suh's axiomatic design (Suh 1990) impose a systematic and rational design approach without consideration for the designer's own preferences and design strategy. Empirical study showed that designers design with different approaches and strategies, particularly among designers who are regarded as experts (Von Der Weth and Frankenberger 1995). This is why current prescriptive and normative methodologies are not widely adopted. This is supported by findings from empirical studies showing that designers rarely apply current prescriptive and normative design methodologies in industry (Hansen and Ahmed 2002; Tomiyama et al. 2009). Various reasons for the delay in acceptance of these methodologies by industry have been suggested (Eder 1998; Hansen and Ahmed 2002). This problem is further compounded by the need for designers to adapt the design methodologies to the specific problem, time scale and others as these methodologies are formulated in a very general and abstract manner.

In short, the step-oriented design methodologies only promise a systematic approach but not assuring they will deliver successful design outcomes and they do not accommodate differences in design approach and strategy. Allowing designers to design according to their preferences and approaches is critical because designers are not generally familiar with the established prescriptive design methodologies (Eder 1998). Although designers are not interested in current descriptive and normative design methodologies, studies have shown that they still need design support. This is because they can still make poor design decisions (Ullman 1995) and have difficulty in determining the source of design errors (Cooke et al. 2003). Further analysis of empirical findings showed that a design methodology that possess flexible characteristics to accommodate different strategies and approaches of designers is crucial to producing good design performance (Bender and Blessing 2003). Therefore, there is a need to derive a design methodology that describes the actual design process and supports designers in designing based on their individual preferences and approaches.

Contrary to the normative and prescriptive design methodologies, descriptive methodologies allow designers to design according to their preferences but are rarely employed to support designers. Most descriptive design methodologies, such as protocol analysis, have been applied for the purpose of analysis, validation, and investigation of the design process (Cross et al. 1996). This is because descriptive design methodologies are often used only to provide a better understanding of prescriptive or normative methodologies and as a means to formulate them. Findings gathered from descriptive methodologies can be diverse and conflicting. A reason for this is the different approaches and preferences of designers (Fricke 1996; Von Der Weth and Frankenberger 1995). Designers were found to design according to their past experience, knowledge, approach and pace. Hence, findings from descriptive methodologies do not always lead to prescriptive or normative methodologies. Furthermore, the most common descriptive design methodology, protocol analysis, is impractical as it requires designers to speak aloud, captures irrelevant information, and produces records that can be misinterpreted (Galle and Bela Kovacs 1996). Hence, it is not surprising that design researchers have resorted to experience and logical argument to derive prescriptive and normative methodologies. For example, Pahl and Beitz (1995) proposed a design methodology based on the assumption that searching a wider solution space improves a design outcome (Blessing et al. 1998). Such an assumption may seem legitimate but studies have found otherwise (Günther and Ehrlenspiel 1999). Empirical studies have indicated that most designers do not follow any design methodologies and did not search a wide solution space when they design (Günther and Ehrlenspiel 1999; Stempfle and Badke-Schaub 2002). Further empirical studies also show that different designers adopt different design strategies and approaches.

One of the crucial findings of the literature review is that the step-oriented methodologies are also found to be less focused when compared with the function-oriented ones. Searching for design solutions to meet design requirements encourages exploration for design solutions but such exploration increases the chances of finding design solutions which may not meet the design requirements or sub-design requirements identified by the designers, particularly when the design methodology is mere guidelines and advice. This problem is currently dealt with by design iteration i.e. by redesigning and a significant number of design iterations may be needed to finalise a design output that meets all design requirements. Unlike step-oriented methodologies, function-oriented methodologies are more focused as it encourages designers to derive the design solution based on every sub-requirement and design requirement with little exploration. These methodologies also provide some level of traceability. With these findings and from the perspective of having better focus in meeting the design requirements, Nam Suh's axiomatic design model

(Suh 1990) when compared with the prescriptive step-oriented based models (Pahl and Beitz 1995; Pugh 1991; Roozenburg and Eekels 1995; Ulrich and Eppinger 2000) seem to be a better design methodology. However, an exploration of design solutions has its advantages because it increases the probability of coming out with more innovative design solutions. Is it possible to derive a design methodology that is more focused on meeting design requirements and which allows designers to design based on their own approach and preferences and yet allows designers to explore for design solutions in a systematic manner? Can such a design methodology be derived in a way which also provides a structure to support the various available design and decision tools?

The differences among different design methodologies are shown in Table 2.3. These differences and the strength along with the deficiencies of the three design methodologies will be analysed and investigated with the empirical studies that explore the needs of a designer during the design process to derive a desired design methodology to support designers. The practical and additional features need to be investigated so that they can be incorporated into the desired design methodology to provide better support to designers. This desired design methodology should allow designers to use their preference and enable them to review their design preference while producing a design output focus on meeting all functional requirements.

The literature review also identified the way design solutions are accepted from the perspective of meeting the design requirements. The normative design methodologies all look for an optimum design solution with exception of Suh's axiomatic design. This is possible when the design process is quantified based on utility value or probability. The other methodologies use the "satisficing" method to decide on how well design solutions meet the design requirements. "Satisficing" is a term coined by Simon (1982) for a solution that is good enough to meet a requirement.

From the empirical research work of Quinn (1980) who has investigated how design was conducted by designers in many companies showed that most designers derive design solutions that just met the design requirements ("satisficing") instead of searching for an optimum design solution. He also found most designers derive a design solution based on an approach known as the "logical incrementalism" philosophy. The "logical incrementalism" philosophy is a pro-active branching approach where designers act on urgent design requirements first and can recognise the available time to explore the remaining design requirements. There are times when design decision cannot be made because significant information necessary for that decision is not available. Hence, comes the idea that the design decision should not be taken if important information cannot be determined at a single point in time but has to be developed over time and through the building up of experience. The designer uses time to refine his understanding of the proposed development and to gain acceptance for the solution. Through this process, a solution emerges and is developed over a considerable period of time. The decision takes the form of a series of actions that explore and develop the solution while building a greater commitment of resources and a consensus to support the development. Quinn (1980) also found that designers design to "satisfice" most of the time in reality. Quinn (1980) demonstrated his proposed model with the case study of a well-known UK producer of hand cleansers. However, Quinn's model (1980) lacks a clear structure. As shown in Figure 2.25 which represents the root and branch approach, the "logical incrementalism" has a similar approach.

This empirical finding is critical as no design methodologies focus on advising or suggesting to designers that they should delay making design decisions if there is insufficient vital information and that the design decision can be delayed. This deficiency is probably because the current design methodologies are lacking of structures that enable the tracking of design decisions throughout the design process. The process of tracking design decisions allow designers to know where the design process is going, where the current design process is at present, and how the design process has progressed.

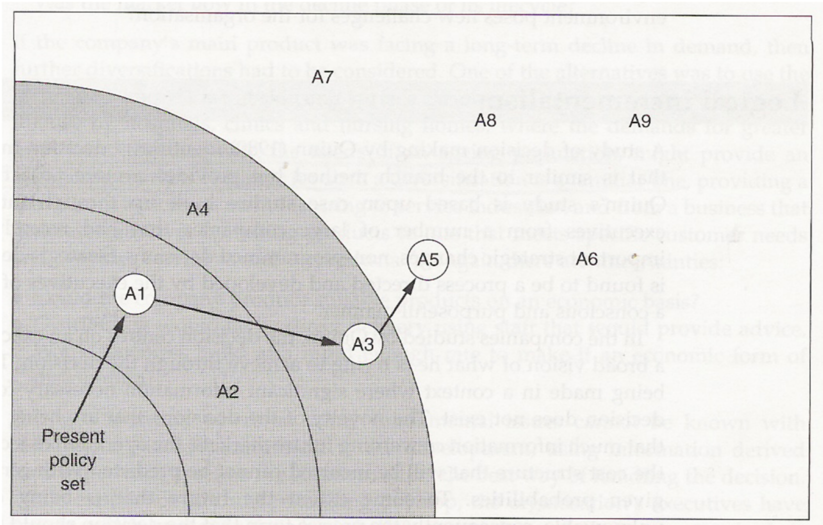


Figure 2.25 The "logical incrementalism" is similar to the root and branch approach which is shown here (Quinn 1980)

2.7 Desired Features of Engineering Design Methodology

Summarising the deficiencies and strength of current established design methodologies highlighted earlier, the new design methodology model should have features to address the following issues:

1) Allow designers to design in accordance to his or her preference or natural way
2) Enable traceability of minor design decisions
3) Provide tracking of design progress and direction
4) Be able to attract/encourage designer to use it
5) Facilitate the meeting of the design requirements while trying to generate the design solution

6) Facilitate the re-use of design information and design knowledge.

Horváth (2000) showed there is a link between characteristic 1 and characteristic 4. Hence, a design methodology that assists designers to design in their approach and natural way will likely be used.

In order to assist designers to design in accordance to their preference, a flexible and pro-active design methodology is needed. Quinn's study showed that designers design with uncertainty (due to incomplete information or knowledge) and enrich themselves with information and knowledge as their design progresses. This is needed as mentioned by Quinn (1980) in his "logical incrementalism" theory. In order to help designers from the context of "logical incrementalism", the tracking and traceability framework is crucial. With a tracking and traceability framework, a design methodology can provide time checking with dynamic updates on new information input and for changes of information from stakeholders throughout the design process. To enable the capture of minor design decisions, a traceable framework is important as it allows the possibility of capturing the designer's thoughts and understanding. Although, protocol studies have been developed and used to perform this, it is not a practical approach. However, a protocol studies approach is a good starting approach to analyse design more deeply, particularly as an opportunity to measure designing (Kavakli and Gero 2002). Finally, a tracking framework also helps to a designer to make better decisions when they make changes to earlier design decisions by providing a track-able link and indicators to them to foresee the effects of the changes made.

Table 2.3 Summary of differences between normative, descriptive and prescriptive design methodologies

Comparison Criteria	Normative Design Methodologies Model		Descriptive Design Methodologies Model		Prescriptive Design Methodologies Model		
	Suh Axiomatic Model	***Decision-based Model, Probabilistic Model***	***Question-based Model***	***Analysis Protocol, Interpreting Design Model, Reflective design Model, Logbook***	***Step-Oriented Model***	***Function-based Model, Issue-based Model, Design Reflection-based Model***	***Collaboration Design Model***
Basis of Orientation	Function-oriented	Step-oriented	Step-Oriented	Function-oriented	Step-oriented	Function-oriented	Social-oriented
Basis of Model	Axioms-based	Utility and probabilistic	Question-based	Recording/Capturing-based	Phases of design and reflection on design situation	Function-behaviour- based and issue-based	Social activity-based
Type of solution	Satisfice	Optimum	Satisfice	Basis for solution	Satisfice	Satisfice	Satisfice
Way of solving	Solving matrix	Aggregation of utility/probability	Posing the appropriate questions	Not applicable (For understanding/ interpretation/ recording)	Guidelines and tools	Facilitating reuse and reflection of information	Facilitating communications and negotiations
Type of Model	Non-compensatory	Compensatory	Compensatory	Compensatory	Compensatory	Compensatory	Compensatory
Main Strength	• Better focus to meet all requirements • Has traceability	• Deterministic and rational design solution	• Provide cognitive help to designers via questions	• Capture and describe the actual design process • Has limited traceability • Allows designers to design according to their preferences	• Systematic design approach • Flexible	• Capture causal aspect of design • Enable reuse of design knowledge	• Cater solving design issues for multiple designers • Enhance communications and interactions • Reduce conflict and confusions
Main Weakness	• Difficult to find design solution within time frame • Bias and flawed • Does not consider design criteria and constraints	• Allow compromises on design requirements • Need a lot of past data or parametric modelling • Difficult to quantify preferences	• Lack of relational structure • Too abstract	• Unable to support designer to solve design problems • Limited relational structure	• Without a structure • Too abstract • Not focus on meeting design requirements • Not widely used by designers	• Unable to sufficiently represent complex design problem • Limitation on taxonomical terms • Requires excessive inputs from designers • Not widely used by designers	Guidelines and tools
Preliminary target	Meeting axioms	Maximise utility value	Meeting requirements	Meeting requirements	Meeting requirements	Meeting requirements	Meeting requirements

Tracking and traceability in design methodology is also crucial for the purpose of meeting design requirements. This is because ultimately a design methodology should assist a designer in generating a design solution that meets design requirements. This may seems difficult to achieve but a design methodology should at least provide indications on how well a design process is progressing with regards to meeting design requirements. This may not help a designer in generating a "satisficing" design output in the first design iteration but it will significantly assist in reducing design iterations. Indicators and a traceable framework will allow designers to review and improve their design decisions in a systematic and well-directed way. The ability of a design methodology to provide an indicator to a designer on how well his or her design process is progressing towards meeting its design requirements will be an important basis for intelligent design. These indicators will also be able to assist a designer to review his or her knowledge and information that lead to particular design decisions. In addition to that, the indicators will also allow the designer to recognise what information is needed to enable them to make better decision. This will allow him or her to decide when to postpone a decision and how long they can delay it. Thus, a design methodology with such indicators will provide a pro-active support to a designer to assists him or her to make better design decisions. In order to achieve this pro-active support, the indicators will be developed on a dynamic model, which encourages flexibility and agility in application. Finally, such indicators will also offer the designer a visual feature to predict the design output if design changes are made. Table 2.4 summarises the results of analysis on current established design methodologies, Suh's axiomatic methodology and desired design methodology.

The literature review on design studies also noted that sketching (Schütze et al. 2003; Yang 2003) and computer-aided design (CAD) (Horváth 2000) are very important in any design process. Sketching is a crucial part of the ideation process and CAD plays a critical role in defining the physical parameters of a design outcome during embodiment and detail design. Hence, it is not surprising that further investigations found out that designers prefer methodologies that link with sketching and computer-aided design (CAD as a tool to model the physical conceptualisation of a design output) (Horváth 2000). The need to link to CAD is further strengthened by Kroes (2002), who found that there is a gap between functional conceptualisation and physical conceptualisation in design methodology.

From the findings and analysis of the current design methodologies, normative, descriptive and prescriptive design methodologies, a desired design methodology can be formulated and is as shown in Table 2.4. The importance of capturing the design information and the utilisation of design support tools in a design methodology suggested that there is a need to derive a descriptive design methodology that supports designers and is flexible enough to accommodate the differences of design approach and preference. It also is important to note the crucial differences between a design methodology for a single designer and a group of designers.

A design methodology for a collaborative design environment is more likely to facilitate the design activities related to communication and socialising. Though this is important, due to the limitations of this research time and the fact that most current established design methodologies are for a single designer environment, the scope of this research will focus on the single designer environment but will include an accountability feature in the desired descriptive design methodology that supports designers. Designers were also found to dislike the requirement to provide excessive input of information and specific data throughout the design process, especially those that are interrogative and required designers to do too much work (Reymen and Melby 2001).

Table 2.4 Results of analysis on current design methodologies and desired design methodology

Comparison Attribute	Current Established* Step-Oriented Design Methodology	Suh's Methodology**	Desired Design Methodology
Basis of Technique	• Guidelines, mathematical tools & rules	• Matrices/Mathematical tools	• Graphical and textual framework has good visualisation and links to CAD and sketch files
Basis of design decision	• Both cognitive and utility analysis	• Axioms	• Both cognitive and utility analysis with indicators on meeting design requirements
Quality Design Output	• Plausible solution	• Unique optimum solution	• Plausible solution and unique optimum solution (depending on time constraint)
Trade off* Character	• Prefer trade off	• No trade off	• Prefer trade off
Limitation	• Difficult to quantifying attributes accurately and consistently • Little traceability	• May be difficult to find solution; Biases designer • Plausible traceability	• None
Strength	• Systematic management of design	• Solution found will meet design requirements	• Systematic design approach • Solution found will meet design requirement • Good traceability and track able • Able to accommodate any design approach, preference and tools in an integrated architecture • Able to pro-actively support the designer in making design decision or delay design decision based on information available • Capture and record relevant information and data during design process • Able to determine the designer accountable for a particular design decision • Minimal input and disruption to the normal design activities of a designer
Type of decision making model	• Prescriptive design methodology	• Normative design methodology with framing	• Descriptive design methodology that is able to support designers

* The term "established" in this table refers to step-oriented design methodologies such as Pahl & Beitz (1995), Pugh (1991), Roozenburg (1995), Ullman (1997), Cross (1994), Hubka (1995), French (1971) and Ulrich (2000) (exclude Suh's axiomatic method (Suh 1990)).

** No trade-off is also known as non-compensatory which means all design requirements must be met while allow trade off (also known as compensatory) means any design requirements can be replaced by another one.

2.8 Summary

The findings suggest current established engineering design methodologies are not widely used in practice, lacking in traceability and that most of the established design methodologies are of a step-oriented type. Step-oriented design methodology does not focus on deriving design solutions from design requirements but rather searches for a design solution, which may or may not meet the design requirements at the end. Function-oriented design methodology such as Suh’s axiomatic design has better focus in meeting design requirements but enforces bias. Hence, there is a need to derive a design methodology that allows designers to design based on their preferences with a traceable and track-able framework, with some level of cognition capturing in a structured manner, able to support different design tools and linked to sketching and CAD. Finally, that design methodology should also provide indications to designers on how well their design is meeting design requirements throughout the design process. These indications will provide a basis for flexible intelligent design support development and provide intelligent assistance to designers where appropriate. It is also very important to make sure that the desired design methodology does not require a designer to provide excessive information and data. The amount of information required from the designer should be equivalent or slightly more than the information required by the existing utilisation of a design logbook.

From these findings, it is apparent that the descriptive design methodology is the design methodology that has the flexibility and pro-activeness to accommodate different designer’s approach and preference. However, the current descriptive design methodology does not provide any support to a designer. The next chapter explores the literature review on the variety of design support that can be provided to support a designer before exploring the derivation of a descriptive design methodology that is able to support the designer to design in systematic manner to meet the specified design requirements.

Chapter 3
Design Support Facilities for a Design Methodology

3.1 Overview of Design Support

Design support covers a wide domain, which generally involves any methods, tools, approaches, or frameworks that ultimately assist designers to make better design decisions directly or indirectly. This domain of research is very wide as its definition is also general and abstract. From the literature review perspective, all research work on design domain is related to design support or investigations about design. Design methodology provides the design support facilities known as a design methodology-related support in this research work. The design methodology itself provides a range of support facilities to designers. For example, Suh's axiomatic design methodology (Suh 1990) provides some kind of framing support to the designers and the axioms applied provide guidance support to the design solution. Similarly, the Pahl and Beitz design methodology (Pahl and Beitz 1995) provides guidelines as support to designers for different design phases. From the literature review of design methodologies, only normative and prescriptive design methodologies provide design support to designers. Descriptive design methodologies are usually not used for providing design support but for investigation and studying the design itself with exception of question-based methodology. Question-based design methodology provides general guidelines about how to raise appropriate questions throughout the design process. However, if the question-based methodology recommends different types of questions that should be raised for different design phases, then the methodology should be considered as a prescriptive design methodology as not all designers consider design phases in their design activity.

From the application perspective, the design support facilities can be applied throughout the design process or at specific points of design unlike a design methodology, which covers the entire design process. Therefore, these support facilities can be categorised into four types as below:

i) Design methodology-related support

ii) Computational-platform-related support

iii) Concept selection support

iv) Concept ideation support.

3.2 Design methodology-related support

This type of support is inherent in the design methodology itself. Each methodology provides some level of design support to the designers if the methodology is proposed to assist or improve the designer in designing. The facilitation of such support can be divided into several groups. These design-methodology-related support facilities groups are:

i) allow designers to record or capture their ideas and thoughts

ii) enable designers to trace and track their ideas and thoughts

iii) enable designers to decide whether to delay their design decision when information is not available; this is crucial as it might be possible to make a better decision if the relevant information were to be available (Landauer and Bellman, 2003)

iv) provide a way for designers to add, edit, or remove their design decisions any time throughout the design process

v) indicate the effects of any change in their past design decisions on past and current design decisions

vi) enable designers to reuse ideas and information recorded during the design process in future design problems with similar requirements.

These six support facilities are not exhaustive because the design-methodology-related support facilities are solely dependent on the design methodology architecture. Hence, each design methodology will have their list of design methodology-related support facilities though there will be differences for the same kind of support provided. For example, both Suh's normative design methodology and Pahl and Beitz's step-oriented prescriptive design provide designers support on managing design activities but Suh's methodology provided links between functional requirements and design parameters to support design management, which are more precise and constructive. The Pahl and Beitz methodology only provided guidelines and a systematic division of design activities into sequences of design phase.

Only normative and prescriptive design methodologies utilise design methodology-related support facilities to help designers. The descriptive methodologies though may provide some of the support facilities listed earlier but these support facilities are only used for investigation and studying on design purposes with exception of question-based design methodology.

As explained earlier, question-based design can be categorised as of descriptive or prescriptive type depending on the circumstances. In general, a question-based design methodology that is applied to a design process from design requirements to design output is a descriptive type as it does not prescribe to the designer any specific way to design. However, if the question-based methodology is applied to a design process with design phases, such as from conceptual design phase to embodiment design phase, etc. then it is considered a prescriptive type because of the introduction of design phases. This difference is illustrated in Figure 3.1.

The previous chapter has already elaborated on the differences between design methodologies and some of their design-methodology-related support facilities to a designer. Table 3.1 summarises the design-methodology-related support facilities with reference to the six facilities listed above. It is very important to note that even in a design methodology such as the step-oriented type, it is obvious that designers are allowed to edit, remove and add design decisions at any time during the design process but the methodology itself is merely guidelines and does not provide any support facilities.

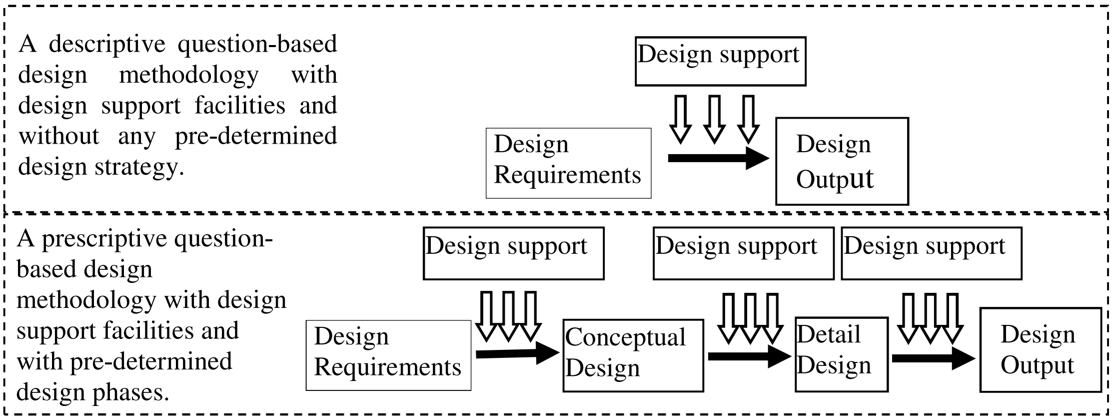


Figure 3.1 The differences between a descriptive question-based design methodology and a prescriptive question-based design methodology

Table 3.1 Summary of the design methodology-related support facilities provided by normative, descriptive and prescriptive design methodologies

Design Methodology-related Support Facilities	**Normative Design Methodologies Model**		**Descriptive Design Methodologies Model**		**Prescriptive Design Methodologies Model**		
	Suh Axiomatic Model	***Decision-based Model, Probabilistic Model***	***Question-based Model, Interpreting Design Model***	***Analysis Protocol, , Reflective design Model, Logbook Model***	***Step-Oriented Model***	***Function-based Model, Issue-based Model, Design Reflection-based Model***	***Collaboration Design Model***
Allow designers to capture their ideas and thoughts	*Limited*	*Limited*	*Nil*	*Yes*	*Nil*	*Limited*	*Limited*
Enable designers to trace and track their ideas and thoughts	*Limited*	*Nil*	*Nil*	*Limited*	*Nil*	*Limited*	*Nil*
Enable designers to decide whether to delay their design decision when information is not available	*Nil*	*Nil*	*Nil*	*Nil*	*Nil*	*Nil*	*Nil*
Allow designers to add, edit, or remove their design decisions any time during throughout the design process	*Yes*	*Yes*	*Nil*	*Limited*	*Yes*	*Yes*	*Yes*
Indicate the effects of any change in their past design decisions on current and future design decisions	*Limited*	*Nil*	*Nil*	*Nil*	*Nil*	*Nil*	*Yes*
Enable designers to reuse ideas and information recorded during the design process in future design problems with similar requirements	*Limited*	*Nil*	*Nil*	*Limited*	*Nil*	*Limited*	*Nil*

3.3 Computational-Platform-Related Support

Computational-platform-related support facilities are those that are made possible only if information technology is used. However, this statement does not mean that if information technology is used, these facilities are automatically available. These facilities are only available if the design methodology itself is well enough structured to be explored as a software tool to help designers. The availability of these support facilities is dependent on how a design methodology is structured. Some of the computational-platform-related support facilities are:

i) saving all records of ideas and requirements at any time for later use; this is necessary as design work can go on for weeks and months

ii) searching for and visualisation of the designer's design decisions at any stage during the design process

iii) providing a flexible input interface so that the designer can record his ideas through text, sketches, or graphical representation.

The list of support facilities is again not exhaustive and more facilities can be added when they are needed. In Suh's axiomatic design methodology (Suh 1990), a software tool known as "Acclaro DFSS (Design for Six Sigma)" that allows the designer to input their functional requirements and design parameters is developed and this tool also allows the designer to save their work as the design process spans over a period of time. Similarly, the reflection-based model of Reymen (2001) also provides similar facilities but different design methodologies will present a different visualisation to the designers. Unfortunately, the majority of the established prescriptive design methodologies, particularly the step-oriented type, are merely guidelines and have not developed into software tools to assist designers more effectively. For descriptive design methodology, Bender developed a software tool (Bender et al. 2002b) to ease the capturing of design activity and his tool provides a platform for the saving, observing and analysing of design activities. One of the important features of a descriptive design methodology is that, if the designer prefers it, a descriptive design methodology may be able to capture the designer-utilising additional add-on support facilities such as decision matrix to enable future reviews on how they can improve their design decision. It is apparent that the ability to capture design activities has a significant advantage from the context of scalability.

3.4 Concept Selection Support – Decision Analysis Techniques

Concept selection supports are design support facilities that are provided by models developed by researchers to assist decision making by selecting the best options. These models are also known as a decision analysis models. In design, in order to perform the selection of a solution concept, a list of solution concepts is required. These types of support facilities can only be considered or utilised after solution concepts are derived. From the step-oriented design methodologies perspective, concept selection support can only be carried out after conceptual design phase. This is important to step-oriented design methodologies as they encourage designers to explore for solution concepts and they believe that the wider the solution space explored, the better the design process is (Blessing et al. 1998). However, empirical studies also found that some experienced designers may not explore and search for a list of solution concepts in their design approach but to improvise a solution concept until it meets all the design requirements (Von der Weth 1999).

From the literature review there is a lot of research work on decision analysis models such as subjective expected utility, reason-based choice, SMART, SMARTER, TOPSIS and many more. Some of these established decision analysis models are already used by designers to make decisions, particularly in selecting the final solution concept from a list of solution concepts. From the perspective of decision-making, these techniques may also be divided into categories similar to those used for design methodologies such as normative, descriptive and prescriptive. This research work will not explore decision analysis technique in details as these techniques can be used in any design methodology as external add-on tools to assist designers to decide on which design concept is the best to be further developed in detail. It is also vital to note that some of the decision analysis models used, such as subjective expected utility, are similar to those used for a design methodology. Decision-based design methodology proposed that all design decisions made throughout the design process be quantified with values based on subjective expected utility. However, a decision analysis technique in this scenario is applied merely to provide support facilities for the process of selecting the best design concept instead of applying it to the entire design process. The distinction is illustrated in Figure 3.2. Table 3.2 summarises the different characteristics of several decision analysis models that can provide concept selection support to designers and their features such as ease of use, type, strategy, etc. Thirteen decision analysis models were presented to provide an overview the solution selection support facilities domain.

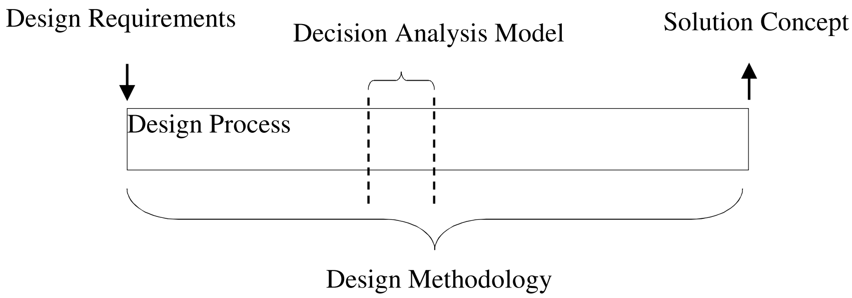


Figure 3.2 The differences of scope between a design methodology and a decision analysis models.

Though these decision analysis models provide crucial support in selecting the appropriate design by comparing the options, most of the techniques mentioned have been developed from decision-making research with some adopted by engineering designers. These models are domain-independent and can be used in any problems related to decision-making for a selection process. Different models have different characteristics. Certain models are more appropriate than others in certain circumstances. The techniques mentioned in Table 3.2 are further elaborated with brief descriptions in Table 3.3.

Table 3.2 Summary of the concept selection support facilities provided by decision analysis models

Decision Analysis Models	**Category**	**Ease to use**	**Type (Compensatory /Non-compensatory)**	**Strategy (Holistic / Non-Holistic)**	**Analytical/ Non-analytical**
Subjective Expected Utility (SEU)	Normative	Fair	Compensatory	Non-holistic	Analytical
Image Theory	Descriptive	Simple	Both	Both	Non-analytical
Recognition Primed Decision	Descriptive	Simple	Both	Holistic	Non-analytical
Reason-based choice	Descriptive	Simple	Non-compensatory	Holistic	Non-analytical
Lexicographic strategy	Descriptive	Simple	Non-compensatory	Non-holistic	Non-analytical
Elimination by aspects	Descriptive	Simple	Non-compensatory	Non-holistic	Non-analytical
Satisficing (sequential decision making)	Descriptive	Simple	Non-compensatory	Holistic	Non-analytical
Garbage Can	Descriptive	Simple	Non-compensatory	Holistic	Non-analytical
Simple multi-attribute rating technique (SMART)	Prescriptive	Simple	Compensatory	Non-holistic	Analytical
Simple multi-attribute rating technique exploiting ranks (SMARTER)	Prescriptive	Simple	Compensatory	Non-holistic	Analytical
Value-focused thinking	Prescriptive	Simple	Compensatory	Non-holistic	Analytical
Technique for order preference by similarity to an ideal solution (TOPSIS)	Prescriptive	Difficult	Compensatory	Non-holistic	Analytical
Analytic Hierarchy Process (AHP)	Prescriptive	Difficult	Compensatory	Non-holistic	Analytical

Table 3.3 Brief description of several decision analysis models

Decision Analysis Models		**Description**
Normative	***Subjective Expected Utility (SEU)*** (Savage 1954)	A mathematical approach that maximises a subjective expected utility function in the process of selecting the optimum solution. SEU differs from expected utility theory (EUT) of John von Neumann and Oskar Morgenstern (1953) where probabilities were assumed to be "objective" while SEU utilises subjective probabilities. Hence, SEU = $\sum sp_j u_i$ where *sp* is subjective utility and *u* is utility. SEU is developed based on an axiomatic basis and below are some of the axioms : • Decidability; either alternative, A_i=A_j or A_i<A_j A_i>A • Transitivity; if A_i>A_j and A_{ii}>A_{jij} , then A_i > A_{jjj} • Invariance; underlying structure is important • Independent of utility and probability; one's judgement of its future occurrence should be affected by the importance of an event. Among the weaknesses include prone to bias, irrational and tend to simplify process in decision making particularly strategic decision making (Schwenk 1984). Subjective expected utility also does no cater for trade-off and mandatory requirement.
Descriptive Decision Analysis Model	***Image Theory*** (Beach 1990)	This model is developed by Beach and Mitchell (1990; 1987a) based on the Tversky's Lexicographic model (Tversky 1972) over a period of twelve years and the Strategy Selection model (Beach and Mitchell 1978). It is a descriptive model that attempts to describe two types of decision-making: Progress Decisions, about whether past decisions are being adequately carried out and, Adoption Decisions, making decisions to replace incorrect or unachievable decisions made previously. However, the concept has a number of critics. Vlek (1987) posed a number of application limitations of Image Theory and claimed that Image Theory considers preferential decisions but seems to neglect the area of diagnostic decisions. Similarly, Montgomery also criticises Image Theory for ignoring theories of rational decision making (Montgomery 1987). As expected, Beach refuted to the two criticisms (Beach and Mitchell 1987b). Nevertheless, Beach and Strom (1989) via a laboratory study of decisions to reject or accept hypothetical jobs, proved to support the image theory prediction. Dunegan (1993) later also showed that different framing does affect decision mode, which means positive framing is associated with perceptions of compatibility between current and trajectory projected images while negative framing is linked to image incompatibility. Seidl and Traub (1998) found out that the compatibility test of image theory has consistency rates of about 15% for the editing hypothesis of the elimination of dominated choice alternatives and he recommended a method which increase the consistency rates to about 70%.
	Recognition Primed Decision (RPD) (Klein 1989)	This model is presented by Klien (1989) and shows how people use experience to avoid some limitations of analytical strategies and was developed based on observations and questionings of 150 professional decision makers. The RPD model contains four major components: recognising cases as typical, situational understanding, serial evaluation and, mental simulation that are typically employed in a sequential manner and involve revisiting and comparing previous decisions along with simulating how various options might be carried out and what their outcomes might be. The weakness of this model is that there will be a lot of different recognition model for different domain and different situation. A similar decision model known as requisite decision model (Phillips 1984) is also based on developing a model whose form and content are sufficient to solve a particular problem, which is not based on recognition aspect but constructed based on an interactive and consultative process between problem owners and specialists (decision analysts).

Table 3.3 Brief description of several decision analysis models (continued)

Decision Analysis Model		**Description**
Descriptive Decision Analysis model	***Reason-based choice*** (Shafir et al. 1993)	This model offers an alternative perspective on the way people make decisions. Based on this model, when faced with the need to choose, decision makers often seek and construct reasons to resolve conflict and justify their choice to themselves and to others. This model can lead to some unexpected violations of the principles of rotational decision making.
	Lexicographic strategy (Goodwin and Wright 2004)	This heuristic model allows decision maker to either select attributes at random or uses attributes that have been used to make the decision in the past. In some situations, the decision maker may be able to rank the attributes in order of importance. This model is non-compensatory.
	Elimination by aspects (EBA) (Goodwin and Wright 2004)	In this heuristic strategy, the most important attribute is identified and cut-off point is then established. Any alternative falling below this point is eliminated. The process continues with the second most important attribute and so on. This method is easy to apply.
	Satisficing (sequential decision making) (Goodwin and Wright 2004)	Among the oldest descriptive theory is the Satisficing model and is linked to the idea of Bounded Rationality (Simon 1982). Behaviour of organisations in learning and choice situations fall far short of the idea of "maximising" postulated in economic theory but adapt well enough to satisfice, they do not, in general, optimise.
	Garbage Can (Cohen et al. 1972)	Cohen et al., (1972) developed the Garbage Can model in response to "organised anarchies". Organised anarchies, also referred to as decision situations, are characterised by three general properties: problematic preferences, unclear technology and fluid participation (Cohen et al. 1972). In an organised anarchy, it is difficult to assign preferences to a specific decision problem because the organisation is partly consists of a loose, ill-defined group of ideas rather than a clear set of preferences and characterised by its ambiguous operating procedures and a "learn from our mistakes" philosophy. The garbage can model is fundamentally distinct from other published descriptive theories. When most decision situations arise, conventional practice is to determine the most appropriate action by whatever means. Garbage can theory states that the organised anarchy is faced with a number of choices, for which compatible problems are sought. In order to understand processes within an organisation, one can view a choice opportunity as a garbage can into which various kinds of problems and solutions are dumped by participants as they are generated. Most descriptive models do not involve optimisation instead they focus on procedural approach and the way people actually make decision.
Prescriptive Decision Analysis Model	***SMART (Simple multi-attribute rating technique)*** (Goodwin and Wright 2004)	Simple multi-attribute rating technique is a riskless, well-structured technique used to assist decision maker to make decision. SMART consists of eight main stages in its analysis(Goodwin and Wright 2004). The main stages are Stage 1: Identify the decision maker/makers Stage 2: Identify the alternative courses of action Stage 3: Identify the attributes which are relevant to the decision problem Stage 4: For each attribute, assign values to measure how well do the alternatives compare Stage 5: Assign a weight for each attribute Stage 6: For each alternative, take a weighted average of the values assigned to that alternative Stage 7: Make a provisional decision Stage 8: Undertake sensitivity analysis to observe how robust the decision is Stage 3 is done by using value tree (similar to decision tree) where the decision maker can develop links between criteria and attributes. Characteristics of a good value tree as below: • Completeness - All important attributes should be included • Operationality - The lowest level attributes can be evaluated • Decomposability - Performance on one attribute is independent from others • Absence of redundancy - e.g., no double-counting • Minimum size The characteristics of a good value tree also means that these characteristics are the weak links of SMART. SMART cannot be used if the decision is of high risk and when uncertainty is very high. Finally, decomposability of a decision can be difficult to achieve as a lot of attributes are related in the real world.

Table 3.3 Brief description of several decision analysis models (continued)

<table>
<tr><th colspan="2">Decision Analysis Model</th><th>Description</th></tr>
<tr><td rowspan="3">Prescriptive Decision Analysis Model</td><td>SMARTER (Simple multi-attribute rating technique exploiting ranks) (Edwards and Hutton 1994)</td><td>This a rather relatively simple technique to use though assignment of value functions and swing weights can still be difficult tasks. Hence, it may still lead to inaccurate reflection of the decision maker's true preferences. Edwards and Barron (1994) have suggested a simplified form of SMART known as SMARTER (SMART Exploiting Ranks). SMARTER is different from SMARTS in two ways. First, value functions are normally assumed to be linear unlike SMART instead of a curve. Hence, preliminary checks should be made to prevent poor approximation. Second, the elicitation of the swing weights is different, the decision maker need to rank the swings in order of importance. Then the decision maker uses what is known as "rank order centroid" or ROC weights to convert these rankings into a set of approximate weights. Table 1 illustrates the ROC weights.
<table>
<tr><td>Rank</td><td colspan="4">Number of attributes</td></tr>
<tr><td>1</td><td>75.0</td><td>61.1</td><td>52.1</td><td>45.7</td></tr>
<tr><td>2</td><td>25.0</td><td>27.8</td><td>27.1</td><td>25.7</td></tr>
<tr><td>3</td><td></td><td>11.1</td><td>14.6</td><td>15.7</td></tr>
<tr><td>4</td><td></td><td></td><td>6.3</td><td>9.0</td></tr>
<tr><td>5</td><td></td><td></td><td></td><td>4.0</td></tr>
</table>
Table 1: Rank order centroid (ROC) weights
Barron and Barrett (1996) has researched into the efficacy of SMARTER by assessing the efficacy associated with each of four rank-based rules – Rank order Centroid, (ROC) rank sum (RS), rank reciprocal (RR) and equal weights (EW) - in selecting a best multi attribute alternative. The results showed that ROC is the best.</td></tr>
<tr><td>Value Focused Thinking (Keeney 1992)</td><td>This model is proposed by Keeney (1992). Keeney felt that decision makers have focused too much on the choice among alternatives and that the fundamental notion in decision-making should be values. He quoted that alternatives are the means to achieve the more fundamental values. However, detail observations and review show similarity between value-focused thinking and SMART. The differences between them are alternative courses of action (stage 2) are identified prior to determining the relevant attributes (stage 3) for SMART while value-focused thinking reverse the two stages, i.e. stage 2 become stage 3 and vice versa. Goodwin (Goodwin and Wright 2004) classified value-focused thinking as a variant of SMART. Value-focus thinking initially determine your "values" which is the objectives and hence what attributes are important to the decision maker. Then the decision maker creates alternatives that might help you to achieve these objectives. This approach is to make decision makers "think outside the box". However, Wright (1999) thinks value-focused need more development before it can provide effective support for identifying these fundamental values and objectives.</td></tr>
<tr><td>TOPSIS (Technique for order preference by similarity to an ideal solution)</td><td>This technique basically chooses alternative that should have the shortest distance from the ideal solution and the farthest distance from the negative-ideal solution. This technique uses vector normalization and the normalised value could be different for different evaluation unit of a particular criterion. However the later version of TOPSIS uses linear normalisation.
TOPSIS procedure has the following steps (Opricovic and Tzeng 2004) :
1. Compute normalized decision matrix
2. Calculate the weighted normalised decision matrix
3. Identify the ideal and negative-ideal solution
4. Calculate the separation measures, using the n-dimensional Euclidean distance.
5. Calculate the relative closeness to the ideal solution.
6. Rank the preference order
The highest ranked alternative by TOPSIS is the best in terms of the ranking index, which does not mean that it is always the closest to the ideal solution.</td></tr>
</table>

Table 3.3 Brief description of several decision analysis models (continued)

<table>
<tr><th colspan="2">Decision Analysis</th><th>Description</th></tr>
<tr>
<td>Prescriptive Decision Analysis Model</td>
<td>AHP (Analytic Hierarchy Process)

(Saaty 2008)</td>
<td>
This technique was developed by Professor Thomas L. Saaty in 1970s and was widely well known as well as has many applications in a lot of areas. However, the technique is has been criticised on its axiomatic basis questioned and the extent to which it can lead to a reliable representation(Goodwin and Wright 2004). There are also lot variants of the AHP. AHP have five stages. The five stages are

Stage 1: Structure the decision hierarchy using value tree (similar to decision tree).
Stage 2: Perform pairwise comparisons of attributes and alternatives.
Stage 3: Transform the comparisons into weights and check the consistency of the decision maker's comparisons.
Stage 4: Use the weights to obtain scores for the different options and make a provisional decision.
Stage 5: Perform sensitivity analysis.

For a reasonable common problem, stage 3, 4 and 5 will require computational aid like "Expert Choice" because of the complexity of the calculations involved. In stage 2, pairwise comparisons of attributes are carried out via verbal responses. Scales of measurement are usually used in verbal responses where scale 1 would equally important or preferred while scale 9 would mean extremely more important or preferred. Table 2 illustrates the scales of measurement of AHP (Harker 1989). The first two stages are rather straight forward but stage 3 will involve conversion into a set of weights, which usually uses a mathematical approach based on eigenvalues.
<table>
<tr><th>Numerical</th><th>Definition</th></tr>
<tr><td>1</td><td>Equally important or preferred</td></tr>
<tr><td>3</td><td>Slightly more important or preferred</td></tr>
<tr><td>5</td><td>Strongly more important or preferred</td></tr>
<tr><td>7</td><td>Very strongly more important or preferred</td></tr>
<tr><td>9</td><td>Extremely important or preferred</td></tr>
<tr><td>2, 4, 6, 8</td><td>Intermediate values to reflect compromise</td></tr>
</table>
Table 2 : Scale of Measurement for AHP
Along with the weights, AHP also produces consistency index that can be calculated from Consistency index, C.I. = $\frac{\lambda_{\max} - n}{n-1}$ where $\lambda_{\max}$ is maximum eigenvalue (Perron root) of the matrix. $\lambda_{\max}$ is always greater than or equal to n for positive, reciprocal matrices and is equal to n if and only if it is a consistent matrix (Harker 1989). For each size of matrix n, random matrices were generated and their mean C.I. value, called the random index (R.I.), was computed. Using R.I. values, the consistency ratio (C.R.) is defined as the ratio of the C.I. to the R.I.; thus, C.R. is a measure of how a given matrix compares to a purely random matrix in terms of their C.I.'s. Therefore C.R. = $\frac{C.I.}{R.I.}$ A typically accepted C.R. value is less or equal to 0.1; larger values require the decision maker to reduce the inconsistencies by revising judgments but minimising inconsistency may not lead to the 'best' solution. Sensitivity analysis is a way to examine how sensitive the preferred course of action is to changes in the judgments made by the decision maker. Some of the relative strength of AHP are
1. Formal structuring of problem
2. Simplicity of pairwise comparisons
3. Redundancy allows inconsistency to be checked
4. Versatility (can be applied in wide range of applications)

The major weaknesses of AHP are
1. Conversion from verbal to numeric scale where the correspondence between the two scales is based on untested assumptions (Belton and Goodwin 1996).
2. Scaling problem of 1 to 9 where extreme ratios into decision model is bound to create inconsistencies.
3. Meaningfulness of responses to questions where weights are elicited without reference to the scales on which attributes are measured (Belton 1986).
4. The rank of existing alternatives can be reversed by new alternatives because of the AHP normalizes the weights to sum to 1 (Belton and Gear 1983).
5. Number of comparisons required maybe large because AHP allows redundancy to be built in.
Axioms of AHP are claimed to be "flawed" and the rankings which AHP produces are "arbitrary" (Dyer 1990). However, this statement is refuted by Harker and Vargas (1990) as he stressed that pair comparison must be performed on homogeneous scale.
</td>
</tr>
</table>

3.5 Concept Ideation Support

The ability of a designer to derive ideas that subsequently develops into a product or to solve a design problem whilst able to meet the predetermined design specifications is the key to the success of a design task. This ability is closely linked to the knowledge, creativity and experience of the designer. Studies have shown that there are significant differences in design activities between novice and experienced designers due to differences in their knowledge and experience (Ahmed and Wallace 2004; Ahmed et al. 2003; Ho 2001; Kavakli and Gero 2002; Liikkanen and Perttula 2009) . For novice designers, the need for concept ideation support facilities is crucial as empirical research work showed that novice designers are unaware of design strategies (Ahmed et al. 2003) and often not able to decompose design problems efficiently (Ho 2001). The limited knowledge and experience of a novice designer hindered their effectiveness in deriving ideas to solve design problems (Ahmed and Wallace 2004). All current engineering design methodologies play a minimal role in assisting the designer to produce these ideas. The importance of assisting designers to generate solution ideas or ideation support is critical, especially for novice designers. In addition to that, there is also a need to look into ways to provide an integrated concept ideation support system within an engineering design methodology framework to help designers to generate design ideas and solutions more effectively.

There are two types of approaches to the deriving of ideas to solve design problems, the cognitive-based design approach and the generative design approach. There are distinct differences between the two approaches. The cognitive-based design approach is widely applied and is solely dependent on the creativity, knowledge and experience of the designer. The generative design approach is to apply computers to generate solution ideas but the approach is dependent on how the design parameters of the current solution are modelled and on having an existing design solution. For the generative design approach, the need to have an existing design solution before it can be applied implies that the approach merely evolves the current design solution, often with some restrictions, to generate new design solutions. The process of evolving the current design solution is usually performed with a search algorithm on the design parameters such as physical dimensions. Therefore, generative design approaches are domain specific. These two ideation approaches will be explored next to determine their differences, strength and deficiencies.

3.5.1 Cognitive-based design ideation approach

The notion of providing concept ideation support to designers is not new and cognitive-based design ideation approaches has been widely used by designers for a long time. The cognitive-based design ideation approach is also known as the creative problem solving approach. These approaches provide guidelines and ways to simulate the thoughts of designers to enable them to "think out of the box" or explore from different perspectives. Some of these techniques, such as brainstorming (Rawlinson 1981) and Delphi method (Linstone and Turoff 1975), encourage group activities among designers to stimulate the derivation of solution concepts while other techniques such as lateral thinking (de Bono 1977), mind mapping (Buzan 2005), creativity template (Goldenberg and Mazursky 2002), TRIZ or "Theory of Inventive Problem Solving" (Altshuller 1997; Mann 2002), Synectics (SYN) (Gordon 1961), and morphological analysis (Fargnoli et al. 2006) to provoke the thoughts of designers to explore for solution. These cognitive-based design ideation approaches are divided into two groups (Pham and Liu 2006), disciplined thinking methods and divergent thinking methods.

According to Pham and Liu (2006), disciplined thinking methods such as morphological analysis and creativity template are methods that depend on a logical structure to derive new solution concepts. Divergent thinking methods such as TRIZ, lateral thinking and mind mapping are able to create completely new solution concepts based on breaking the "psychological inertia". "Psychological inertia" is defined as a strong preference towards conventional or usual ways of solving design problems. In order to derive new ways to solve a particular design problem, it is important to break the effects of "psychological inertia" (Mann 2002). Though TRIZ, lateral thinking, and mind mapping are divergent thinking methods, TRIZ, unlike the others, was derived from a vast knowledge base of patents and will be further elaborated in the next section.

3.5.1.1 TRIZ (Theory of Inventive Problem Solving)

TRIZ or "Theory of inventive problem solving" was created by Genrikh Saulovich Altshuller (Orloff 2006) after years of studying design patents in the context of generic features and inventive principles. TRIZ is also known as Teoriya Resheniya Izobreatatelskikh Zadatch and one of TRIZ tools, the technical contradiction matrix, was created with twin aims; inventions are created to solve technical contradictions and conflicts emerge from the inconsistent individual component development in technical systems (Mann 2002).

The classical technical contradiction matrix of TRIZ is a matrix having 39 improving features and 39 worsening features. According to Altshuller's TRIZ problem-solving method, the designer is required to identify a list of improving features and worsening features from the technical contradiction matrix. The cell that coincides with each improving feature and worsening feature will have a list of inventive principles or solutions. This list is restricted to a maximum number of four possible inventive principles in the conventional matrix. There are a total of 40 inventive principles that can be used to solve all the design problems based on this contradiction matrix. However, there are two weaknesses with this matrix. Some of the cells are empty i.e. there are no recommended inventive principles and the cells that coincide with the same improving feature and worsening feature are always empty and have no recommendation as to inventive principle. Table 3.4 shows the classical technical contradiction matrix in a schematic table consisting of 39 improving and worsening features as the entire matrix is too large to be shown in this thesis. The complete classical matrix is available in the book by Mann (2002).

Table 3.5 illustrates the 40 inventive principles proposed by TRIZ depending on the contradicting features.

Table 3.4 Schematic table of the classical TRIZ contradiction matrix (the numbers in italic are numbers representing inventive principles adapted from Mann (2002))

Worsening Feature → / Improving Feature		39: Productivity
1: Weight of moving object	. . . .	35 3 24 37
2: Weight of stationary	. . . .	1 28 15 35
3: Length of moving object	. . . .	14 4 28 29
4: Length of stationary	. . . .	30 14 7 26
5: Area of moving object	. . . .	10 26 34 2
6: Area of stationary	. . . .	10 15 17 7
7: Volume of moving object	. . . .	10 6 2 34
8: Volume of stationary	. . . .	35 37 10 2
9: Speed	. . . .	-
10: Force (Intensity)	. . . .	3 28 35 37
11: Stress or pressure	. . . .	10 14 35 37
12: Shape	. . . .	17 26 34 10
13: Stability of the object	. . . .	23 35 40 3
14: Strength	. . . .	29 35 10 14
15: Durability of moving obj.	. . . .	35 17 14 19
16: Durability of non-moving obj.	. . . .	20 10 16 38
17: Temperature	. . . .	15 28 35
18: Illumination intensity	. . . .	2 25 16
19: Use of energy by moving	. . . .	12 28 35
20: Use of energy by stationary	. . . .	1 6
21: Power	. . . .	28 35 34
22: Loss of Energy	. . . .	28 10 29 35
23: Loss of substance	. . . .	28 35 10 23
24: Loss of Information	. . . .	13 23 15
25: Loss of Time	. . . .	-
26: Quantity of substance/the	. . . .	13 29 3 27
27: Reliability	. . . .	1 35 29 38
28: Measurement accuracy	. . . .	10 34 28 32
29: Manufacturing precision	. . . .	10 18 32 39
30: Object-affected harmful	. . . .	22 35 13 24
31: Object-generated harmful	. . . .	22 35 18 39
32: Ease of manufacture	. . . .	35 1 10 28
33: Ease of operation	. . . .	15 1 28
34: Ease of repair	. . . .	1 32 10
35: Adaptability or versatility	. . . .	35 28 6 37
36: Device complexity	. . . .	12 17 28
37: Difficulty of detecting	. . . .	35 18
38: Extent of automation	. . . .	5 12 35 26
39: Productivity	. . . .	*

Table 3.5 The 40 inventive principles of TRIZ (Altshuller 1997)

Inventive Principles
1. Segmentation
2. Taking Out
3. Local Quality
4. Asymmetry
5. Merging
6. Universality
7. "Nested Doll"
8. Anti-Weight
9. Preliminary Anti-Action
10. Preliminary Action
11. Beforehand Cushioning
12. Equipotentiality
13. "The other way round"
14. Spheroidality - Curvature
15. Dynamisation
16. Partial or Excessive Actions
17. Another Dimension
18. Mechanical Vibration
19. Periodic Action
20. Continuity of Useful Action
21. Skipping
22. "Blessing in Disguise"
23. Feedback
24. "Intermediary"
25. Self-Service
26. Copying
27. Cheap Short-Living Objects
28. Mechanics Substitution
29. Pneumatics and Hydraulics
30. Flexible Shells and Thin Films
31. Porous Materials
32. Colour Changes
33. Homogeneity
34. Discarding and Recovering
35. Parameter Changes
36. Phase Transitions
37. Thermal Expansion
38. Strong Oxidants
39. Inert Atmosphere
40. Composite Materials

The research work on classical TRIZ was led by Altshuller and was completed in 1985. Since then a number of variants of TRIZ have been derived in this domain. Algorithm of Inventive Problems Solving (ARIZ) (Fey and Rivin 2005), Unified Structured Innovative Thinking (USIT) (Nakagawa et al. 2002), and Systematic Inventive Thinking (SIT) (Horowitz and Maimon 1997) are a few of the variants of TRIZ found in the literature.

There are several well-known issues with the application of TRIZ. One of the common ones is that the inventive principles are poorly defined with general terminology, allowing only very abstract interpretation of the inventive principles. Another common issue of TRIZ is that most design problems at a high level pose a large list of improving and worsening features. Using the technical contradiction matrix, as large amount of inventive principles would be recommended, it is better to apply TRIZ at the root level or to carry out a "root contradiction" analysis (Mann 2002). Hence, all the variants of TRIZ either reduce or increase the number of contradicting features or inventive principles. Also they provide manual algorithmic guidelines or a step by step reduction approach to solve design problems at root level. However, these variants of TRIZ are found to be either too simple or too difficult to be used by designers (Pham and Liu 2009).

The application of a technical contradiction matrix to solving design problems is not new. This is because most design problems involve deriving solutions that solve one or more contradiction features or factors. Mann (2002), Savransky (2001), Fey and Rivin (2005), Rantenen and Domb (2008), Markus (2011) and many others have shown a variety of ways to use the technical contradiction matrix to solve design problems. Consistent with what Altshuller suggested, all TRIZ problem-solving tools, including the contradiction matrix, advocate that the ultimate aim of solving a design problem is to achieve the ideal result. The ideal result is defined as a design solution that has all the useful functions wanted and has no harmful functions or weaknesses (Mann 2002). Such advocacy is intended to help designers to break "psychological inertia" that would hinder the chances of deriving an innovative design solution. However, the difficulty in applying the technical contradiction matrix of TRIZ is apparent. To apply the technical contradiction matrix, a designer needs to identify at least one or more of the improving features and the worsening features related to his design problem. The task of identifying improving features and worsening features related to a design problem is usually not straightforward. For some problems, this task is not an easy one because different designers may not select the same list of improving and worsening features for the same design problem. A different list of improving and worsening features may lead to different inventive principles recommended for the design solution. Therefore, the difficulty in using the contradiction matrix of TRIZ can be attributed to the difficulty in translating the requirements, constraints or criteria of design problems to the appropriate improving and worsening features. The guidelines for this are to select the nearest or try to match these requirements to the best-related features.

From the perspective of inventive solutions, the recommended inventive principles are very abstract and general. Inventive principles such as "The other way round", "Blessing in disguise" and "Preliminary action" are a few of the inventive principles that are ambiguous and can be interpreted differently by different designers. For a novice designer, these inventive principles may not provide any help for them to get nearer to the design solution. The current method of TRIZ in helping designers is to provide a few examples of design solutions related to the respective inventive principle. For example, a double sequential flash performed by a camera in capturing a photograph is the design solution to reduce the "red-eye" effect in photography and this design solution is related to the "periodic action" inventive principle (Mann 2002). Though these examples are important in providing help to designers with some ideas of the possible ways of solving a design problem in relation to the recommended inventive principles, the effectiveness of such help is limited. There was some research on TRIZ to explore the possibility of improving the definition of features and inventive principles into more specific attributes to help designers better. Pham and Liu (2009) derive a symbol representation of TRIZ based on I-Ching concept to describe TRIZ improving and worsening features and expand the inventive principles into more specific attributes.

Another issue with the classical matrix is the recommendation of inventive principles. The designers that apply the technical contradiction matrix of TRIZ to help them to solve design problems are solely dependent on the recommended inventive principles that TRIZ proposes. However, looking at the classical technical contradiction matrix of TRIZ, the number cells without recommended inventive principles are 275 or 18.08% of the matrix. Therefore, there is almost one in five of a chance that TRIZ cannot help a designer to solve design problems at all. Since the research work by Altshuller was completed in 1985, a huge amount of new patents have been granted and hence the deployment of these 40 inventive principles has been changed with relation to the contradicting features (Mann et al. 2003). In view of this, Mann (2003) developed a new contradiction matrix which has 48 improving features and 48 worsening features but with the same number of inventive principles (40). The new matrix still offers no recommendation of any inventive principles for the case when the same improving feature and worsening feature coincide but all the other cells have recommended inventive principles, unlike the classical one. This is important as the new matrix was developed based on updated information from the patents. The new matrix has only 48 empty cells out of a total of 2304 cells or 2.08% empty cells (only diagonal cells are without the recommendation of inventive principles). Table 3.6 illustrates all the 48 improving and worsening features of the new matrix by Mann (2003). The new matrix provides significantly more design–related knowledge and better design support compared to the classical TRIZ. The new contradiction matrix is then further updated with the improving and worsening features increased to 50 from 48 but with the inventive principles retained at 40 (Mann 2009). The two new addition features are the positive intangibles feature and the negative intangibles feature. Table 3.7 shows the new 50 improving and worsening features of the new contradiction matrix.

Table 3.6 The 48 improving and worsening features of TRIZ contradiction matrix by Mann (reproduced from (Mann et al. 2003))

Improving and Worsening Feature
1: Weight of moving object
2: Weight of stationary object
3: Length/Angle of moving object
4: Length/Angle of stationary object
5: Area of moving object
6: Area of stationary object
7: Volume of moving object
8: Volume of stationary object
9: Shape
10: Amount of substance
11: Amount of information
12: Duration of action of moving object
13: Duration of action of stationary object
14: Speed
15: Force/Torque
16: Energy used by moving object
17: Energy used by stationary object
18: Power
19: Stress/Pressure
20: Strength
21: Stability of the object
22: Temperature
23: Illumination intensity
24: Function Efficiency
25: Loss of Substance
26: Loss of Time
27: Loss of Energy
28: Loss of Information
29: Noise
30: Harmful Emission
31: Other harmful effects generated by system
32: Adaptability/versatility
33: Compatibility/Connectivity
34: Trainability/Operability/Controllability/Ease of operation
35: Reliability/Robustness
36: Reparability / Ease of repair
37: Security
38: Safety/Vulnerability
39: Aesthetics/Appearance
40: Other harmful effects acting on system
41: Manufacturability/Ease of manufacture
42: Manufacturing precision/Consistency
43: Automation/Extent of automation
44: Productivity
45: Device complexity
46: Control Complexity
47: Ability to detect/Measure/Difficulty of detecting
48: Measurement accuracy/Measuring Precision

Table 3.7 The 50 improving and worsening features of TRIZ contradiction matrix by Mann (reproduced from (Mann 2009))

Improving and Worsening Feature
1: Weight of moving object
2: Weight of stationary object
3: Length/Angle of moving object
4: Length/Angle of stationary object
5: Area of moving object
6: Area of stationary object
7: Volume of moving object
8: Volume of stationary object
9: Shape
10: Amount of substance
11: Amount of information
12: Duration of action of moving object
13: Duration of action of stationary object
14: Speed
15: Force/Torque
16: Energy used by moving object
17: Energy used by stationary object
18: Power
19: Stress/Pressure
20: Strength
21: Stability of the object
22: Temperature
23: Illumination intensity
24: Function Efficiency
25: Loss of Substance
26: Loss of Time
27: Loss of Energy
28: Loss of Information
29: Noise
30: Harmful Emission
31: Other harmful effects generated by system
32: Adaptability/versatility
33: Compatibility/Connectivity
34: Trainability/Operability/Controllability/Ease of operation
35: Reliability/Robustness
36: Reparability / Ease of repair
37: Security
38: Safety/Vulnerability
39: Aesthetics/Appearance
40: Other harmful effects acting on system
41: Manufacturability/Ease of manufacture
42: Manufacturing precision/Consistency
43: Automation/Extent of automation
44: Productivity
45: Device complexity
46: Control Complexity
47: Positive Intangibles
48: Negative Intangibles
49: Ability to detect/Measure/Difficulty of detecting
50: Measurement accuracy/Measuring Precision

The new improved technical matrix contradiction also has grouped the improving and worsening features into six groups to facilitate some general guidelines to help the designers to identify the relevant improving or worsening features related to their design problems. The six groups are physical, performance, efficiency, “itility”, manufacturing/cost and measurement feature groups. The “Itility” feature group implicates the features related to the design for X features such as adaptability, compatibility, controllability, reliability, and others. In addition to that, the new technical contradiction matrix has different recommended inventive principles for some corresponding improving and worsening features because of the changes in inventive trends within the last two decades of new design patents.

3.5.1.2 Brainstorming

This cognitive-based technique is a very popular group technique to generate ideas. This technique has been used by many people, including designers, for many decades. There are many versions of this technique but generally, this technique is applied in a group with a facilitator. The initial aim of using this technique is to create a list of central questions pertaining to the problem that needs to be solved. Then the next phase is to generate ideas about solving the problem and then critically examine the ideas generated. Orloff (2006) summarises the brainstorming methods and its strength and deficiencies in Figure 3.3.

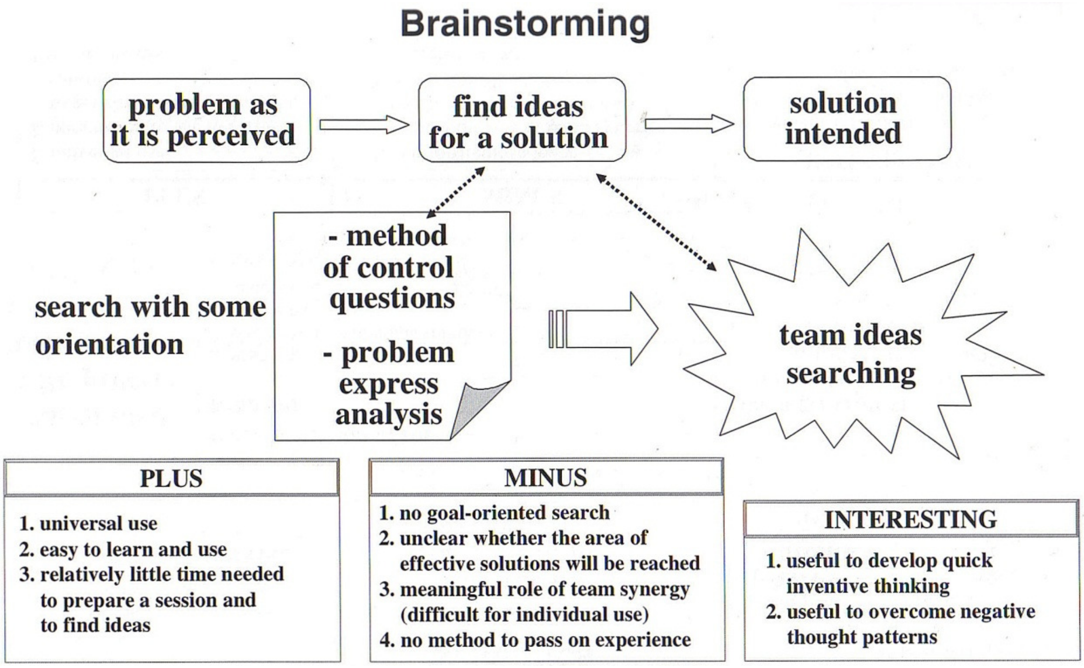


Figure 3.3 The brainstorming method with its characteristics (reproduced from (Orloff 2006))

3.5.1.3 Lateral Thinking

Lateral thinking is a very well-known cognitive-based approach to solving problems and this method has been developed by de Bono (1977). The basis of this approach is to encourage a designer to provoke his own thinking or “think out of the box” to solve a design problem. In order to “think out of the box”, de Bono has proposed several lateral thinking techniques such as “six thinking hat”, simple focus, challenge, alternatives, suspended judgement, etc. (Rosenbaum 2001) though six thinking hat is related to parallel thinking. Designers are encouraged to assume the different roles from different perspectives and by looking into a design problem from different perspectives, the mind of the designer will explore wider rather than deeper. This will inevitably improve the chances of creating ideas that solve a design problem. One of the criticisms about the lateral thinking approach is that it lacks structure and organisation but is rather merely a set of tools and techniques to encourage designers to think differently in solving design problems (González 2001).

3.5.1.4 Morphological Analysis

Morphological analysis has been one of the common approaches used by designers for many years. Morphological analysis is performed by creating a table or a matrix where the relevant features of a product that needs to be designed are tabulated on the matrix and the designers morph or evolve the features and the parts to form the final design that meets the design requirements. Though designers usually used morphological analysis manually, Belaziz et al. (2000) utilised computational tool that integrate morphological analysis during the design process to morph form features to form new products. Figure 3.4 illustrates the concept of the morphological analysis method and its characteristics. One of the main issues of morphological analysis is the difficulty it has in dealing with incompatibility between parts or features that may occur during the during the morphing process to form new design solutions.

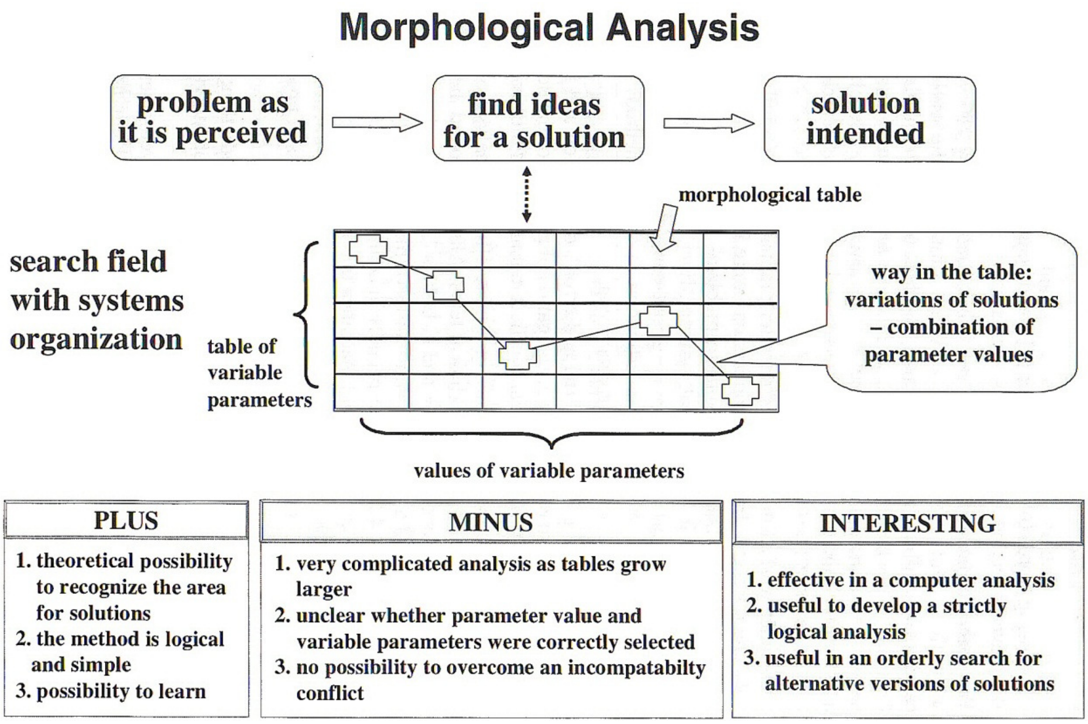


Figure 3.4 The morphological analysis method with its characteristics (reproduced from (Orloff 2006))

3.5.1.5 Delphi Method

The Delphi method was developed by the RAND Corporation as a forecasting tool (Murray 1979) and has been widely used to predict a variety of future events. It was later expanded to develop goals as well as for problem-solving purposes. This method is a systematic group decision process utilising a group of experts and based on questionnaires to create ideas to solve problems. The generic flow chart for the implementation of the Delphi method is as shown in Figure 3.5. The flow chart clearly shows that the Delphi method exploits the knowledge and the experience of expert panels to generate ideas to answer those the questionnaires raised. These ideas are then evaluated and analysed. The experts are expected to revise their ideas after evaluation and the analysis of their earlier ideas if there is no final consensus.

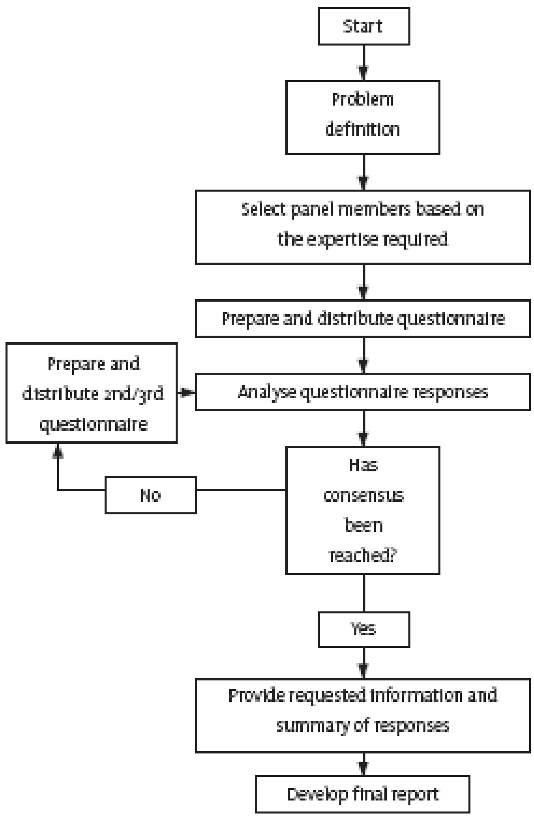


Figure 3.5 The Delphi method flow chart (reproduced from (Slocum 2005))

3.5.1.6 Synectics

Synectics is another cognitive–based technique developed by Gordon (Orloff 2006). This technique is similar to brainstorming but more sophisticated (Nolan 2003). Unlike brainstorming and lateral thinking, Synectics emphasises the role of the metaphorical process in generating creative solutions and the process views the initial solutions generated as "springboards" which then further encourage the process to continue to generate solutions throughout the developmental judgement phase. All ideas are evaluated positively. Hence, it has direction and there will be iteration between ideas and their constructive evaluation as the process moves towards a course of action. Synectics process is more difficult to learn when compared to others. The Synectics process and its characteristics are briefly described in Figure 3.6 below.

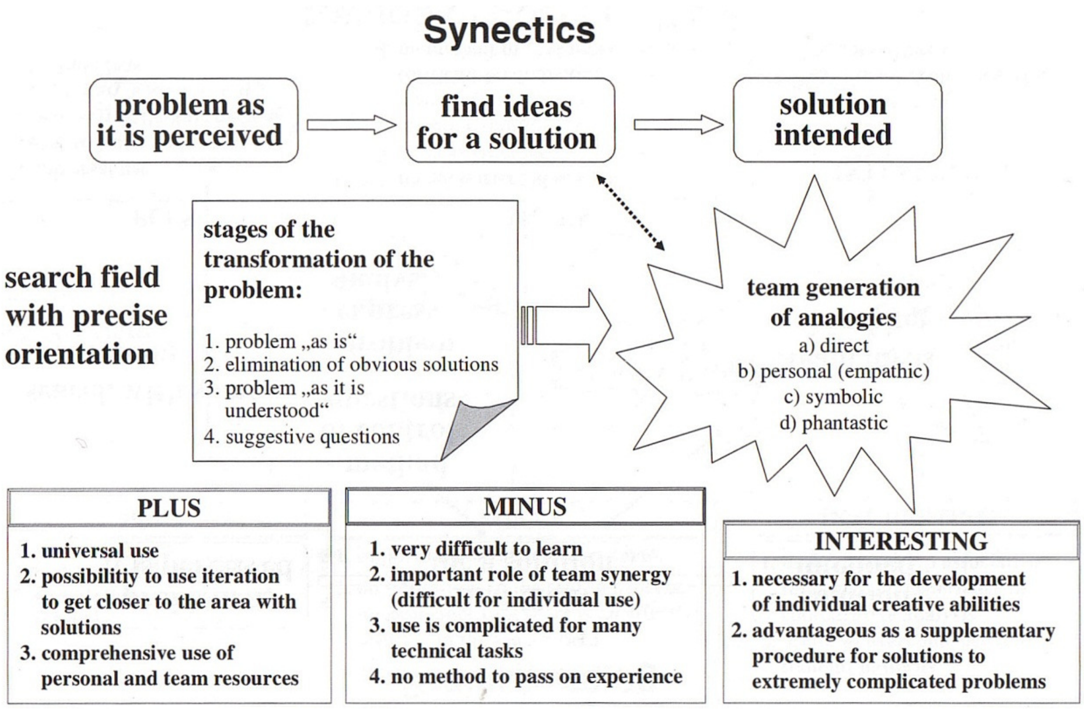


Figure 3.6 The Synectics method with its characteristics (reproduced from (Orloff 2006))

3.5.1.7 Creativity Templates

Creativity templates are methods to generate ideation developed by Goldenberg and Mazursky (2002) and were initially identified using mapping research, a backward analysis of product innovations. There are five creativity templates derived to support ideation - attribute dependency, replacement, displacement, division and component control with the attribute dependency template as the dominant template (Goldenberg et al. 1999b). The templates are a sequence of formal operations on the initial structure of a system (Goldenberg et al. 1999a) derived from six elementary (first principle) operators (Goldenberg et al. 1999b). The six operators are explained in Figure 3.7 (Goldenberg et al. 1999b).

Operator	*Definition*	*Illustration*
(1) Exclusion	The exclusion operator removes an unlinked component from the configuration boundaries.	
(2) Inclusion	The inclusion operator introduces an external component into the configuration boundaries.	
(3) Unlinking	An unlinking operator eliminates a link.	
(4) Linking	A linking operator connects two unlinked components or attributes.	
(5) Splitting	A splitting operator removes an internal component from the link. The link maintains the original functions.	
(6) Joining	A joining operator adds a (new) component to a *dangling link*.	

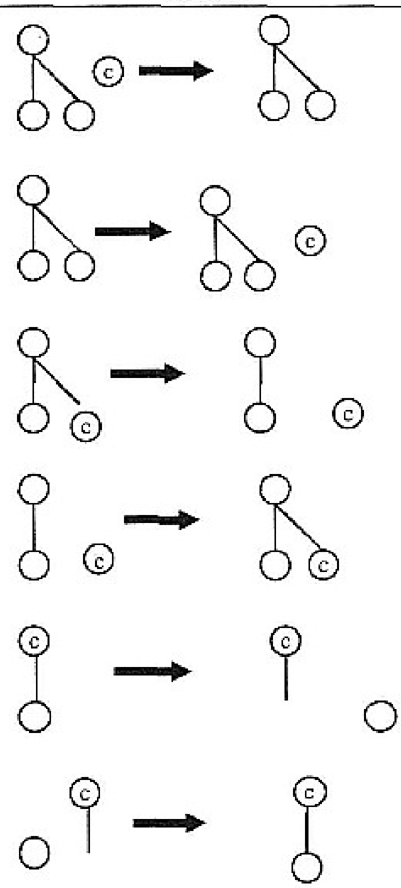


Figure 3.7 The operators involved in the templates (reproduced from (Goldenberg et al. 1999b)

The attribute dependency template is a template that is obtained by applying inclusion and linking operators sequentially and this attribute dependency template operates in the context of product attributes. The next four templates, namely, the component control, replacement, displacement and division templates operate in a product components context. The detailed descriptions of these four templates are presented in Figure 3.8 with explanation and examples. Though templates are ideation methods, they require an existing product as a reference (Goldenberg et al. 1999b).

Template	*Description*	*Example*	*Sequence of Operators*
Component Control	The template involves the creation of a link in the form of control of one internal component over another internal or external component.	A new electronic device connects the battery of a car to the car body to inhibit corrosion and rust. The control is obtained by providing an excess of electrons to the cathode, thus enabling regulation of the electrostatic charge, because positive charge hinders electrochemical corrosion.	Inclusion and linking
Replacement	Application of this template involves the removal of an essential internal component from the configuration while maintaining the link between the removed component and the remaining components. This operation creates a temporarily inconsistent abstract structure. Because of the dangling link, the operation is completed only when the missing component is replaced by another existing component. The replacement must be an external component that can perform a function similar to that provided by the one removed.	Consider a car radio. The internal component, in this case the car antenna, is removed, but its associated intrinsic function (reception of broadcast waves) is maintained. The resultant intermediate configuration is a necessary step in the replacement procedure, even though it represents an incomplete product structure. The unsaturated function can be fulfilled by a component that is external to the car radio, in this case a defroster. Finally, the external component is incorporated by applying the joining operator, and the configuration of a new product is obtained—a car radio that does not require an external antenna.	Splitting, excluding, including, and joining
Displacement	An essential internal component is removed from the configuration. However, in contrast to the Replacement template, its associated link is removed as well. In this case, a new idea for the product must be based on a new appeal, one that the former product did not provide.	Excluding the car roof and its function, and the new product is a convertible car.[a]	Splitting, excluding, and unlinking
Division	Splitting one component into several components that either contribute individually to the accomplishment of its function or become responsible for differential subfunctions.	Dividing a shock-absorbing system into a four-way suspension to improve smooth driving and balance.	Splitting and linking

[a]Some convertible cars have a folding roof, whereas in others, the roof is attachable but not an integral part of the car. The latter is a case of displacement.

Figure 3.8 The other four templates: component control, replacement, displacement and division templates (reproduced from Goldenberg et. al. (1999b)

3.5.2 Generative design ideation approach

Generative design is a very actively researched domain with a large amount of on-going research work. A generative design ideation approach is "a method that generates product concepts based on a set of input specifications" (Eckert et al.

1999). Hence, the key factor that enables the utilisation of this method is a set of input specifications based on an existing or current product. Among the popular methods to generate product concepts based a set of input specifications are the application of shape grammars (Agarwal and Cagan 1998; McCormack et al. 2004) with optimisation techniques such as genetic algorithm (Bentley 2000; Case et al. 2004; Graham et al. 2001), simulated annealing (Shea et al. 1997) and the bees algorithm (Pham et al. 2008). These generative design ideation approaches utilise the series of random changes of a set of input specifications to explore the solution space. The changes of the set of input specifications can be done manually or automatically using a programme and each change will create a new solution concept. Each new solution concept generated will be evaluated against a set of pre-defined constraints. The schematic representation of a generative design ideation approach is shown in Figure 3.9.

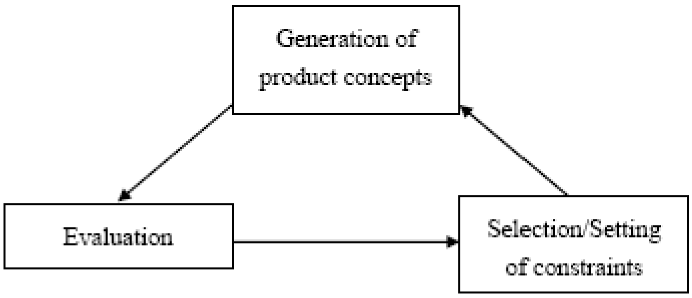


Figure 3.9 The schematic representation of a generative design ideation system (reproduced from (Eckert et al. 1999))

3.6 Analysis of Design Support Facilities for a Design Methodology

The design support facilities are widely applied by designers and these facilities play a vital role in helping designers to solve design problems. Out of the four types of design support facilities, namely design methodology-related support, computational-platform-related support, concept selection support and concept ideation support, only the design methodology-related support for descriptive design methodologies and the computational-platform-related support are basic support facilities that can be provided to a designer without interfering with the designer's preferences in their approach. The concept selection and concept ideation support facilities are consist with methods and tools that can be applied at a particular point in the design process to solve a specific design problem but may not be preferred by the designers. Design methodology-related support facilities for normative design and prescriptive design methodologies impose specific ways to solve design problems and hence interfere with the designer's preferences.

In addition to that, some support facilities are dependent on the design methodology itself. Not all design methodology is able to provide a computational-platform-related support because some design methodologies are just guidelines. Even though they are just guidelines, these design methodologies do provide some basis of design methodology-related support. As for the computational support facilities, they are not only free from interfering with the designer's preferences, they are also vital for improving the effectiveness of the design process and necessary to enable further integration with the application of the computer-aided design during the detail design phase. Among the design methodologies, only the descriptive design methodologies do not interfere with a designer's preferences in design approach but the current descriptive design methodology-related support facilities are only utilised to assist in studying how designers design. Hence, there is a need to derive a descriptive design methodology that supports designers.

The other two design support facilities, namely, concept selection support and concept ideation support are independent of the design methodology itself. This means that these two support facilities can work as stand-alone support facilities and operate individually without any design methodology. Hence, they are optional design support facilities. Optional design support facilities are support facilities that may or may not be used by a designer in the process of design depending on the designer's preferences. Moreover, the current design methodologies are unable to support these facilities in comprehensive manner because of the following four deficiencies:

1. Lack of relationship linkage - some of the design methodologies are general guidelines and have difficulty in integrating with the other tools via computerisation because computerisation requires a specific relationship between design data from conceptual design to the detail design phases.
2. Lack of information capturing for important design ideas and decisions which can provide the information to be processed downstream whilst minimising interruption to the design process.
3. Inability to accommodate different type of design information as design information can come in the form of sketches and texts.
4. Inability to provide graphical visualisation of the design process to improve the direction and the flow of the on-going design process.
5. Current design methodologies are not able to support a broad range of design support tools to facilitate the design process in tandem and synergistic way.

The findings from the literature review on various design support facilities and design methodologies concurred with the findings of the National Research Council of the United States of America (NRC 1991). The report from NRC on approaches to improve engineering design that described design support tools as valuable but each support tool has its strength and specific focus of application but working in isolation i.e. "no one tool can do it all". The current design methodologies encourage designers to utilise various support tools but have little role in linking and supporting these tools within the design process to enable to designers to utilise them in synergy.

Although the concept selection support and concept ideation support is independent of the design methodology itself, a design methodology should also include features that assist and integrate the utilisation of these tools in a more effective manner. This is because, unlike experienced designers, novice designers do need help in making better design decisions in concept selection and in deriving new solution concepts.

The final decision to seek the assistance of concept selection tools and concept ideation tools are dependent on the designers and a design methodology should have this flexibility.

From the knowledge perspective, there has been ample research work on expert systems and on the capturing and re-use of design knowledge to assist designers in concept selection and concept ideation. Most expert systems and systems that re-use design knowledge such as design rationale capturing systems are inherently domain-specific systems or systems that are restricted to a limited pool of knowledge. The design knowledge acquired and re-used is limited for solving problems in a narrow engineering domain. In addition to that, another problem with existing design rationale capturing systems is that, although the amount of knowledge captured may be limited, the volume of information captured is very large and difficult to process meaningfully. This means it can take a long time to accumulate sufficient broadly applicable knowledge.

Unlike the conventional expert systems and design rationale capturing systems, TRIZ inventive principles are derived from a good source of design knowledge and expertise such as the patent office that has a vast collection of ideas and solutions for design problems from a broad range of domains. The TRIZ contradiction matrix was developed based on knowledge from design patents (Altshuller 1997; Mann et al. 2003) and is the only tool that is scientifically and systematically developed to overcome “psychological inertia”. Though the TRIZ contradiction matrix was developed based on knowledge from design patents (a vast pool of design ideas and knowledge), the user needs to decompose design problems into a lower level in order to be able to utilise the tool effectively. Due to this issue, ARIZ was developed to assist designers manually to decompose design problems systematically into lower levels so that the TRIZ contradiction matrix could be utilised more effectively. The process of decomposing design problems into lower levels is crucial to the effective use of TRIZ.

With the type design methodologies reviewed and analysed as well as the type of design support facilities involved, it is apparent that a new design methodology that can capture the thoughts of designers as they design are needed. With the thoughts and ideas of designers captured during the design process, this design methodology should also provide a basis for providing a range of design support facilities to improve their design decisions about the design solution can be created. It is also vital that a design methodology is able to provide support facilities to designers without affecting their preference or approach. This is to allow designers to design with the approach or strategies that they are comfortable and successful with while retaining the ability to obtain the essential support that improves on the limitations affecting all designers. Finally, any design methodology should provide the support of concept ideation facilities to designers, in particular to novice designers should the designers wish to utilise them. These optional support facilities enable a more integrated and flexible design methodology to assist the designers in designing better.

The findings from the literature review on design support facilities are tabulated in Table 3.7. The support facilities could be utilised by the designer to assist them in design. Knowing the strength and deficiencies of these facilities is important to the designers so they can utilise these facilities more effectively. For this research work, the analysis of the design support facilities provide key features that a design methodology should support so that such facilities can be incorporated effectively into a design methodology.

With the conclusions of the literature review on design methodologies and design support facilities that point to the need to have a design methodology that is able to accommodate different design strategy and design approach and yet able to provide specialist support when needed, the next chapter will describe the conceptualisation of such a design methodology.

3.7 Summary

The findings on the design support facilities showed that some could help designers without influencing their decisions and affecting their approaches and strategies. Computational-based and design methodologies-based support facilities, such as the descriptive design type, are among those support facilities that are independent of the designer's approaches and strategies. The study also found that the support facilities to select the best solution concept and the ideation support facilities affect the designers' decisions and their approaches.

Table 3.8 A summary of comparisons between different design tools and their support facilities to help the designer.

Type of design support	Support Facilities	Advantages of the support facilities	Deficiencies	Effects on designers' approach/strategies
Design methodology-related support	Capture ideas and thoughts.	Able to retain knowledge and ideas for reuse in the future.	May require tedious efforts.	Only descriptive design methodologies have no effects on the designer's approaches or strategies
	Trace and track ideas, thoughts and design decisions.	Able to predict the direction and foresee the effects of design change.	The trace and track can be very complicated and extensive to be displayed for huge design projects.	
	Decisions on delaying making design decision until the relevant information are available.	Allow designers to make decisions more effectively when the relevant information is available after the delay.	The relevant information can be very a lot and may not be sufficiently represented.	
	Edit/Delete/Add design decision.	Able to make decision change during the design.	Inter-related design decisions need to be considered.	
	Indicate effects of change of past design decisions.	Allow to designer to visualise effects of any decision change he made upstream and downstream of the design flow.	Visualisation maybe difficult if a lot of changes are involved.	
	Reuse past ideas and information.	To enable utilisation past knowledge and experience to solve existing design problems.	Effective of re-utilisation of knowledge is dependent on how well the key parameters of knowledge are identified and classified.	
Computational-platform-related support	Ability to save design data.	To allow designers to store and recall design data during the design process that may last several	Massive amount of storage maybe be needed to store information for a large design project.	No
	Ability to search and visualise design decisions.	To enable an overview of the direction and the review of design process.	The effectiveness of a search is crucial and needs a good search engine that may be difficult to develop.	
	Flexible input interface.	Facilitate the design information acquisition from the designers.	Some designers may not be accustomed to such input interface.	
Concept selection support (optional)	Selecting a final solution concept based on a rational basis.	To justify and make design decisions based on facts consistently.	Designers may not be able to derive a rational basis due to various constraints and sometimes designers may tend to select on preferences, which may not be rational.	Yes
	Assigning weight or priority to solution concepts.	Allow important requirements to be considered first.	Consistency of assigning weight can be an issue.	
Concept ideation support (optional)	Generate new solution concepts based on past solution concepts.	Assist designers to utilise past solution concepts to be reuse to solve current design problems.	Past solution concepts may not be available or usable.	Yes
	Suggest solution concepts/principles to solve design problems.	Enable to the utilisation of design principles derived from research work on past solution concepts.	Suggestion of solution is usually at an abstract level and may not be interpreted correctly by the designer.	
	Provide guidance in stimulating designers to think "out of the box" or think more effectively to solve design problems.	To utilise psychological approach to influence designers to come up ways to solve design problems more effectively.	Such support is very dependent on the knowledge and the experience of the designer.	

Chapter 4
A Descriptive Design Methodology to Support Designers – Conceptualisation and Implementation

4.1 Overview of the need for a descriptive design methodology to support designers

The literature study on design methodology and investigations on a variety of design support facilities presented in Chapters 2 and 3 showed that current design methodologies do not provide comprehensive and integrated support facilities to designers. Most of the design methodologies, especially prescriptive design methodologies, are guidelines that facilitate design support in isolation to designers and do not utilise the advantage of information technology to help designers. The normative design methodologies were also found to be applied by subjective quantifications of utility value (Hazelrigg 1998; Thurston 1993) or conformance to axioms in the case of axiomatic design methodology (Suh 1990), in which they were usually applied in a specific domain or frequently that is difficult to implement in practice. In the case of descriptive design methodologies, they were found merely to provide a better understanding of the design process. Even then, studies have shown that the most common descriptive design methodology, protocol analysis, was found to be impractical and the information captured was prone to misinterpretation (Galle and Bela Kovacs 1996).

With such deficiencies, there is clearly a need to derive a new design methodology that is able to support a range of design support facilities on an integrated basis utilising information technology and other available design tools as well as allowing the re-use of design information and knowledge. This is because new products are getting ever more complex, require broad knowledge from multiple domains to create, and need to enter the market fast. Therefore, the new design methodology must be able to support a broad range of design tools and be able to assist designers who use different approaches and strategies.

Findings from the literature study showed that the descriptive design methodologies were found to reflect the actual design process. Hence, the descriptive design methodologies were able to accommodate different design approaches and strategies. However, current descriptive design methodologies are utilised for study purposes, are poorly structured, impractical and are unable to support any design tools. This is because of their lack of structure, which causes the descriptive design to capture enormous amount of irrelevant design information and knowledge, easily misinterpreted, very inefficient and unable to provide any support to designers. Due to these deficiencies, a new descriptive design methodology that is more practical and better structured and with the ability to support designers, is needed.

4.2 Conceptualisation and Derivation of a Descriptive Design Methodology to support Designers

Blessing (1998) has suggested that the findings from descriptive design methodologies could be used to facilitate the derivation of a prescriptive design methodology that reflected the actual design process. However, such a suggestion was difficult to realise as researchers have found that findings gathered from descriptive design methodology are diverse and conflicting because of differences in design approach and the preferences of designers (Von Der Weth and Frankenberger 1995). Hence, as Blessing (1998) has noted, most descriptive and normative design methodologies were derived based on experience and logical argument. The findings from the literature review on design support facilities have also clearly shown that some design support facilities do not affect a designer's preferences and strategies while some do.

In order to conceptualise a descriptive design methodology that describes the actual design process and supports designers in designing based on their individual preferences and approaches, a novel descriptive design framework that can accommodate design support facilities that do not interfere with the preferences and strategies of a designer needs to be derived. This novel framework will also need to be able to accommodate the design support facilities that will interfere with the preferences and strategies of a designer as optional features. Optional features provide flexibility to a designer to strategise their design work to involve additional design support tools that will influence their decision upon request. In addition to that, conceptualisation of the novel framework also requires a detailed study on what are the critical factors that should be included into the framework to reflect a realistic representation of the design process. What are these common characteristics and the critical factors that affect them? The need to investigate and identify these common characteristics is crucial as the current descriptive methodologies merely provide the characteristics of the existing design processes so that researchers can study these characteristics (Blessing et al. 1998).

4.2.1 Factors affecting the common characteristics of design tasks and the aims of a designer

A product design task has many characteristics. Some are unique and only occur for certain products while others are common. However, it is important to differentiate between design process characteristics and product characteristics. Product characteristics describe the physical and functional requirements of the final product, for example, "the device should be light in weight". This characteristic should be captured in the product design specifications. The common characteristics of a design task and the aims of a designer need to be examined and critical factors which affect them need to be defined. This is to provide the basis for deciding what should form part of the design parameters for the proposed descriptive design framework.

Different sources of information can be utilised to identify, examine and define both the common characteristics of a design task and the aims of a designer. The four sources are empirical studies, interviews, design experience and the results of analysis from literature reviews (Ahmed and Hansen 2002; Cantamessa 2003; Cash et al. 2010; Chakrabarti et al. 2004; Court et al. 1998; Galle and Kovács 1992; Girod et al. 2003; Hatamura 2006; Heisig et al. 2010; Mehalik and Schunn 2006; Nakakoji et al. 1999; Oxman 1995; Reymen et al. 2006; Sivaloganathan et al. 2000; Stauffer and Ullman 1988). This work employed most of the findings of a well-reported empirical study at Delft (Cross et al. 1996) and of other empirical studies (Chakrabarti et al. 2004; Kavakli and Gero 2003; Stauffer and Ullman 1991; Stempfle and Badke-Schaub 2002).

In addition, common characteristics (refer to Figure 4.1) were also derived from detailed observations of a team carrying out design work over a 6-month period. Using empirical evidence, logical reasoning or axiomatic approaches, researchers identified what designers aim to achieve. The literature on design, psychology, and cognitive science (Cosmides and Toby 1996; Dietz 2003; Harte et al. 1994; Heiser and Tversky 2005; Larkin and Simon 1987; Lee and Dry 2006; Lu et al. 2001; Vlek 1984) was therefore reviewed to construct a list of the main aims of a designer. It is recognised that the lists of common characteristics of a design task and the aims of a designer are not exhaustive and may not be universally accepted. Nevertheless, they were deemed sufficient for the purpose of deriving the proposed framework. Critical factors that are directly linked to the common characteristics of a design task and the aims of a designer are as shown in Fig. 4.1. It is important to note that each critical factor is as important as the others. There are circumstances where there may be additional critical factors linked to either the common characteristics of a design task or the aims of a designer but not shown in Figure 4.1. In some situations, the critical factors may be linked differently from what is shown in Figure 4.1. However, the aim of identifying and defining these critical factors is to determine those that strongly influence the common characteristics of a design task and the aims of a designer.

Critical factors affecting the characteristics/aims \ *Common Characteristics of a design task*	*The task may continue for days and weeks*	*New information continuously arrives throughout the design process*	*A designer has limited memory*	*Visual representation of design problems is important to designers*	*Design decisions are often made under uncertainty but will be reviewed when new information is available*	*Product design specifications are identified and documented*	*All design projects have a time constraint*	*Designers frequently refer to various resources to construct solutions (e.g. books, datasheets, experts, etc.)*	*Designers apply logical deduction and mathematical calculation to assist decision making*	*Current design decisions may affect other past and future design decisions*	*Past experience or design decisions may be re-applied to solve current design problems*	*Information and knowledge that lead to a design decision can be implicit*	*Different designers prefer different approaches*	*Designers introduce ideas, thoughts, and sub-requirements throughout the design process*	*There are usually trade-offs in making design decisions.*	*There is often miscommunication /misunderstanding*
Human memory	○	○	○								○					○
Time	○	○	○		○	○	○			○				○		
Information		○			○			○	○	○		○		○	○	○
Source of Information		○						○								
Presentation				○		○										○
Communication				○		○		○				○			○	○
Change					○					○						
Requirement						○									○	
Auxiliary Storage					○											
Knowledge					○			○	○	○	○	○	○	○	○	○
Psychology													○			
Rationale																
Decision Dependency										○						

The aims of a designer	Human memory	Time	Information	Source of Information	Presentation	Communication	Change	Requirement	Auxiliary Storage	Knowledge	Psychology	Rationale	Decision Dependency
Satisfy pre-defined requirements /constraints /criteria				●		●		●			●		
Make objective unbiased decisions			●	●									
Make traceable decisions		●			●				●				
Explore different design alternatives				●							●	●	
Complete a design task as planned				●			●	●			●	●	
Minimise design iteration				●		●	●		●		●	●	
Make decisions based on minimum risk				●							●	●	
Make decisions based on minimising utility costs				●							●		
Make justifiable & defensible decisions		●		●						●	●		
Minimise inter-related (coupled) decisions	●			●		●					●		
Total	2	2	2	19	2	5	4	8	5	3	17	12	5

Figure 4.1 Critical factors affecting the common characteristics of a design task and the aims of a designer

As expected, knowledge and information dominate, as design is a knowledge- and information-based activity. However, time, internal and external communication, presentation of ideas/thoughts/solutions, functional requirements and human memory also have a significant influence. With these critical factors identified and the type of support facilities that can be provided to a designer to help improve their design decisions without affecting their preferences and approaches defined, the next step is derive the structure that can accommodate these findings. However, it is not an easy task. This is because the structure needs to take into considerations the critical factors that will become the design parameters of the framework, the facilities that can be provided and the flexibility to accommodate other design tools if designers prefer it. The next section describes the conceptualisation of the structure in more detail.

4.2.2 The conceptualisation the descriptive design framework to support designers

The findings from investigations of the critical factors and the support facilities were utilised to conceptualise a descriptive framework that represents these factors and to model the design process. Empirical observations showed that design is a process where the product goal is translated into a set of requirements (stated as product design specifications) before the designer decomposes them into sub-requirements and sub-solutions and then arrives at the final concept solution (Figure 4.2).

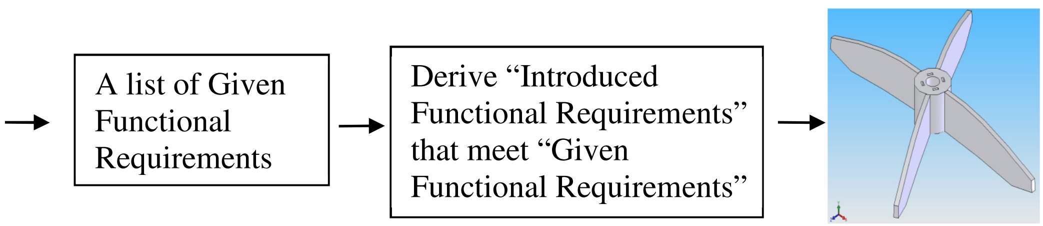


Figure 4.2 The product design process

Using the term "given functional requirement" adapted from the terminology "Given Criterion" used by Ullman (1996) in the design experiments performed at Delft

University of Technology, the requirements stated in the product design specifications could be represented as "given functional requirements" (GFR) and the sub-requirements and sub-solutions can be identified as "introduced functional requirements" (IFR). Based on these "given functional requirements" (GFR), the designers derive "introduced functional requirements" (IFR) to meet the "given functional requirements" (GFR) as illustrated by Figure 4.3. These "introduced functional requirements" include ideas, information, possible solutions, constraints, criteria and sub-requirements. Currently, the descriptive product framework proposed only focuses on functional requirements. Non-functional requirements such as aesthetics and ease of assembly are not considered.

Based on this product design process and with the findings from investigation of critical factors identified as critical design parameters above with the considerations on the type support facilities, the descriptive product design framework is derived by adapting the "cause and effect" model (also known as a fishbone diagram). The "cause and effect" model is adapted to have causal branches (requirements) that lead to the final concept solution, each causal branch having sub-causal branches (sub-requirements) as shown in Figure 4.3. As the inflow of new information throughout a design process is inevitable and changes because of earlier decisions made by designers are common, it is important that this framework is able to capture the inflow of information and data on how designers derive and decide on the appropriate "introduced functional requirements" to meet the "given functional requirements". This framework provides a graphical representation and enables designers to view their ideas, thoughts and design decisions with reference to time. With the graphical representation that links GFR to IFR, the framework is able to capture the process of a designer deriving "introduced functional requirements" (IFR) from "given functional requirements" (GFR). The framework also provides a platform that allows designers to decide on such information or to review their earlier decisions. This is consistent with the findings that the process of accessing and reviewing information is an important factor in the design process

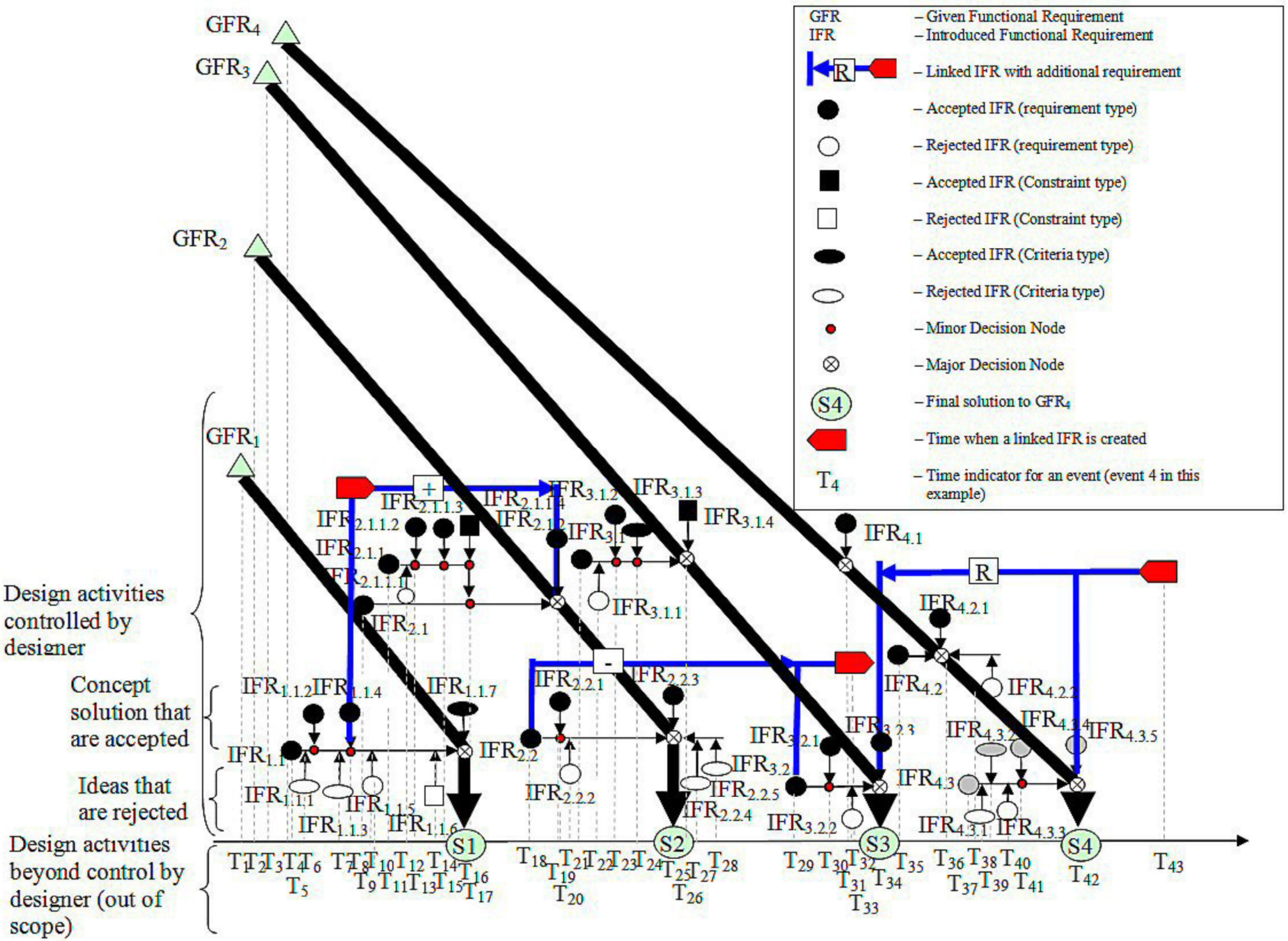


Figure 4.3 Framework of the Descriptive Design Methodology that allows a designer to design according to his preference and approach

(Court et al. 1998). The final outcome of applying this framework will be a concept solution that describes the embodiment of the product. Figure 4.3 is an illustration of the framework concept for a descriptive design framework. The descriptive design tool developed based on this concept is slightly different and may not need the legends indicated in Figure 4.3. This is explained more in the section 4.3.1 and 4.3.2.

4.2.3 Detailed description of the descriptive design framework to support designers

In a typical design project, the stakeholders, who include designers and customers, agree to design a product based on a list of "given functional requirements" (GFR). These "given functional requirements" are expanded from the product's goal during the requirement planning stage of the product design process as shown in Figure 4.3. The flow of new information and utilisation of knowledge throughout the design process are described by the designer when he decide on the appropriate IFRs to meet the GFRs. This provides a platform for the designer to review such information or to re-examine their earlier decisions. This should be a useful facility as accessing and reviewing information are important in design. The designer is encouraged to categorise GFRs and IFRs into six types, namely, requirement, constraint, criterion, issue/information, idea and solution. Initial planning on the type of category for IFR was to have only three types but this was later expanded to six. An IFR is considered a requirement if the designer uses words like "need", "require", "add", "remove", or other command verbs. For example, the statement "apply load on top of the device" is considered a requirement. A criterion is a statement that has a range description with predefined values, e.g. "to be between 5 and 15 mm in height". A constraint is a statement that has a value limitation. For example, "need to fit into a 5mm gap" is a constraint as it means the device cannot exceed 5mm in size. In addition to that, an IFR is considered an idea if the designer comes up with a possible solution while some IFRs are considered as information or issues when the designer wants to inform or remind others of outstanding design circumstances. The final type of IFR is the solution itself, normally the solution of a sub-requirement or constraint or criterion. Some IFRs may merely be issues or information while others are just ideas or solutions.

The differentiation of types of IFR is useful, as when similar design problems are encountered in other design tasks, similar constraints or criteria can be applied. If the designer feels that such categorisation is cumbersome and distracting, they do not have to use it. This gives flexibility to the designer to express their IFRs. The main structure of the framework is derived with the intention of providing a focus for the designer to decompose the GFRs in order to achieve the final concept solution. The final descriptive product design framework proposed is a time-dependent framework. It captures IFRs based on the time when they were created in order to satisfy the GFRs. The triangular symbols in Figure 4.3 represent the initialisation of GFRs. The heights of these triangular symbols were adjusted (to higher positions) to cater for the fitting in of the IFRs that were derived during the design process. The x-axis allows the visualisation of the time for every initialisation of GFR or IFR during the design process and does not follow any scale. This is to reduce the length of the graphical representation, which may continue for months. All IFRs that are thought of by the designer are shown as grey circles (if they are of the "requirement" type) with arrows pointing vertically downwards towards the GFR (triple line) as exemplified by $IFR_{4.1}$ in Figure 4.3. These grey circles are created with "work in progress" or "WIP" status (e.g. $IFR_{4.3}$) and will keep that status until they are final (accepted) or abandoned (rejected) by the designer. The grey circles will turn into black when the corresponding IFRs become "Final" or accepted (remain in a higher position with their arrows pointing downwards) or white when they are "Abandoned" or rejected (moved downwards with arrows pointing upwards). Hence, all black and white circles were once grey circles.

If the newly created IFR requires more sub-IFRs to address it then similar grey circles will be created but these sub-IFRs will have arrows that point towards their respective parent IFRs as shown in Figure 4.3. This means that for an IFR such as $IFR_{2.2}$ that has sub-IFRs (e.g. $IFR_{2.2.1}$ and $IFR_{2.2.2}$), the grey circle moves from its initial upper position to a horizontal position with its arrow pointing horizontally towards its related GFR (GFR_2) before turning into a black circle when it is accepted as shown in Figure 4.3. When a sub-IFR such as $IFR_{2.1.1}$, requires more IFRs to address it, then its arrow remains horizontal and points towards its parent ($IFR_{2.1}$) but it also moves to a higher position to accommodate its sub-IFRs ($IFR_{2.1.1.1}$, $IFR_{2.1.1.2}$ and others) (refer to Figure 4.3). Some IFRs are created after the solution is found, for example, $IFR_{2.2.4}$ and $IFR_{2.2.5}$, which stay on the extreme right of the solution "S_i". There are possibilities that IFRs are proposed after a solution is found in order to improve on the solution. Any IFR proposed and accepted after the initial solution is found would be included as a "Final" or accepted IFR and the GFR thick triple line will shift to accommodate it as shown for $IFR_{3.1.3}$ in Figure 4.3. In addition to that, at times the designer may want to capture some of the design issues or information during the design. Any issue or information that is captured should be given the "WIP" status and when solved or already dealt with will be given the status "Solved".

Similarly, an IFR link can be created before and after a solution is found. Each arrow represents a requirement relationship needed to solve the related GFR or IFR. Constraint-type IFRs are represented by squares and criterion-type IFRs are represented by ellipses. As mentioned earlier, a completed design should not have any "WIP" IFRs (no grey circles, squares or ellipses should exist).

4.3 Development of the Descriptive Design Methodology Tool to support Designers

The conceptualisation of the descriptive design framework merely provides a conceptual structure for the descriptive design methodology that supports designers. This conceptual structure was further developed into a prototype software tool to enable the facilitation of the methodology and the support facilities to the designer in the actual process of designing. The development of such tool also provides the opportunity to identify the strengths and deficiencies of the methodology from a practical perspective so that further improvement can be made as prescribed by the active research methodology loop adopted in this research work. The development of the tool is focused on three areas, namely, the architecture of the software tool, the user interface (text and graphical user interfaces) and the database structure of the tool.

4.3.1 The architecture of the descriptive design methodology tool

The development of the descriptive product design tool was based on the descriptive product design framework and the type of support facilities that can be incorporated in the tool in order to successfully facilitate these supports as well as to capture the ideas and thoughts of a designer.

The descriptive product design tool was developed on the Windows platform. The tool depends on the designer to record his "introduced functional requirements" during the design process. Designers are encouraged to record anything that they can think of in relation to each "given functional requirement". Descriptions of ideas are expected to be concise and specific.

The architecture of the descriptive product design tool is shown in Figure 4.4. As shown in Figure 4.4, the tool is initialised with a temporary repository or database system to enable real-time capturing of the designer's thoughts. As the designer inputs his thoughts, the temporary repository will capture his input via the multi-layered text-based interface.

4.3.2 Design of user interface for designers

There are two main text input interfaces for designers, i.e. the “given functional requirement” interface and the “introduced functional requirement” interface. From the “given functional requirement” interface, the “introduced functional requirement” interface can be launched to provide the designer with the opportunity to input a hierarchical structure of “introduced functional requirements”. The tool is initialised with a text input user interface. There will be a button known as “Display Design Time Line” to launch the graphical user interface and the designer can proceed with either one of the interfaces at any phase of the design process.

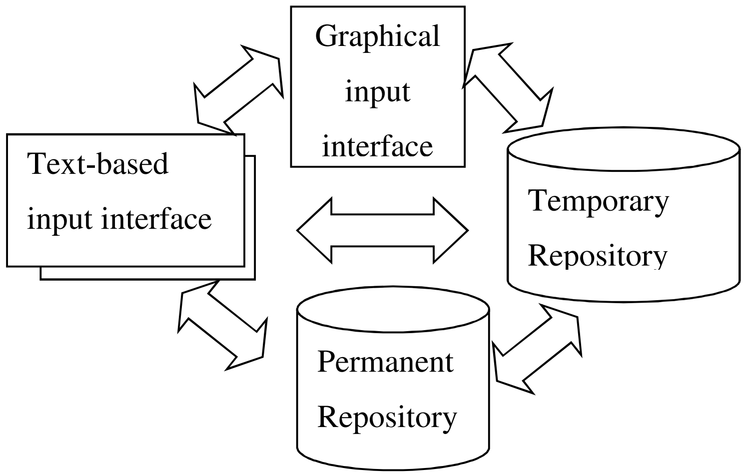


Figure 4.4 Architecture of the Descriptive Design Tool

4.3.2.1 Text Input User interface for Given Functional Requirements (GFRs)

The text input interface is initialised with a “given functional requirement” interface as shown in Figure 4.5. This interface allows designers to provide information similar to that found in the product design specifications used by Pugh (1991). In this way, the tool helps designers to find solutions that meet the pre-defined “given functional requirements”. The designer can specify a large number of “given functional requirements” as there is a scrollbar for him to scroll down and view or input more GFRs. At any time, the designer is allowed to change and edit his “given functional requirements” until they start the user interface for “introduced functional requirements". The designer is also allowed to remove any "given functional requirement" at any time but the tool will not delete it from the database. The removed "given functional requirement" will be transferred from the GFR field in the database to a "Removed GFR" field for reference purposes. After starting the user interface for "introduced functional requirements", the designer can only remove or add "given functional requirements" but is not allowed to edit the existing GFRs. All changes at this stage are recorded.

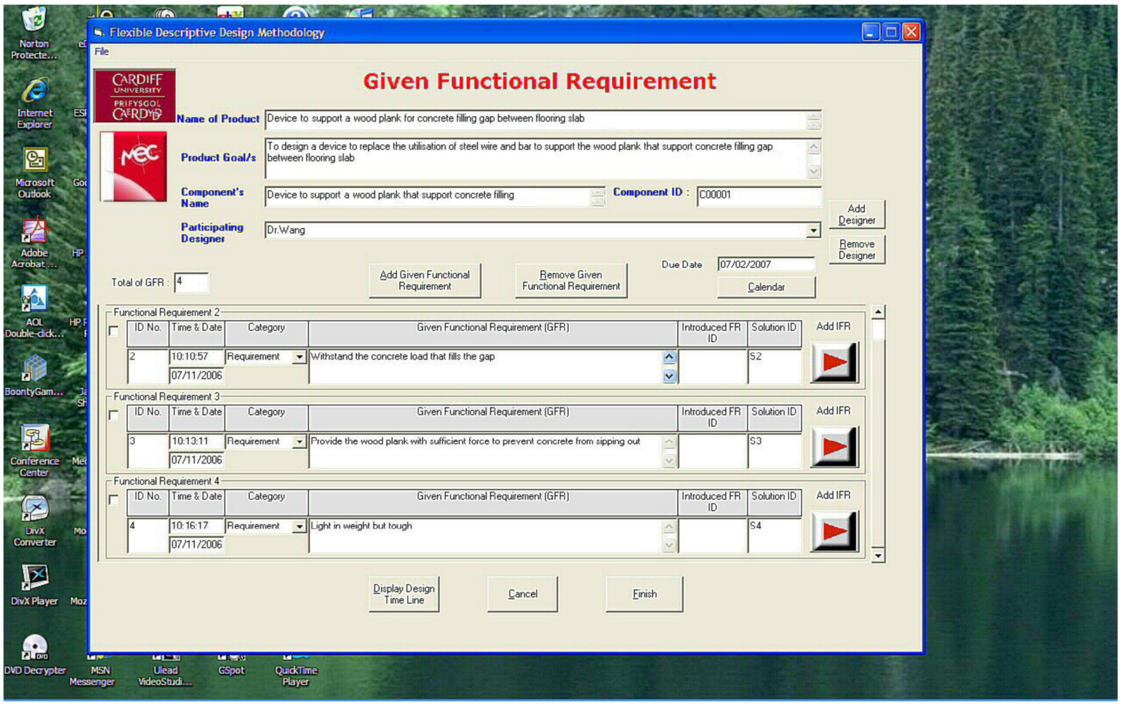


Figure 4.5 The text-based interface for "Given functional requirements"

During the process of capturing "given functional requirements", a designer is also allowed to specify constraints and criteria. When the "given functional requirements" are recorded, the designer can proceed to think of possible "introduced functional requirements" to satisfy the "given functional requirements". The designer can start without any particular order and proceed to derive "introduced functional requirements" for each "given functional requirement" respectively. Similarly, the designer can also categorise the IFR as a requirement, constraint or criterion. For example, if the designer wishes to derive an "introduced functional requirement" for "given functional requirement" number 3 (refer to

Figure 4.6), he can click the button with an arrow icon at the respective "given functional requirement".

4.3.2.2 Text Input User Interface for Introduced Functional Requirements (IFR)s

The text input user interface for "introduced functional requirements" is shown in Figure 4.6. The interface seeks additional information from designers during designing. This tool was developed to support the recording of all "introduced functional requirements" suggested by designers. Only a single tool is needed and shared by all designers involved in the design process. However, if the designers are not at the same location, this tool will need a separate conferencing tool to allow different designers to propose their "introduced functional requirements". Each "introduced functional requirement" is attached to the designer who proposes it. Any justification of the proposed "introduced functional requirement" can be recorded along with the "introduced functional requirement" itself. Similarly, any disagreement or counter proposal of "introduced functional requirements" is also recorded. If one of the "introduced functional requirements" is accepted by all after discussion and brainstorming, the status of the "introduced functional requirement" is then changed from a "WIP" to "Final". "WIP" is an abbreviation that stands for work in progress. Additional justifications for the final "introduced functional requirement" can be also recorded. In addition to that, since an IFR can be an issue or information idea or solution, the status of an IFR can also be "Abandoned", "WIP" or "Solved".

As shown in Figure 4.6, designers can also add links from any proposed "introduced functional requirement" to any "given functional requirement" or "introduced functional requirement" and provide a description of the effect of the link. To ease the effort of stating the effects of the link, any link that improves and enhances another given or introduced functional requirement is considered to have a positive effect. For example, one of the "given functional requirements" in the design of a device for concrete filling is to be light in weight and tough. A positive effect on this GFR from another IFR implies the proposed IFR will decrease the weight. If the terms positive and negative cannot distinctly describe the effects, the designer is allowed to create a requirement link effect. This allows the designer to input a detailed description of the link. The user interface for “introduced functional requirements” can be launched when there is a need to create additional sub-requirements, constraints or criteria to evolve the related “introduced functional requirements”. A button is provided for the designer to launch further sub-level user interfaces for each “introduced functional requirement”. Each sub-level interface is similar in features and appearance to the one shown in Figure 7. Similar to the interface for “given functional requirements”, the “introduced functional requirement” interface allows the designer to edit their proposed IFR. This is not allowed when the designer has moved on or has added another “introduced functional requirement”. As mentioned previously, the designer can still remove any “introduced functional requirement” but the database will still keep a record of the removed IFR by changing the IFR to “Removed” or “Edited” status. Otherwise the default status will be “Active”.

One of the important pieces of information captured by the descriptive design framework is the actual source of the sub-requirements, criteria, constraints, issues or information, ideas, and solutions proposed by the designer. In this research work, this source is known as “Source of evidence”. In the process of designing products, designers came up with various ideas, solutions, information and others from many sources. These sources are important in enabling a designer to trace the source of their decisions if there is any errors occurred during the design process. This research work was adapted and expanded from some of the sources mentioned in the work of Hicks et. al. (2002). In this research work, the source of evidence field was set to a selection list of “Knowledge/Experience", "Colleague", "Expert", "Consultant", "Experiment", "Standards", "Customer", "Supplier", "Calculation", "Literature", "Patent Info", "Government", "Advertisement", "Institutions/Association/Societies", "Experiment", "Standards", "Customer", "Supplier", "Calculation", "Literature", "Patent Info", "Government", "Advertisement" and "Institutions/Association/Societies".

This list of sources of evidence is also crucial for the designer to perform a search on the sources that have influenced their decisions during the design and be aware of the sources of data which are prone to error. Any “introduced functional requirement” can include a sketch (by hand or by computer) or document or CAD file for reference. This is done by clicking the “Add DOC” button next to the related “introduced functional requirement”. A menu will provide an option to preview and store any selected image or documents or to link to a CAD file. This is launched when the “Add DOC” button is clicked. Once the image of a sketch or the document or a CAD link is stored for an “introduced functional requirement”, there will be a small icon on the right hand side of the “Add DOC” button to indicate that there are linked images or documents or a CAD file for that particular IFR.

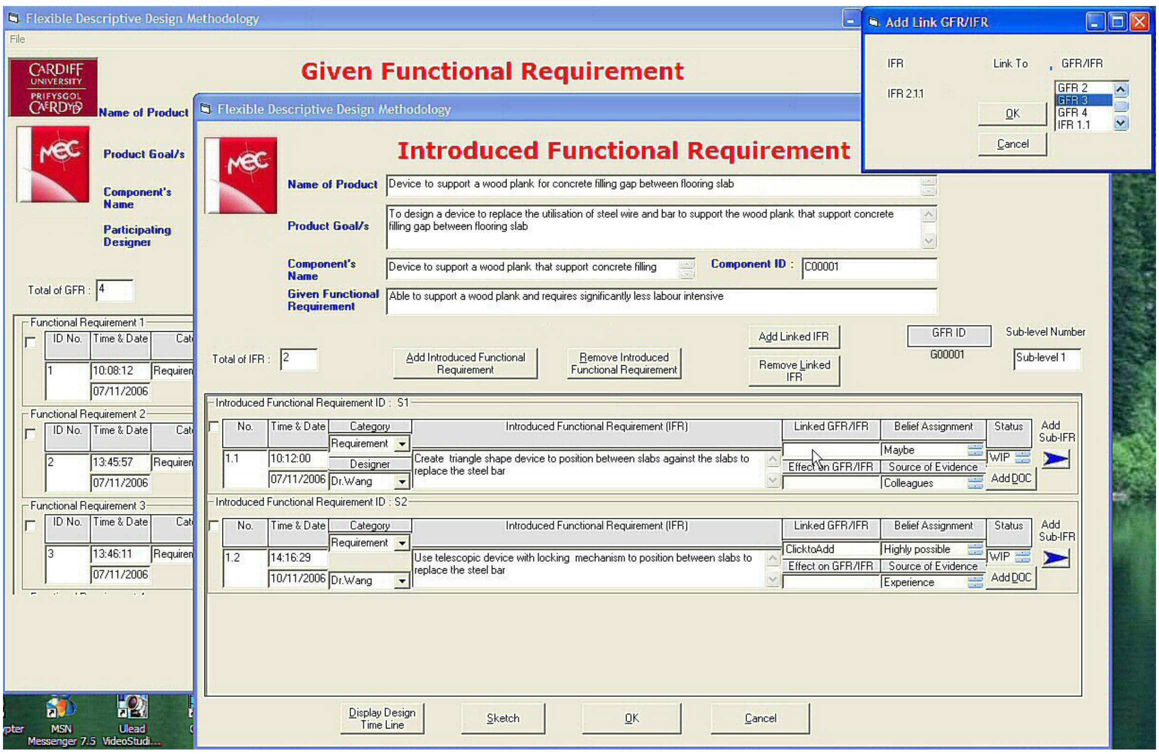


Figure 4.6 Text-based interface for “Introduced Functional Requirements” with the link menu launched

One of the important support features that this descriptive design framework brings is a facility to support designers in deciding whether to delay or to make a design decision when information is missing. The tool provides a dialogue box for the designer to record relevant information that can be available later. The tool will remind the designer of the information at the appropriate moment. This dialogue box is shown in Figure 4.7. The designer needs to double click the "introduced functional requirements" textbox to launch this dialogue box. This is because the delaying of a design decision is linked to individual "introduced functional requirements". For example, in order to determine whether a triangular device is may be slotted into the gap between a pair of housing slabs, the designer decides to refer to experts for detailed measurements of the slab dimensions.

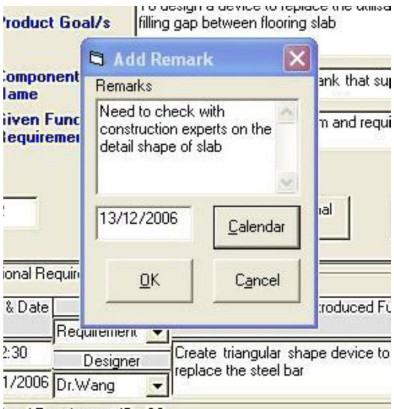


Figure 4.7 Dialogue box for designers to provide information on reminder for delaying design decisions

4.3.2.3 Graphical User interface for designers

The graphical user interface would be similar to the text but the designers are able to view the links between the designer's ideas and thoughts as "introduced functional requirements" (IFR) and the "given functional requirements" (GFR) defined in the product design specification list with reference to time and date. The graphical user interface is illustrated in Figure 4.8. It has features similar to the text-based interface with an additional feature of searching for matching ideas and thoughts (IFRs) that can be categorised into requirements, constraints, criteria and solutions. The results of a search will be highlighted by showing the entire branch of IFRs including sub-

IFRs that matches the single word specified in the search. The results could also point to several different branches of ideas or thoughts if similar words are used.

A zoom snapshot of the graphical user interface is shown in Figure 4.8 in one of the case studies, the design of a concrete filling support device (X-shape device) before the design improvement phase starts. The designer is encouraged to perform a search which includes the category in order to improve the results of searching. However, the designer can also search without specifying the category. It is also important to note that if the designer has not specified the category of their ideas or thoughts earlier during the design process, the outcome of the search with or without category specified will be similar.

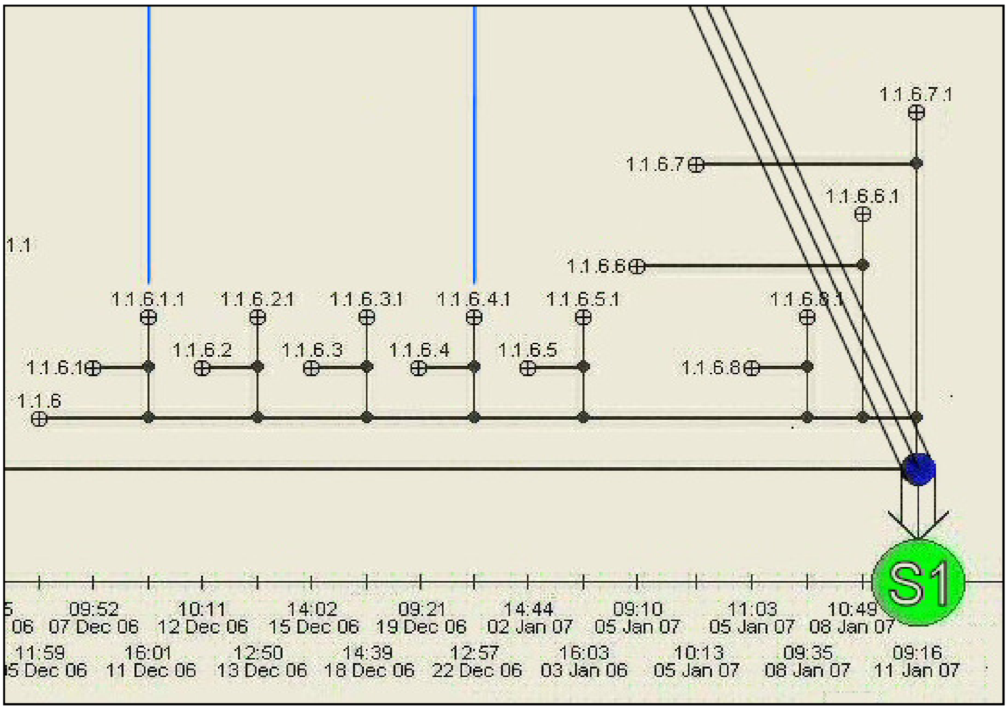


Figure 4.8 The graphical interface of the descriptive design tool (a zoom-in view)

The graphical user interface also allows a designer to add IFR to any GFR or sub-IFR to any IFR via a popup menu or the toolbar menu when the designer right-clicks at the GFR or IFR respectively. Similarly, the popup menu also allows a designer to remove one or more IFR to GFR or sub-IFRs to IFR and allows a designer to add positive, negative or a remarked link between two IFRs. Whenever the designer make changes using the text input user interface, the tool will update the changes to the graphical user interface and vice versa. All settings, such as colour and thickness of lines, can be changed as shown from the graphical user interface menu but not in text input user interface shown in Figure 4.5 and Figure 4.6. The settings can be set in the graphical user interface through the popup menu or the toolbar menu. For any changes made by the designer to the IFR or GFR, the graphical user interface will highlight the sub-branches of IFRs affected by the changes to inform the designer the effects of the changes. The popup menu has feature similar to the toolbar menu such as add, remove, modify GFR or IFR or link. There are also sub-menus for GFR or IFR to add or remove a jpeg image or document file, to set the status (Abandoned, Solved, WIP, or Final), and to input the source of evidence and others. As shown in Figure 4.8, the triple line is the first GFR and the blue circle represents the node of where IFRs meet their predecessors. The IFR is represented by the bold dark horizontal line and the circles with cross nodes are the sub-IFRs.

4.3.2.4 Database design

The current prototype software tool was developed with links to the Microsoft Access database system. This tool is a stand-alone system and its database schematic structure is shown in Figure 4.9. The temporary repository and the permanent repository have a similar structure but differ in name. Both repositories consist of database repositories, which are linked as shown in Figure 4.9. Each repository was developed based on their data group. The data groups are products, components, "given functional requirements" (GFRs), "introduced functional requirements" (IFRs), pictures (images of sketches) or documents, linked IFR, CAD data file and participating designers. The advantages of linking various data groups together are to enable different database repositories to deal with different data so that the main database of IFR, which contains all the ideas and thoughts is not too large. Designers can trace their ideas and thoughts to the product design specifications represented by GFR to their sketches, the documentation containing their calculations performed for a particular design and the related computer–aided design files, which they have created. Such traceability allows the designer to directly call the image, CAD or document files by just clicking the links created. The big green circle is the final solution that fulfils the first GFR or GFR 1. Figure 4.10 shows a screen capture of the database system for the prototype descriptive design tool that stored the data captured via the user-interface shown in Figure 4.5 to Figure 4.8 for Case Study 1. The design of the database system was later improved to introduce tables to capture the results generated by a TRIZ-based ideation system in Case Study 2. Figure 4.11 shows a screen capture of the improved database system for the prototype descriptive design tool for Case Study 2.

Figure 4.10 and Figure 4.11 depict the major differences between the database for Case Study 1 and Case Study 2. They arise because the involvement of the data capturing of the TRIZ-ideation tool significantly expands the database system as well as the complexity of the relationships between tables of data.

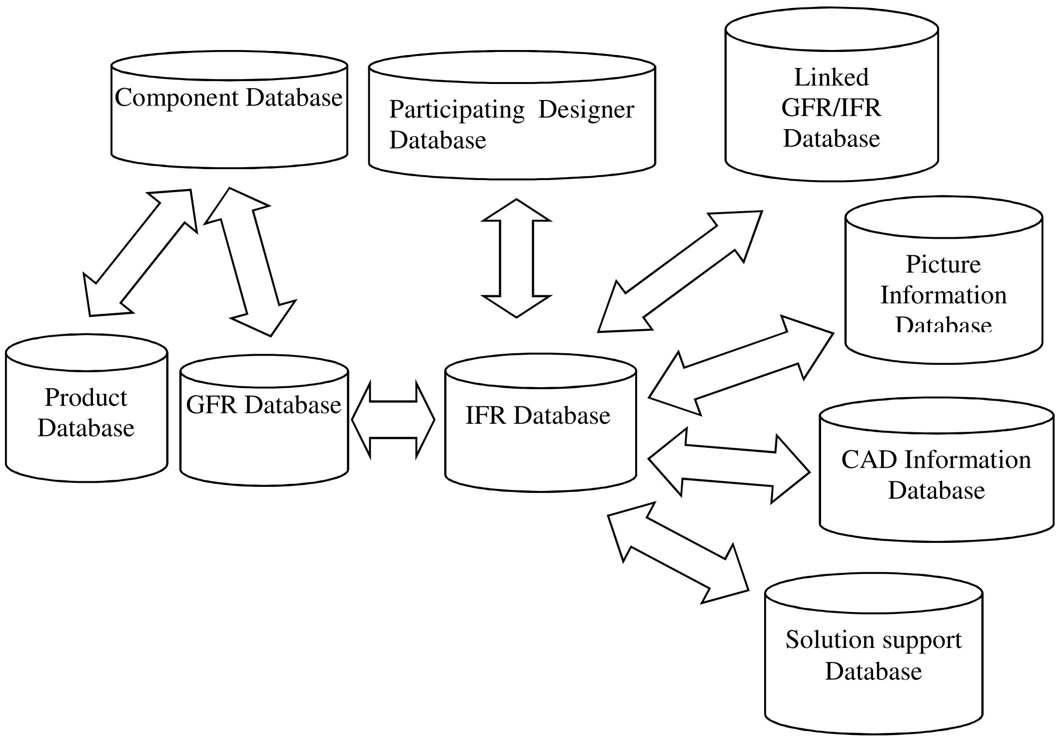


Figure 4.9 The schematic structure of the repository system for the descriptive product design tool

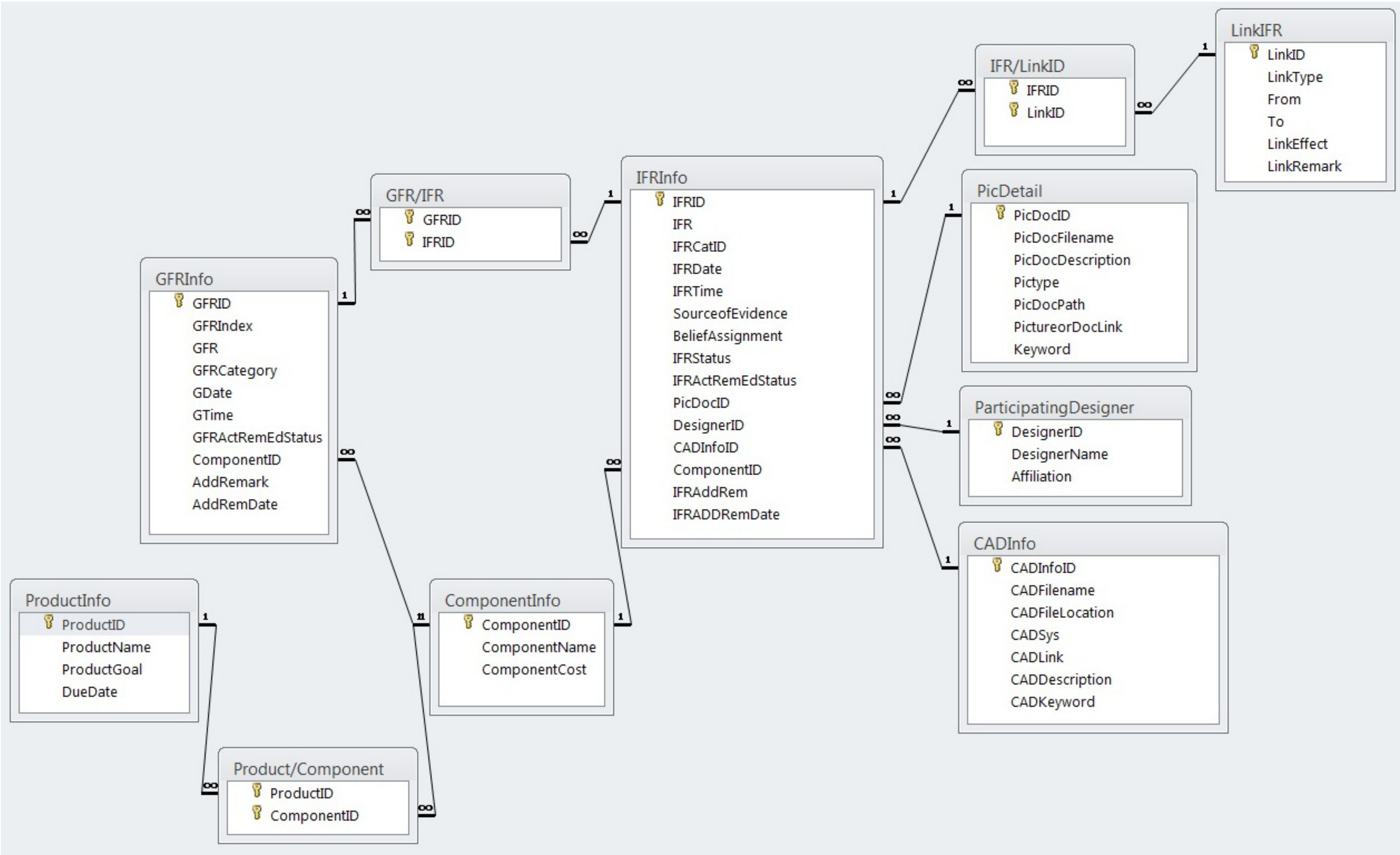


Figure 4.10 A screen capture of the database system for the prototype descriptive product design tool for Case Study 1 (Phase 1 and Phase 2)

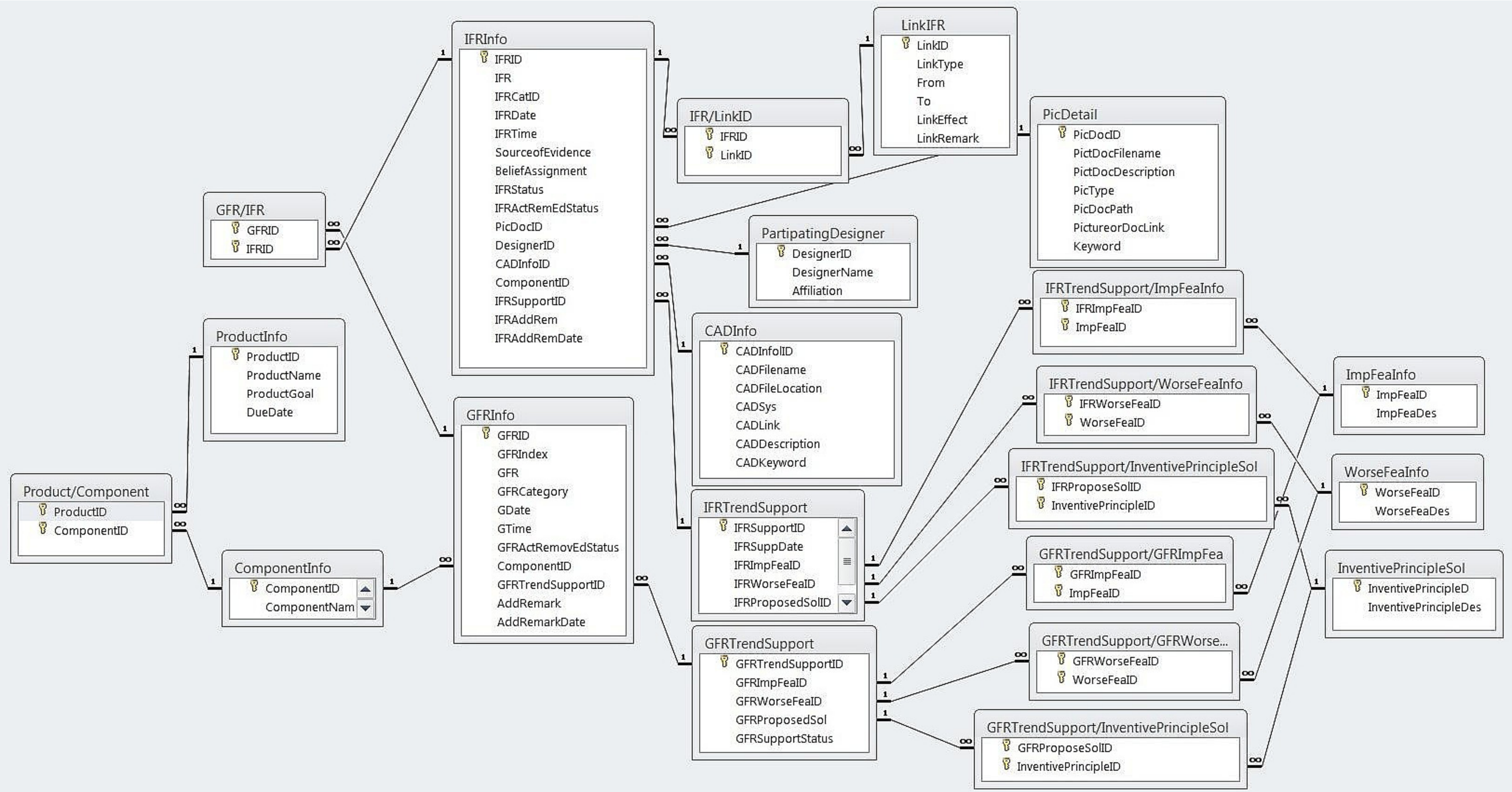


Figure 4.11 A screen capture of the improved database system for the prototype descriptive product design tool for case study 2

4.4 Summary

The derivation and the conceptualisation of the descriptive design methodology tool were inspired by the cause and effect diagram, which was then adapted to become the framework of the descriptive design methodology. The representation of the descriptive design methodology parameters was decided after further investigation that explored beyond the scope of engineering design into the domain of psychology and cognitive science to determine the common critical factors that affect a design process.

With the descriptive design framework conceptualised, the development of the descriptive design tool was carried out with the major tasks of developing the architecture of the software tool, the user interface, which consisted of a text-based user interface and a graphical-based user interface and the design of the database for the tool. With the software tool developed, two main case studies were conducted later and are explored in the next chapter.

Chapter 5
Case Study 1: Designing A Device/Method to Support Concrete Loading In Between Beams

This case study is the first one to be conducted to verify the advantages and the effects of the novel descriptive design methodology for an experienced designer who does not follow any established design methodology but who applies their own approach and strategies in design instead. This case study consisted of two phases. The first phase or Phase 1 was carried out with the text-user interface of the prototype descriptive design software tool (see Figure 4.5 to 4.7 of the previous chapter) only as the graphical user interface was still in development. In all circumstances, the designer will start with the text-based user interface for determining the GFRs (refer to Figure 4.5) as these are core design requirements. The designer can then use the text-based user interface (Figure 4.6) or graphical user interface (Figure 4.8) to capture their IFRs (sub-requirements, ideas, solutions, etc.). The graphical user interface will display both the GFRs and IFRs in terms of time and links. The second phase of the case study was a continuation of the first, where several improvements to the device were carried out using the descriptive design prototype tool. The details of both phases are elaborated in the next few sections.

The implementation of the descriptive design methodology also depended heavily on the willingness of the designer to utilise the descriptive design tool to capture their ideas and thoughts throughout the design process. The findings from this case study showed that the descriptive design methodology was able to capture the thoughts and ideas of the designer throughout the design process. The outcome from this case study reflected the versatility of the descriptive methodology to support designers irrespective of their approaches and strategies of design, whilst providing a structured and systematic method of capturing design knowledge for reusability purposes and for design improvement aims.

5.1 Overview of Case Study 1 (Phase 1 and Phase 2)

The case study involved a single experienced designer who was knowledgeable in product and mould design. The design project involved designing a device or a method to replace the current method of supporting wood planks placed in between and along the gap of beam slabs in the construction industry. The current method used is illustrated in Figure 5.1. The case study (Phase 1) was initially conducted using the descriptive product design tool with the text-based user interface only and some results of using just the text-based user interface were published (Pham et al. 2007). Case study 1 (Phase 2) was the continuation of the same project but for some modifications to the designed device using the completed descriptive product design tool with both text–based and graphical user interface used during the process to improve on the initial design. The information gathered using the text-based user interface in the initial design was retrieved and reused to improve the initial design and with the graphical user interface included, the designer is able to gain additional support and better visualisation of their design activities and past decision-making.

In designing a device to support the weight of concrete between beams in a structure (refer to Figure 5.1), five “given functional requirements” were specified by the customer. The current method of support is to place a wood plank along the gap at the bottom of a beam. The current method requires each plank to have a pair of holes at several intervals. The construction personnel will put steel cables through these holes and then coil the steel cables around a steel bar. The steel bar is then rotated to twist the steel cables, which then pull the wood plank against the beams. The steel bar, steel cables and wood planks provide support for the weight of concrete between the gaps as the pouring of concrete takes place. The concrete is then allowed to dry and harden for days. The steel bars along with the sections of steel cables that protrude from the concrete and the wood planks have to be removed after the concrete has hardened. This is the existing method and this method is labour intensive, arduous and costly.

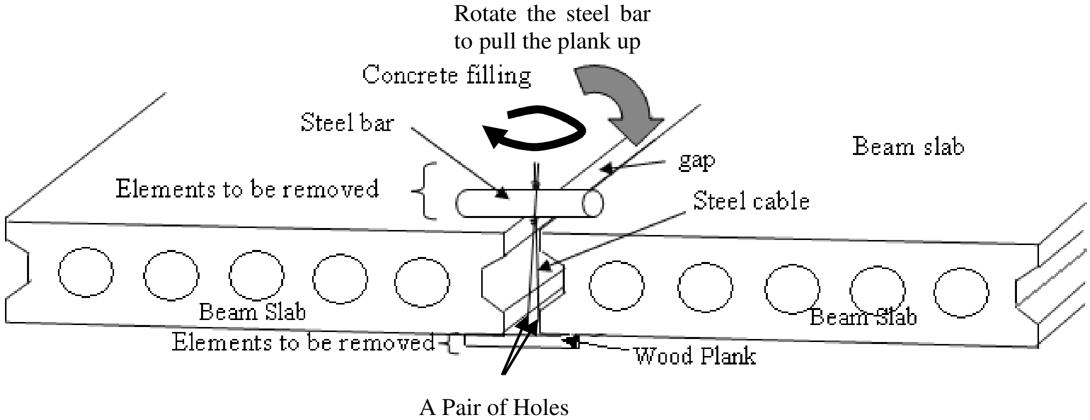


Figure 5.1 The existing method to support concrete loading in between beam slabs with steel bars, steel cables and a wood planks

5.2 Implementation of the Descriptive Design Methodology in Case Study 1 (Phase 1)

This project duration was about 4 months (Phase 1 and Phase 2) and it was initiated by a customer who was looking for ways to replace the current method with a method that is less labour intensive and should cost less.

5.2.1 The aims of the Case Study 1 (Phase 1)

The aims of Case study 1 (Phase 1) were:

i) to validate that the descriptive design framework (via the prototype software tool) is able to assist and support a designer that uses their own design approach and does not use any established design methodologies

ii) to ascertain how well the descriptive design framework can capture the thoughts and ideas of a designer throughout the design process

iii) to find out whether the framework is able to help the designer to decide on delay in making design decision when the design information is insufficient or unavailable

iv) to show that the descriptive design framework via the prototype software tool allows the designer to edit, save and add design decisions throughout the design process.

v) to find out whether the descriptive design framework (via the prototype software tool) can perform searching of design data.

5.2.2 The details of implementation for the Case Study 1 (Phase 1)

Looking at the aims of the case study 1 (Phase 1), i) to v) are design-methodology-related support facilities while vi) is a computational-platform-related support facility. For Phase 1 of case study 1, the case study was carried out without the graphical user interface. Hence, several design support facilities were not available for verification but these support facilities will be verified in the next phase or the next case study.

In the implementation, the prototype descriptive design software tool was provided to the designer to test how well it can support him in designing the device. The designer after discussion with the customer came up with the design specifications for a new improved device. These design specifications were then entered to the software tool as "Given Functional Requirements" or GFRs. The GFRs to design the new device that pulls the wood planks against the beams so that concrete can be poured into the gaps between the beams are:

1. Able to support wood planks and is significantly less labour intensive.
2. Withstand the load of concrete that fills the gap.
3. Provide the wood planks with sufficient force to prevent concrete from leaking out.
4. Light in weight but tough.

Prior to the implementation of the descriptive design tool, the designer was given an introduction and briefing on the prototype software tool and some basics on how to use it. One of the important explanations given was on how the prototype descriptive design software tool works, which requires further elaboration of the concept of the descriptive design methodology framework. The proposed descriptive design methodology framework was derived based on the concept of product design process shown in Figure 4.2 (refer to previous chapter) which depicts the design process as one of evolving a list of design requirements i.e. GFRs into IFRs (sub-requirements, constraints, criteria and ideas) into the final concept solution. This is fine if the final concept solution consists of a single component solution but for a multiple components solution, there is a need to expand the evolution of GFRs further into multiple sets of IFRs for each component that finally depicts the final concept solution.

With reference to that concept (see Figure 4.2), the descriptive design methodology initialised with a particular product in mind (can be new or an existing one). Hence, the descriptive design tool will be initiated with a request for the designers to provide the name of the product, the product's goal, the name of the designer, the due date of the design project and lastly, the name of main component. In this circumstance, it is assumed that the designer started to design the product from the main component and as they design, they may expand to additional components, which are possibly available off the shelf or need to be designed. The design process using this descriptive design tool can be schematically illustrated as in Figure 5.2.

This means that the final solution concept could be a single or multiple components product depending on the decision of the designer.

For this case study, the design process started with a single product with a single main component in mind. The GFRs for the product were identified and were expanded into IFRs for the main component and if the expansion of IFRs led to the involvement of additional components that need to be designed, then an additional separate flow of the descriptive design process for the component would be needed. However, if the additional components can be procured off the shelf then it is not necessary to use an additional flow of the descriptive design process. The designer just needs to describe some data or procurement information related to the components, as shown in this case study with the usage of cable tie to secure the wood planks which support the concrete. Based on this elaborated descriptive design methodology, the descriptive design software tool was used in this case study to capture the designer's thoughts and ideas throughout the design process for a device or method to support concrete loading between beams.

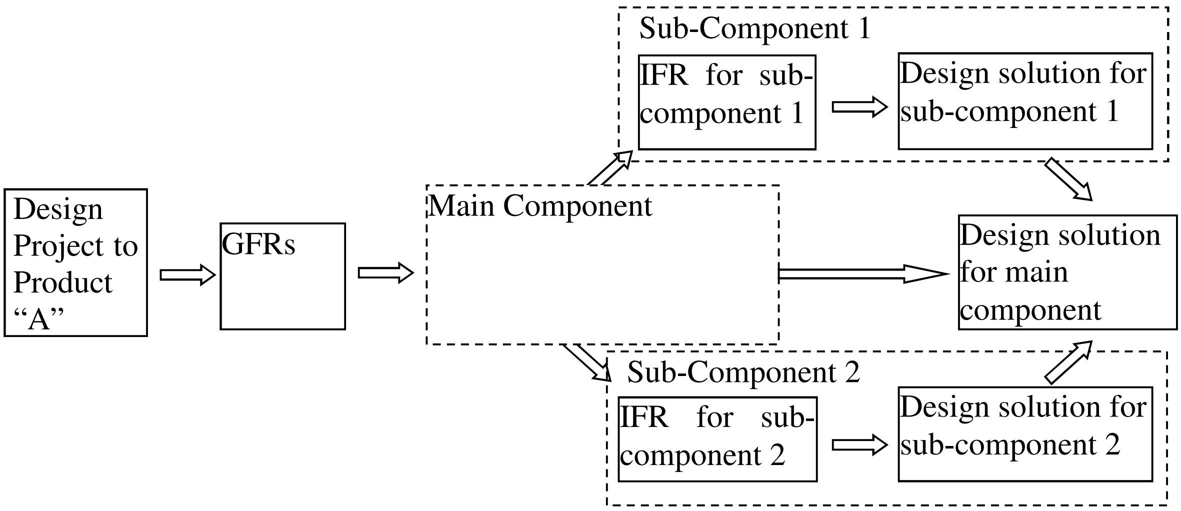


Figure 5.2 The design process of the descriptive design tool

Figure 4.9 of the previous chapter illustrates the entire database system developed using the Microsoft Access as the repository system for the descriptive design tool. These database tables are merely developed for repository purposes behind the prototype descriptive design software tool, which was developed based on the descriptive design methodology framework to capture the thoughts and ideas of designers throughout the design process. The prototype descriptive design software tool also allows the designer to perform basic search, tracking and tracing of their design decisions. This is done via visualisation of the design process and by conducting simple queries on the database system. These are the computational-platform support facilities that are made possible by the architecture of the descriptive design concept.

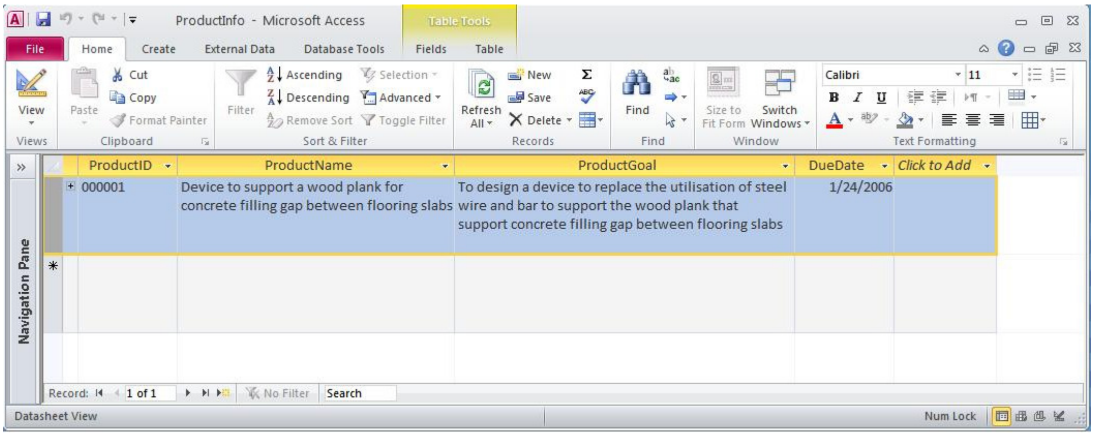
5.3 Results from the Implementation of the Descriptive Design Methodology in Case Study 1 (Phase 1)

The implementation of the descriptive design methodology using the prototype software tool has resulted in the success outcome of the derivation of the device to support the concrete loading between beams as well as the capturing of the designers design decisions and the design path involved. In order to evaluate and analyse the effectiveness of the proposed descriptive design methodology, two methods have been adopted. The first method is to observe and analyse the information captured by the database to determine whether the various support facilities that were provided by the proposed descriptive design methodology can be verified. The second method will involve interviewing the designer about their opinions and suggestions on the proposed descriptive design methodology. The feedback from the simple interview would substantially contribute to the validation of the proposed descriptive design methodology from the perspective of how well the tool supports the designer. The interview was carried out after Phase 1 and Phase 2 of the case study 1 to obtain an overall feedback from the designer on the tool and the support framework. This section will show screen captures of the information stored in the database and the outcome of the two methods will be elaborated in the next two sections.

5.3.1 The database information captured in Case Study 1 (Phase 1)

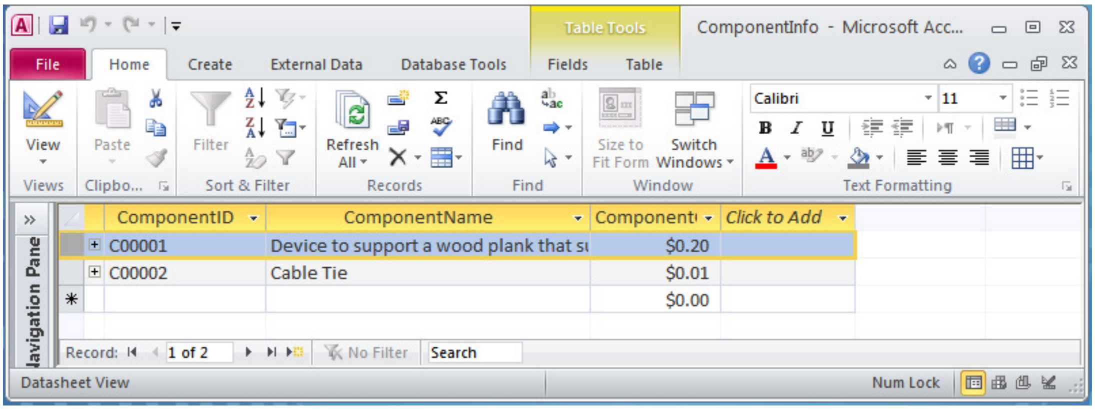
As mentioned earlier, the first information provided by the designer was the information for the product he was designing. Table 5.1 shows the product information table that stores the details of the product designed by the designer at the GFR text-based user interface (refer to Figure 4.5). This product information and details are stored in the ProductInfo table (Table 5.1) of the Microsoft Access database. The ProductInfo table is linked with a relationship of many-to-many with the ComponentInfo table (refer to Table 5.2). This means that a product can have one or more components and a component can be used by a single product or multiple products. The component info table will then be linked to the GFRInfo table (refer to Table 5.3) where the GFR itself will be linked to the IFR table. The GFRInfo table and the IFR table are linked via a many-to-many relationship, as one IFR is used for several GFRs while several GFRs may be linked to a single IFR. This means a sub-requirement, constraint, criterion, information, idea or solution can be used to address several core requirements and vice versa. In order to create a many-to-many link between tables, junction tables such as Product/Component, GFR/IFR and IFR/LinkID. Table 5.4 show the IFR data captured during this case study and stored in the IFRInfo table before the case study (Phase 1) was completed. As seen in Table 5.4, a lot of the IFRs are with "WIP" status as the case study is in progress. The other database tables such as the CADInfo table, PicDetail table, LinkIFR table, and ParticipatingDesigner table were merely created as a supplementary database to the IFRInfo database. The entire database with its tables and the relationships between tables is shown in Figure 4.10.

Table 5.1 A screen shot of the data captured in ProductInfo table by the descriptive design prototype software tool



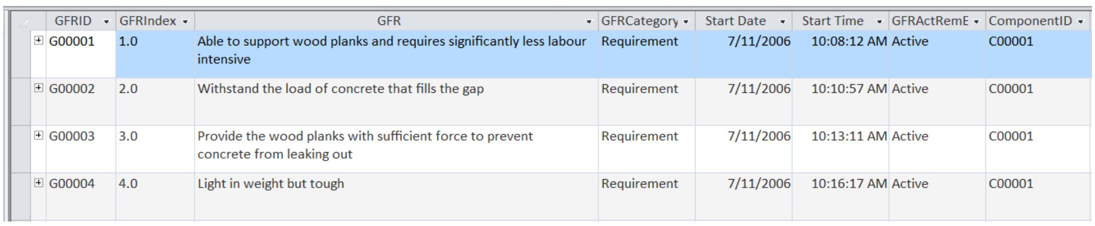
ProductID	ProductName	ProductGoal	DueDate	Click to Add
000001	Device to support a wood plank for concrete filling gap between flooring slabs	To design a device to replace the utilisation of steel wire and bar to support the wood plank that support concrete filling gap between flooring slabs	1/24/2006	

Table 5.2 A screen shot of the data captured in ComponentInfo table in by the descriptive design prototype software tool



ComponentID	ComponentName	Component	Click to Add
C00001	Device to support a wood plank that su	$0.20	
C00002	Cable Tie	$0.01	
		$0.00	

Table 5.3 A screen shot of some data captured in the GFRInfo table by the descriptive design prototype software tool



GFRID	GFRIndex	GFR	GFRCategory	Start Date	Start Time	GFRActRemE	ComponentID
G00001	1.0	Able to support wood planks and requires significantly less labour intensive	Requirement	7/11/2006	10:08:12 AM	Active	C00001
G00002	2.0	Withstand the load of concrete that fills the gap	Requirement	7/11/2006	10:10:57 AM	Active	C00001
G00003	3.0	Provide the wood planks with sufficient force to prevent concrete from leaking out	Requirement	7/11/2006	10:13:11 AM	Active	C00001
G00004	4.0	Light in weight but tough	Requirement	7/11/2006	10:16:17 AM	Active	C00001

CADInfo table (refer to Table 5.5) is the database that was created to store the pathname and some details of the related CAD files developed by designers to address IFRs. In addition to that, the table also stored a direct object link or Object Linking and Embedding (OLE) link to the actual CAD file where the designer can open the CAD file by double clicking the file. The field “CADLink” is the field that provides the OLE link to the actual CAD file. The CADInfo table did not store the actual file.

Similarly, PicDetail table (refer to Table 5.6) is created to store the pathname information and some details of the scanned images of design sketches or documents throughout the design process. The scanned images or documents can also be opened directly in a similar way for viewing by the designer via the OLE link. This table also did not store the actual images or the documents of the files. The LinkIFR table (refer to Table 5.7) stored information on linkages between IFRs or between IFRs and GFRs while the Participating Designer Table stored the name and affiliation of the designer involved in the project.

In this research work, the tool automatically creates the field “affiliation but the user interface did not provide any input feature for the designer to state their affiliation. In Case Study 1, there were two designers involved in the case study, one from the Manufacturing Engineering Centre (MEC) of Cardiff University and the other from the customer. The designer from the customer provided an initial sketch of his idea in this case study. Due to the confidentiality agreement, the name of the designer and the customer will only refer to the Designer from Customer. Table 5.8 illustrates the information about the participating designer captured in the ParticipatingDesigner table.

Table 5.4 A screen shot of a section of the IFRInfo table before Phase 1 was completed (noticed the WIP status)

IFRID	IFR	IFRCatID	IFRDate	IFRTime	SourceofE	Belie	IFRStatus	IFRActR(	PicDocID	Design	CADIn	Component	IFRAddRem	IFRAddRemDate
1.1	Create triangular shape device to position between slabs of beam	Requirement	7/11/2006	10:22:30 AM	Customer	0.75	Final	Active	0000001	00001		C00001	Need to check with construction experts on the detail shape of slab	13/12/2006
1.1.1	Device will protrude when gap is small (gap = 98mm or 148mm)	Issue/Information	7/11/2006	12:10:46 PM	Knowledge/ Experience	1	Solved	Active	0000002	00002		C00001		
1.1.1.1	Need to redesign the shape to prevent protrusion	Requirement	8/11/2006	12:34:32 PM	Knowledge/ Experience	0.75	Final	Active	0000002	00002		C00001		
1.2	Need telescopic device with locking mechanism to position	Requirement	10/11/2006	2:16:25 PM	Knowledge/ Experience	0.5	Abandoned	Removed		00002		C00001		
1.2.1	Need to estimate the cost	Requirement	10/11/2006	2:22:45 PM	Knowledge/ Experience	0.75	Abandoned	Removed		00002		C00001		
1.2.1.1	Need to determine how many components	Requirement	10/11/2006	2:23:57 PM	Knowledge/ Experience	0.5	Abandoned	Removed		00002		C00001		
1.2.1.1.1	Need to find out how much the three sliding components cost	Requirement	13/11/2006	2:57:44 PM	Knowledge/ Experience	0.75	Abandoned	Removed		00002		C00001		
1.1.1.1.1	Require a device with a flat square top with two rectangular wings	Requirement	16/11/2006	10:19:21 AM	Knowledge/ Experience	0.5	Final	Active	0000002	00002		C00001		
1.2.1.1.1.1	Require cost more than GBP 0.20	Constraint	17/11/2006	3:04:35 PM	Knowledge/ Experience	0.75	Abandoned	Removed		00002		C00001		
1.1.2	Need to determine the mechanism to pull the plank	Requirement	21/11/2006	10:32:12 AM	Knowledge/ Experience	0.75	WIP	Active		00002		C00001		
1.1.2.1	Require cables to pull the plank	Requirement	21/11/2006	11:21:43 AM	Knowledge/ Experience	0.5	WIP	Active		00002		C00001		
1.1.2.2	Need to determine what cable can be used	Requirement	21/11/2006	11:40:22 AM	Knowledge/ Experience	0.5	WIP	Active		00002		C00001		
1.1.2.2.1	Link to external cable tie design (Decide to use a cable tie)	Idea	22/11/2006	2:29:31 PM	Knowledge/ Experience	0.25	WIP	Active		00002		C00002		
1.1.2.3	Need to determine how many slots for cable ties	Requirement	23/11/2006	10:34:26 AM	Knowledge/ Experience	0.5	WIP	Active		00002		C00001		
1.1.2.3.1	Need one slot for a cable tie to go through (use 2 cable ties)	Requirement	23/11/2006	10:43:56 AM	Knowledge/ Experience	0.5	Abandoned	Removed		00002		C00001		
1.1.3	Need to determine max load for the cable tie	Requirement	24/11/2006	11:01:18 AM	Knowledge/ Experience	0.75	Final	Active		00002		C00001		

Table 5.5 A screen shot of the CADInfo table

CADInfoID	CADFilename	CADFileLocation	CADSys	CADLink	CADDescription	CADKeywo
000001	Device.sldprt	c:\project\device\	Solidworks	SolidWorks Part Document	Device to support the plank	device

Table 5.6 A screen shot of the PicDetail table

PicDocID	PicDocFilename	PicDocDescription	Pictype	PicDocPath	PictureorDocLink	Keyword
0000001	ConceptSketch.jpg	Sketch of conceptual solution from customer	Picture	c:\project\device\	Package	triangle shape
0000002	Protrusion problems.jpg	Sketch and evaluation of protrusion issue	Picture	c:\project\device\	Package	protrusion
0000003	X-wing study.jpg	Sketch of possible x-wing shape	Picture	c:\project\device\	Package	X-Wing

Table 5.7 A screen shot of the data captured in LinkIFR table of the database

LinkID	LinkType	From	To	LinkEffect	LinkRemark	LinkDate	LinkTime
000001	IFRtoGFR	IFR1.1.2.2.1	GFR3.0	Remark	Solution for GFR3.0	22/11/2006	2:45:03 PM
000002	IFRtoGFR	IFR1.1.6.1.1	GFR4.0	Remark	Solution for GFR4.	11/12/2006	4:25:32 PM
000003	IFRtoGFR	IFR1.1.6.4.1	GFR2.0	Remark	Solution for GFR2.0	22/12/2006	2:31:39 PM

Table 5.8 A screen shot of the data captured in ParticipatingDesigner table of the database

DesignerID	DesignerName	Affiliation
00001	Designer from Customer	
00002	Dr. Wang	

The prototype descriptive design software tool developed for this case study did not include the capacity to support the engineering contradiction matrix tool from TRIZ which will be utilised and elaborated further in the next chapter for case study 2 in the next chapter. The final design outcome from this case study is the plastic X-wing device shown in Figure 5.3. The device was developed in CAD by the designer and the CAD file has a link to the prototype software tool. The CAD model of the device shown in Figure 5.3 was the first version and the data captured in Phase 1 of the case study 1. The device was then manufactured and fitted in between beams as shown in Figure 5.4 to be tested by the customer. Later, the device was further improved to reduce weight, material and cost. The data captured in Phase 1 of the Case study 1 was retrieved to be improved in Phase 2 of the same case study. The next section will explain Case study 1 Phase 2. During the Case Study 1, only the text-based user interface was used as the graphical user interface was not available. The graphical user interface was completed in Phase 2 and the results of the data captured in Phase 1 were able to be displayed in Figure 5.5. Figure 5.5 shows a snapshot of the graphical user interface of the design process for this case study at the end of Phase 1.

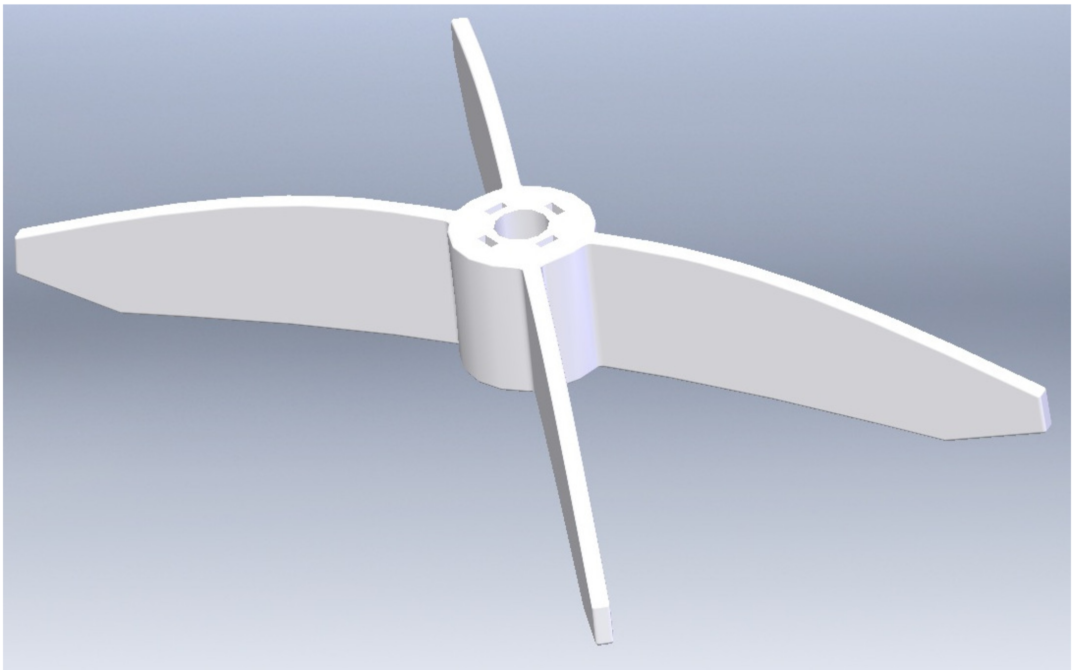


Figure 5.3 The CAD model of the device to support the concrete filling between beams

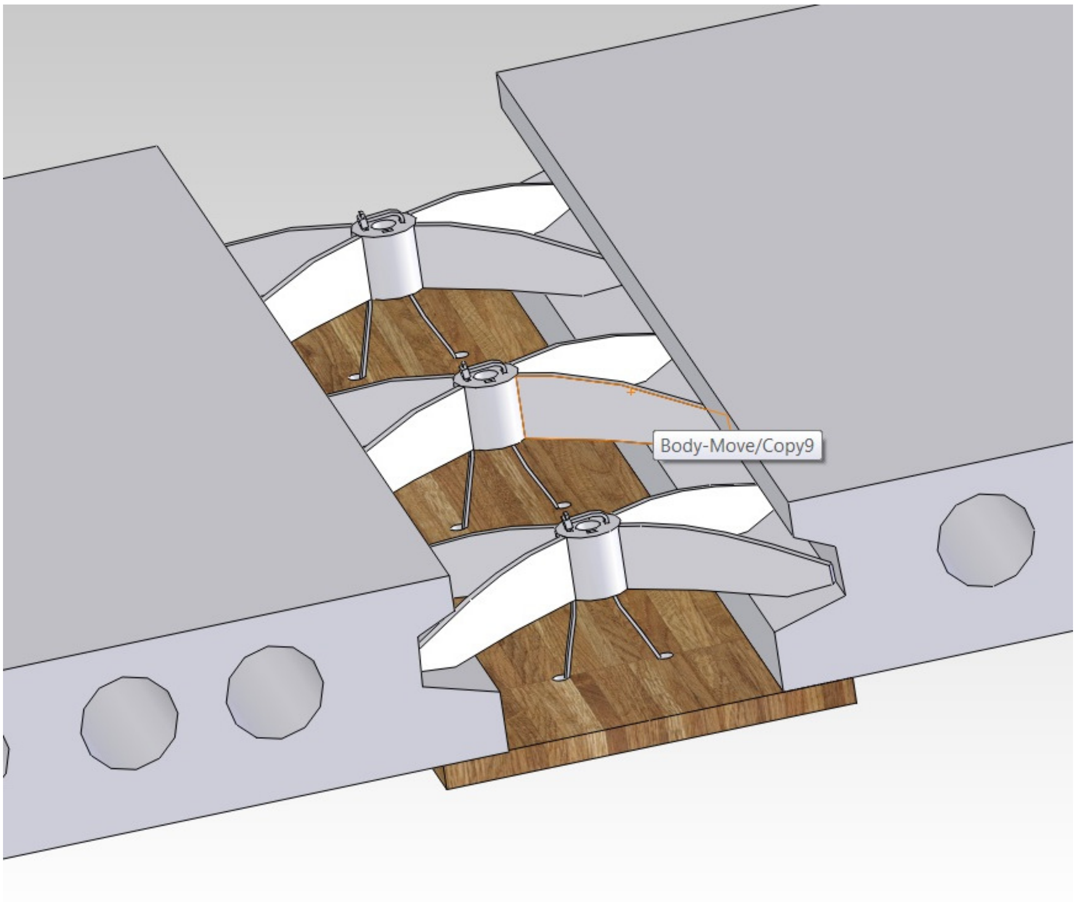


Figure 5.4 The concrete filling support device in operation (noticed the use of cable tie to pull the wood plank)

5.4 Implementation of the Descriptive Design Methodology in Case Study 1 (Phase 2)

Phase 2 of case study 1 is a continuation of the Phase 1. In this phase, the designer utilised the prototype descriptive to make some modifications to the device that supports concrete loading in between beam slabs. The modifications were made due to the request of the customer to reduce the cost of the device further to maximise profits. This request provided an opportunity for this research work to demonstrate and verify the importance of having a descriptive design framework (via the prototype tool) with computational-platform-related support in a design process.

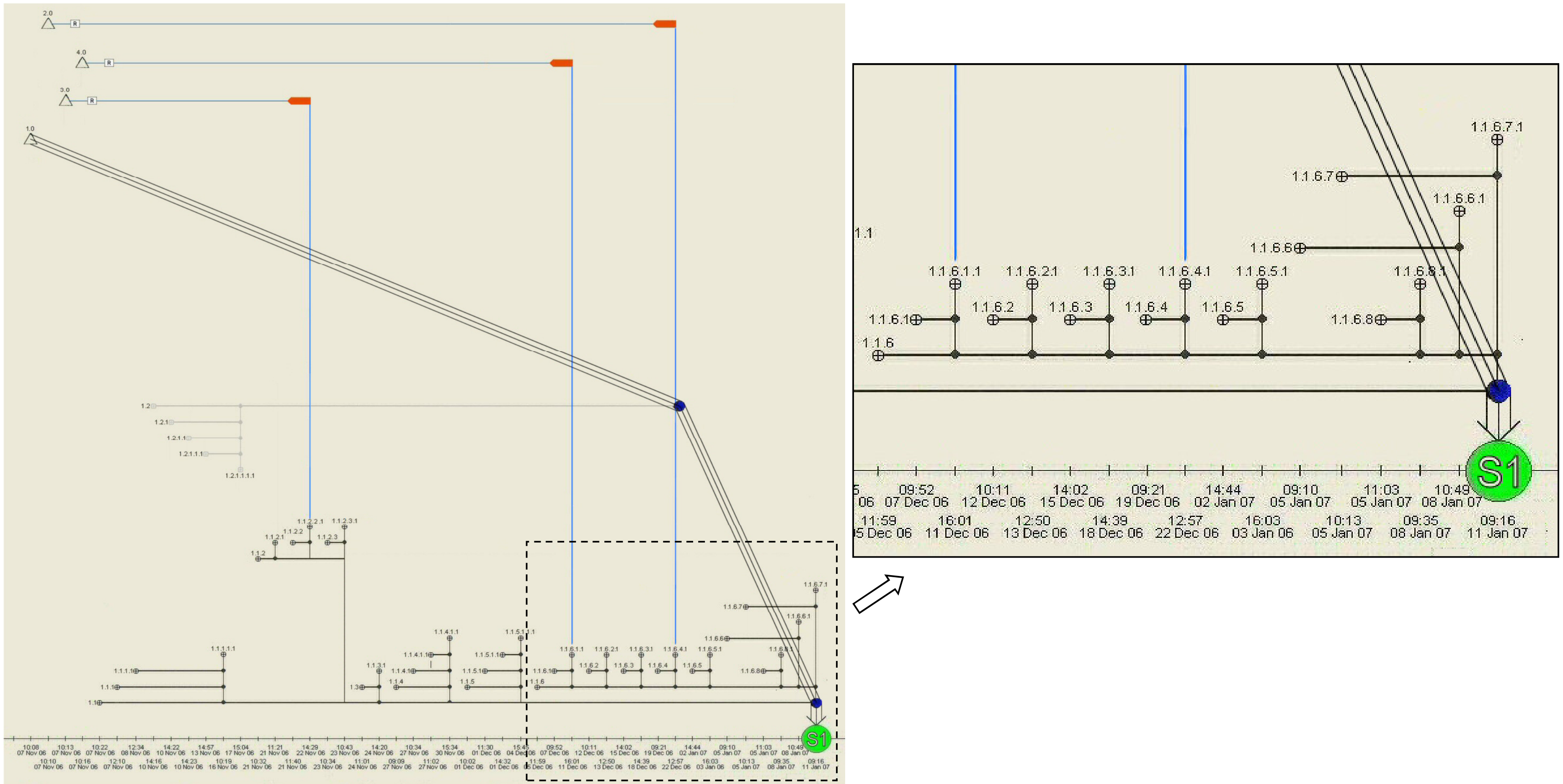


Figure 5.5 The graphical user interface of the descriptive design prototype software tool when Phase 1 of the case study 1 was completed with a zoom in view

5.4.1 The aims of the Case Study 1 (Phase 2)

The aims of this Phase 2 of case study 1 are:

i) to confirm the capability of the framework in retrieving past design data and to allow further addition of design data to support the late design modifications required in most design projects,

ii) to determine whether the descriptive design framework is able to track and trace design decisions made throughout the design process,

iii) to demonstrate how the descriptive design framework is able to support the computer-aided engineering (CAE) analysis process and how it manages the CAE analysis report,

iv) to show and verify the advantages of a flexible input interface that includes the text-based user interface and graphical-based user interface.

5.4.2 The details of implementation for Case Study 1 (Phase 2)

Phase 2 of case study 1 started when the customer requested further reduction of the device cost after a few days when the initial design was completed. The designer then considered the reduction of weight by removing some material from the current device. The data captured in the Phase 1 of the case study was retrieved and the designer provides further IFRs (ideas, information, sub-requirements, etc.) to achieve the reduction of cost requested by the customer mentioned above.

5.5 Results from the Implementation of the Descriptive Design Methodology in Case Study 1 (Phase 2)

Similar to the results obtained from Case Study 1 (Phase 1), the results of the implementation of Case Study 1 (Phase 2) were entirely captured in the database. The data captured by the descriptive design prototype tool were crucial and the results have demonstrated the flexibility of the framework in adopting design changes as well as additional management of design data such as CAE analysis results.

5.5.1 The database information captured in Case Study 1 (Phase 2)

The data capturing in Case Study 1 (Phase 2) was a continuation from Phase 1 and Table 5.9 highlighted the additional data captured (in blue) in the IFRInfo table for Phase 2. A new CAD model of the improved device was created and linked to the CADInfo table as shown in Table 5.10. In addition to that, this new CAD model then underwent CAE analysis to ensure that the device will not fail due to the addition of slots and a through-all concentric hole. The structural analysis process created a report that contained the results of the analysis. The requirement to perform such an analysis to prevent failure was captured by the prototype tool and the analysis report details were stored in PicDetail table as shown in Table 5.11. The new improved CAD model of the improved device is shown in Figure 5.6.

As noted in Figure 5.6, the device has some slots removed and the concentric hole at the centre of the device extended from half way to a through hole. The removal of material saves cost and reduces the weight of the device but may compromise the strength of the device. Hence, there is a negative effect of IFR 1.1.7.1 (Remove material on the device) to the GFR 2 (Withstand the load of concrete that fills the gap). The LinkIFR table for Phase 2 shown in Table 5.12 indicates the effects of IFR 1.1.7.1 to GFR 2.

Table 5.9 A screen shot of the IFRInfo table with highlighted additional IFRs to improve the device in the Phase 2 of the case study 1

IFRID	IFR	IFRCatID	IFRDate	IFRTime	SourceofE	Belie	IFRStatus	IFRActR	PicDocID	Design	CADIn	Component
1.1.6.8.1	Size of slot determined using CAD	Solution	8/1/2007	9:35:08 AM	Knowledge/ Experience	0.75	Final	Active		00002	000001	C00001
1.1.6.6.1	Decide the location for the slot to be at the middle of the central part	Solution	8/1/2007	10:49:58 AM	Knowledge/ Experience	0.75	Final	Active		00002		C00001
1.1.6.7.1	Shape determined using CAD	Solution	11/1/2007	9:16:25 AM	Knowledge/ Experience	0.5	Final	Active		00002	000001	C00001
1.1.7	Need to reduce the cost of the device but maintain the properties	Requirement	16/1/2007	9:45:03 AM	Customer	0.75	Final	Active		00001		C00001
1.1.7.1	Remove materials on the device	Idea	16/1/2007	12:32:56 PM	Knowledge/ Experience	0.75	Final	Active		00002		C00001
1.1.7.1.1	Create slots on the wings on the device	Idea	16/1/2007	2:36:22 PM	Knowledge/ Experience	1	Final	Active		00002		C00001
1.1.7.1.2	Extend the current concentric slot to a through-all slot	Idea	16/1/2007	2:56:22 PM	Knowledge/ Experience	1	Final	Active		00002		C00001
1.1.7.1.1.1	Determine size of slot using CAD	Solution	17/1/2007	10:13:42 AM	Knowledge/ Experience	0.75	Final	Active		00002	000002	C00001
1.1.7.1.2.1	Model the concentric slot using CAD	Solution	18/1/2007	9:08:49 AM	Knowledge/ Experience	0.75	Final	Active		00002	000002	C00001
1.1.7.1.3	Need to perform CAE analysis on the improved device	Requirement	19/1/2007	9:05:37 AM	Knowledge/ Experience	1	Final	Active		00002		C00001
1.1.7.1.3.1	CAE analysis results proved improved device OK	Solution	22/1/2007	3:38:52 PM	Knowledge/ Experience	0.75	Final	Active	0000004	00002		C00001
1.1.7.2	Need to reduce number of cable ties to one	Requirement	23/1/2007	9:40:01 AM	Knowledge/ Experience	0.25	Final	Active	0000005	00002		C00001
1.1.2.3.2	Need to increase slots to four to cater different orientations using	Requirement	23/1/2007	11:53:22 AM	Knowledge/ Experience	0.25	Final	Active	0000005	00002		C00001

Table 5.10 A screen shot of the CADInfo table with an additional CAD file in the Phase 2 of the case study

CADInfoID	CADFilename	CADFileLocation	CADSys	CADLink	CADDescription	CADKeyword
000001	Device.sldprt	c:\project\device\	Solidworks	SolidWorks Part Document	Device to support concrete filling	device
000002	ImprovedDevice.sldprt	c:\project\device\	Solidworks	SolidWorks Part Document	Improved version of the device to support concrete filling	improved device

Table 5.11 A screen shot of the PicDetail table with an additional doc file of the analysis report in the Phase 2 of the case study

PicDocID	PicDocFilename	PicDocDescription	Pictype	PicDocPath	PictureorDocLink	Keyword
0000001	ConceptSketch.jpg	Sketch of conceptual solution from customer	Picture	c:\project\device\	Package	triangle shape
0000002	Protrusion problems.jpg	Sketch and evaluation of protrusion issue	Picture	c:\project\device\	Package	protrusion
0000003	X-wing study.jpg	Sketch of possible x-wing shape	Picture	c:\project\device\	Package	X-Wing
0000004	Improved device analysis-SimulationXpress Study-1.doc	Simulation analysis file after improvement	Document	c:\project\device\	Microsoft Word Document	simulation
0000005	Design with one cable tie.jpg	Sketch of design change with one cable tie	Picture	c:\project\device\	Package	cable tie

Table 5.12 A screen shot of the LinkIFR table with added links in Phase 2

LinkID	LinkType	From	To	LinkEffect	LinkRemark	LinkDate	LinkTime
000001	IFRtoGFR	IFR1.1.2.2.1	GFR3.0	Remark	Solution for GFR3.0	22/11/2006	2:45:03 PM
000002	IFRtoGFR	IFR1.1.6.1.1	GFR4.0	Remark	Solution for GFR4.	11/12/2006	4:25:32 PM
000003	IFRtoGFR	IFR1.1.6.4.1	GFR2.0	Remark	Solution for GFR2.0	22/12/2006	2:31:39 PM
000004	IFRtoGFR	IFR1.1.7.1	GFR2.0	-	May weakened the device; Need CAE analysis	16/1/2007	3:11:29 PM
000005	IFRtoIFR	IFR1.1.7.1.3.1	IFR1.1.7.1	Remark	Proved improved device OK	22/1/2007	4:29:02 PM
000006	IFRtoIFR	IFR1.1.7.2	IFR1.1.2.3.1	Remark	Remove IFR1.1.2.3.1	23/1/2007	10:11:45 AM

The graphical user interface also provides an additional display that showed the negative link effects of IFR 1.1.7.1 to GFR 2.0. However, the following IFR 1.1.7.1.3, which is a requirement to perform CAE analysis that resulted in IFR 1.1.7.1.3.1 (CAE analysis results proved improved device is OK), provides a solution to eliminate the negative effects of IFR 1.1.7.1 to GFR 2.0. Similarly, IFR 1.1.7.2 (reduce the number of cable tie from two to one) was proposed to reduce cost (IFR 1.1.7) and this decision affected the earlier decision (IFR 1.1.2.3.1), which was to create one slot to cater for the idea of using two cable ties to secure the support of the wood plank. This led to the creation of a link to remove the IFR 1.1.2.3.1 and the adoption of IFR 1.1.2.3.2, where 4 slots were created to accommodate the different orientations of the device in operation.

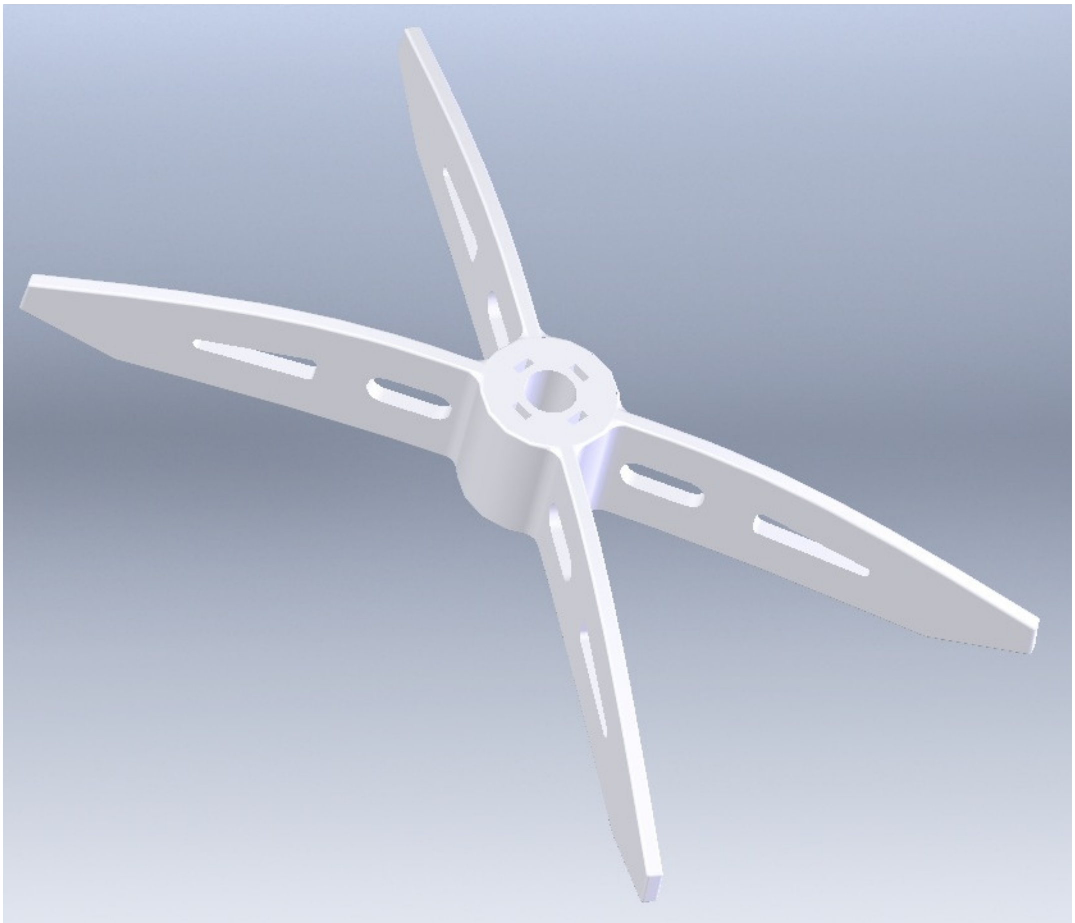


Figure 5.6 The improved concrete filling support device

5.6 Results from the Interview with the Designer on the Descriptive Design Methodology in Case Study 1 (Phase 1 and Phase 2)

The interview with the designer took place at the end of the Case study 1 after Phase 1 and Phase 2 had ended. The interview was about half an hour and the questions asked were simple ones, which were intended to obtain the opinions of the designer on the tool specifically and framework generally. The interview was conducted with a feedback form, in which questions based on the form were asked and the answers jotted down.

The questions raised during the interview are shown in the feedback form shown in Appendix 1. Based on the answers received, the designer found the descriptive design software tool good and helpful. The designer also stated that he did not practise any design methodology and will probably use the software tool in the future. According to the designer, among the strengths of the descriptive design software tool was the ability to allow him to visualise his design work, path and direction using a computer, which is not possible if he uses a log book. In addition to that, the software tool also allowed him to review and make changes to his design decisions due to the availability of the graphical user interface as well as helping him to improve tracking and tracing of his documentations.

The designer agreed that the combination of text-based and graphical-based user interfaces provides better flexibility for him to input than just the text-based user interface. However, the designer also thought that the descriptive design software tool could be improved by adding a voice recording feature, made available in mobile phones or personal digital assistant and linked up to a digital sketch pad. Currently, all images are scanned into the computer before the linkages with the descriptive design software tool are created. Lastly, the designer felt that he is still more used to writing in the log book. Hence, the use of a digital writing pad rather than typing via keyboard into his computer would be preferred.

5.7 Analysis and Discussions of the Implementation of the Descriptive Design Methodology and the Results Obtained in Case Study 1 (Phase 1 and Phase 2) – Design of concrete filling support device for gap between beam slabs case study

The analysis and discussion were partitioned into two sections with the first section focused on assessing the results from case study 1 (Phase 1) from the perspective of the aims stated in the case study 1 (Phase 1). The second section (section 5.6.2) will focus on the results obtained in the Phase 2 of the case study, to review how well the descriptive design framework fulfilled the aims of the case study 1 (Phase2).

The analysis of this case study were based on observations noted during the period of 2 weeks during the initial phase of the design as well as the data inputs provided by the designer directly into the prototype software tool. The data inputs were analysed by retrieving them from the database repositories shown in Table 5.1 to Table 5.8. In addition, the analysis also included information from the feedback provided by the designer via occasional visits (twice a week) to consult and interview the designer throughout the project. The design project for this case study took several months to complete.

During the implementation of the descriptive product design tool, the designer was found to be handling several projects. He was observed occasionally referring back to his documentation, calculations, catalogues, reports and literature related to current projects. It is often time-consuming to find the related documents and occasionally the designer has to spend time to recall back the right information from the appropriate documentation prior to making any design decisions. Hence, it is not surprising, from observations and the feedback from the interview, that the descriptive product design tool provided a centralised mechanism for him to recall and refer to the appropriate documentation to assist his design decision making. The descriptive product design tool significantly improved his effectiveness in obtaining the appropriate information by reducing the risks of getting the incorrect information from the wrong documents, which can lead to erroneous decisions.

Further feedback from the interviews with the designer also found that the proposed tool, particularly the graphical user interface, enabled the designer to reflect back more effectively on his previous design decisions and to re-evaluate the requirements, sub-requirements, constraints and criteria that were made earlier. The designer was observed to perform frequent reviews by pondering on his earlier proposed IFRs with respect to the GFR and attempts to determine the effects of any changes of his past IFRs to the outcome of future IFRs. The graphical user interface also provided a mean for the designer to systematically decompose "given functional requirements" (GFRs) into "introduced functional requirements" (IFRs) and sub-IFRs and then to address the solutions to solve the sub-IFRs, which then led to the solving of the main IFRs and finally the GFR. The designer was also able to go back and forth, removing and adding sub-IFRs such as sub-requirements, constraints and criteria when new information was received, which can make ideas, thoughts and decisions made earlier inappropriate, incorrect or inaccurate. This is crucial and in this case study, the designer tried to redesign the X-shape device to include the required cost savings in Phase 2 of the case study such as the change to the utilisation of the number of cable ties from to two to one. The designer proposed the use of two cable ties initially but later reduced it to one with the addition of slots on the device in Phase 2 of the case study.

At the end of the re-designing process, after searching and exploring many of his earlier design decisions, he also managed to improve the X-shape device by reducing its weight. The reduction of weight was focused around the central part of the device where a hole of 25mm was bored and two additional rectangular slots along the perimeter of the central part were added.

The case study also showed that the designer made a decision to abandon the telescopic device idea after pondering several issues relating to it. The tool was also able to capture his approach of proceeding into the detail design of the X-shape device in the CAD tool before going back to the conceptual design phase to finalise his decision towards the appropriate mechanism to support the wood plank as shown in Figure 5.4. He later continued his detail design of the tapered part of the X-shape device that fits and in contact with beam slabs. The process of going back and forth from conceptual phase and the detail design phase demonstrates the flexibility of the methodology in accommodating the preference and approach of the designer. Many IFRs have a link to one or more documents, which involves sketches, calculations and documentations. The designer was observed to produce several sketches as he provided his ideas and thoughts throughout the design process. Several IFRs may link to a single document and vice versa. Though the proposed tool was found to be able to provide flexible and systematic support to the designer, the search engine for the tool is still very basic and is only able to match a single word at a time.

For this case study, the design problem demonstrates the importance of the ability and creativity of the designer to derive a totally new concept for a device to replace the existing method. The design problem in this case study did not proceed into the complexity of mechanism, kinematics and components interactions solely because of the ability of the designer to create this simple effective approach to the problem. If the idea of using a telescopic device was to be continued, there would have been such complexities.

The proposed descriptive design methodology is solely dependent on the designer's knowledge and experience to derive an effective design solution. In addition, the design approach of the designer in this case study was consistent with the findings on how an experienced designer works. Experienced designers were found in the empirical study to have a tendency to build on one or two ideas and improve them to achieve the final design rather than to explore a large amount of solution space by generating many ideas or alternative concepts and then deciding on the best one (Badke-Schaub 2003). For future work, additional support to provide some ideas and solutions to requirements and sub-requirements of a design problem would be useful. Additional support that could provide ideas or solutions such as TRIZ and patent search could also be explored. It was also noted while implementing this case study that there was additional time consumed to scan or digitise the appropriate sketches and for the designers to provide their ideas and thoughts input into the tool. It would be significant time-saving if a sketchpad with a digitising pen as well as the documentation is done via computer instead of using pen with paper or book and this is consistent with feedback from the designer via interview.

Finally, the prototype tool is unable to allow the designer to capture their ideas and thoughts when he is not using a computer. The current tool requires the designer to memorise some of their ideas and the thoughts that come up during the period when they are not in front of the computer and then recalls them to be feed into the tool when they are back at their computer. A good way to solve the problem of the designer not being in front of their computer is to develop a simplified version of this tool on handheld devices or personal digital assistance (PDA). However, even with a PDA, this methodology, like any other methodology, depends heavily on the willingness of the designer to use it i.e. in this case to provide his ideas and thoughts.

5.7.1 Analysis and discussion of the results obtained in Case Study 1 (Phase 1)

Based on data captured from the case study and the interview, the analysis was focused on whether the aims of this case study (Phase 1 and Phase 2) have been achieved. In order to determine whether the descriptive design framework is able to assist and support a designer that designs with their own approach, the feedback from the interview was crucial. Based on the feedback from the designer via the interview, the designer did not use any systematic design approach or methodology. In addition to that, the feedback also revealed that the designer found the design methodology-related support and the computational-platform-related support facilities in the descriptive design software tool helpful to him. The graphical user interface offered crucial assistance and support in providing him with a view of the progress, path and direction of his design work.

For the second aim of the case study, ascertaining the ability of the descriptive design framework to capture the thoughts and ideas throughout the design process, this will be based on the data captured in the database using the prototype tool. In reality, a designer's thoughts and ideas can come in several forms. A designer can have thoughts and ideas, which they can express (by speech or actions) or they may not express them (tacit). If they expresses their thoughts and ideas, they may express them using sketching to represent their thoughts and ideas or they may model them using a computer-based tool such as a computer-aided design (CAD) tool. In all cases, the descriptive design framework must be able to capture these various forms of thoughts and ideas from the designer. Analysing the data captured in database from the prototype software tool will provide key evidence to the fulfilment of this aim. From the data captured in the database throughout the design process to create the new device to support the concrete loading, the designer developed sketches during the early phases of the design and then performed calculations on determining the appropriate shape, thickness and length of the device to fit into the gap in between beam slabs and still be able to support the concrete load. These calculations were performed on paper and sometimes in an Excel worksheet. The designer later moved his design into computer-aided design and then performed finite element analysis and simulations on his design. Modifications could be made throughout this process if he wished to improve his models based on feedback from various analyses and simulations. All his CAD and simulation models were filed within the CAD and simulation tool itself but were linked to the descriptive design tool. The current links to simulation and CAD models did not involve any direct interaction with the data from CAD and the simulation models.

However, the descriptive design tool enables the designers to know the respective CAD and simulation files that are linked to their solutions related to “introduced functional requirements” (IFR) and later to “given functional requirements” (GFR). The designer can also open the related CAD file and simulation files by double clicking the name of their respective files stated earlier in the descriptive tool, in which the CAD tool or the simulation tool will be launched.

In order to ascertain whether the descriptive design framework can capture the thoughts and ideas of a designer throughout the design process, a close look at the data captured from the case study is important. The initial “introduced functional requirement” for the original GFR (to take the load of the concrete) was to use a triangular shaped device to fit between the gap and this was proposed by the customer (see Figure 5.7). Hence, the designer considered this as a requirement (refer to IFR1.1 of Table 5.4). However, later the designer thought of trying something that had a telescopic feature to extend to fit itself in between the gap of two beams. Further consideration of this design was later abandoned when it was found to be quite expensive and complicated. Table 5.13 shows the portion of the data captured for these changes in design decisions.

Table 5.13 A screen shot of the data captured for the thoughts about the changing the design from a device with triangular shape to a device with telescopic features highlighted in IFRInfo table

IFRID	IFR	IFRCatID	IFRDate	IFRTime	SourceofE	Belie	IFRStatus	IFRActR	PicDocID	Design	CADIn	Component
1.1.7.2	Need to reduce number of cable ties to one	Requirement	23/1/2007	9:40:01 AM	Knowledge/ Experience	0.25	Final	Active	0000005	00002		C00001
1.2	Need telescopic device with locking mechanism to position	Requirement	10/11/2006	2:16:25 PM	Knowledge/ Experience	0.5	Abandoned	Removed		00002		C00001
1.2.1	Need to estimate the cost	Requirement	10/11/2006	2:22:45 PM	Knowledge/ Experience	0.75	Abandoned	Removed		00002		C00001
1.2.1.1	Need to determine how many components	Requirement	10/11/2006	2:23:57 PM	Knowledge/ Experience	0.5	Abandoned	Removed		00002		C00001
1.2.1.1.1	Need to find out how much the three sliding components cost	Requirement	13/11/2006	2:57:44 PM	Knowledge/ Experience	0.75	Abandoned	Removed		00002		C00001
1.2.1.1.1.1	Require cost more than GBP 0.20	Constraint	17/11/2006	3:04:35 PM	Knowledge/ Experience	0.75	Abandoned	Removed		00002		C00001

Then the design process continued to evolve from the initial design of a triangular shape which was found to be protruding on top of the floor surface if the gap in between beams were smaller. This is because the gap between beams can be in two different length: 98 mm or 148 mm. Hence, the solution was to have a device with wings and shaped like “X” and that differed in dimensions so that when placed at a different orientation, the device will be able to fit into the gap (refer to Figure 5.8).

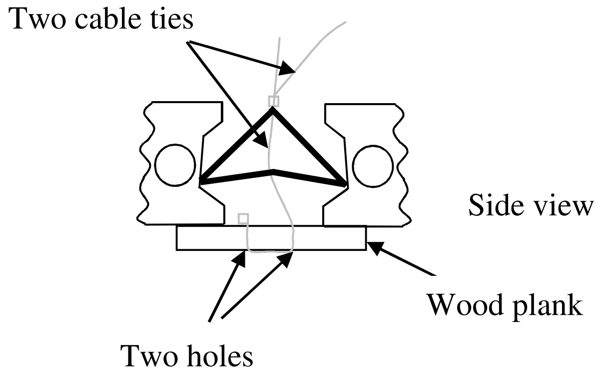


Figure 5.7 The initial idea of using a triangular device with two cable ties suggested by the customer

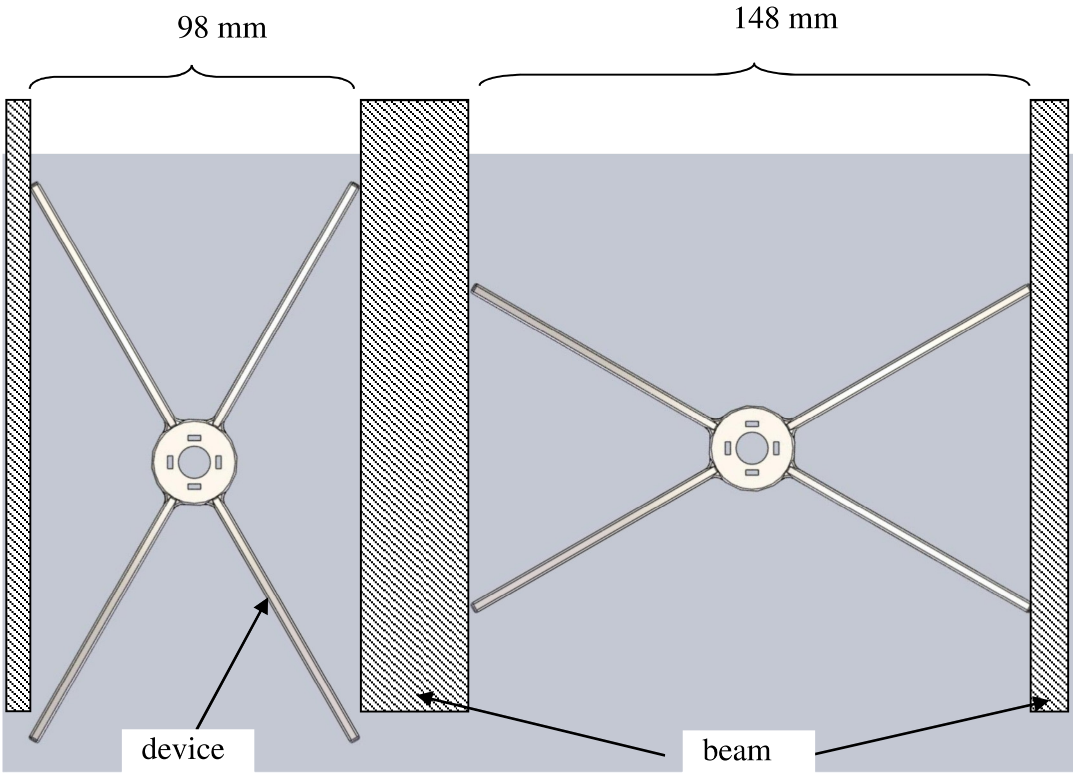


Figure 5.8 The different orientation of the newly designed concrete loading support device allows it to accommodate different sizes of gaps in between beams (viewed from the top)

The X-winged device worked with a single plastic cable tie to replace the steel bars and the steel cables. However, the method still retained the use of the wood planks. The cable tie would pass through the pair of holes on the wood plank at one end and at the other end passed through the triangular device as shown in Figure 5.4. Upon pulling the cable tie, the wood plank would be forced against the beams and this force was expected to prevent leaking of the concrete as well as supporting weight of the concrete. There were several devices used at an interval of fixed distance along the wood plank as shown in Figure 5.4. Though initially the use of two cable ties were required based on a suggestion by the customer, later the designer improved the device to use a single cable tie by creating 4 slots at the middle cylindrical part of the device. This further reduces the cost and the single cable tie has sufficient specification to deal with the load (based on the datasheet of the cable tie).

The literature study in Chapter 3 showed that it is advantageous for a designer to delay making design decisions if there is insufficient information available. As shown in Table 5.13, there was a need to check with the experts in the construction industry to confirm shape of the slabs which the device needs to fit in. As this information was only available at 13th December 2006 when the designer was able to meet the expert, the decision to decide on the shape of the triangular device was delayed until then. The prototype software allows the designer to specify by adding remarks on the issues and reminders can be set to enable the tool to remind him. The remarks and the reminder date are captured in the database (refer to the highlighted data). Note that when a design project is completed, the IFRStatus should be changed to “Final” as shown in Table 5.13.

As this design project lasted several months, the prototype tool enabled the designer to save and retrieve his design decisions throughout the design process and indeed to allow the addition and deletion of design decisions. The IFRInfo table has a field known as IFRActRemEdStatus. This field is created to record any removal or editing of an introduced functional requirement (IFR). However, the status of the edited or removed IFR will only be recorded if the user has decided to edit or remove it after pressing the “OK” button on the IFR user interface (refer to Figure 4.6 in Chapter 4) or if the user has proceeded to create the next IFR. Any changes made during the typing in of the IFR will not be captured in the database. If this field is “Active”, this means that the data is currently in use.

The final aim of the case study (Phase 1) was to evaluate whether the descriptive design framework is able to perform some form of search on design data stored via the prototype tool. The user interface for the prototype tool allows the designer to perform a simple search to match words in the database. The searching is performed via creating a simple query of the database that captured the data throughout the design process and to return the data records with the word that was searched.

The search engine in this case study highlighted only several solutions based on the word searched by the designer. A more powerful and intelligent search engine is needed for a more effective search for complicated and large projects, to avoid too many highlighted solutions to be displayed. A search engine of this type is outside the scope of this research and should be explored in future work, particularly a search engine for a complex tree structure.

Table 5.14 The data captured (remarks and the reminding date) which allows the designer to delay their design decisions when they encountered insufficient information (highlighted row)

IFRID	IFR	IFRCatID	IFRDate	IFRTime	SourceofE	Belie	IFRSt	IFRActR	PicDocID	Design	CADIn	Component	IFRAddRem	IFRAddRemDate
1.1	Create triangular shape device to position between slabs of beam	Requirement	7/11/2006	10:22:30 AM	Customer	0.75	Final	Active	0000001	00001		C00001	Need to check with construction experts on the detail shape of slab	13/12/2006
1.1.1	Device will protrude when gap is small (gap = 98mm or 148mm)	Issue/Information	7/11/2006	12:10:46 PM	Knowledge/ Experience	1	Solved	Active	0000002	00002		C00001		
1.1.1.1	Need to redesign the shape to prevent protrusion	Requirement	8/11/2006	12:34:32 PM	Knowledge/ Experience	0.75	Final	Active	0000002	00002		C00001		
1.2	Need telescopic device with locking mechanism to position	Requirement	10/11/2006	2:16:25 PM	Knowledge/ Experience	0.5	Abandoned	Removed		00002		C00001		
1.2.1	Need to estimate the cost	Requirement	10/11/2006	2:22:45 PM	Knowledge/ Experience	0.75	Abandoned	Removed		00002		C00001		
1.2.1.1	Need to determine how many components	Requirement	10/11/2006	2:23:57 PM	Knowledge/ Experience	0.5	Abandoned	Removed		00002		C00001		
1.2.1.1.1	Need to find out how much the three sliding components cost	Requirement	13/11/2006	2:57:44 PM	Knowledge/ Experience	0.75	Abandoned	Removed		00002		C00001		
1.1.1.1.1	Require a device with a flat square top with two rectangular wings	Requirement	16/11/2006	10:19:21 AM	Knowledge/ Experience	0.5	Final	Active	0000002	00002		C00001		
1.2.1.1.1.1	Require cost more than GBP 0.20	Constraint	17/11/2006	3:04:35 PM	Knowledge/ Experience	0.75	Abandoned	Removed		00002		C00001		
1.1.2	Need to determine the mechanism to pull the plank	Requirement	21/11/2006	10:32:12 AM	Knowledge/ Experience	0.75	Final	Active		00002		C00001		
1.1.2.1	Require cables to pull the plank	Requirement	21/11/2006	11:21:43 AM	Knowledge/ Experience	0.5	Final	Active		00002		C00001		
1.1.2.2	Need to determine what cable can be used	Requirement	21/11/2006	11:40:22 AM	Knowledge/ Experience	0.5	Final	Active		00002		C00001		
1.1.2.2.1	Link to external cable tie design (Decide to use a cable tie)	Idea	22/11/2006	2:29:31 PM	Knowledge/ Experience	0.25	Final	Active		00002		C00002		

5.7.2 Analysis and discussion of the results obtained in Case Study 1 (Phase 2)

Phase 2 of case study 1 was part of the same project as Phase 1. After an initial design of the device to support concrete loading in between gaps of beams was completed and submitted to the customer, the customer requested modifications of the design to reduce the cost of making the device five days later. Hence, the Phase 2 of case study 1 involved using the past data from Phase 1. Due to the ability of the descriptive design prototype tool to save and retrieve past data, the designer was able to utilise past data and continue to modify the design to reduce the cost of the device for the customer. The difference between Phase 1 and Phase 2 was that the descriptive design prototype tool had added the graphical user interface instead of just the text-based user interface.

With the additional user interface, the designer has additional support in doing his work in the form of a graphical display. When the designer retrieved the data from Phase 1, the data could be displayed in a graphical form via the graphical user interface. The designer could visualise the connections and relationships between design decisions throughout the design process in Phase 1 better with the graphical user interface, as shown in Figure 5.6. Note the difference between Figure 5.6, which represented the final screen snapshot of the graphical user interface at the end of Phase 1 and the screen snapshot of the graphical user interface shown in Figure 5.9 when the Phase 2 was completed. With the retrieval and re-utilisation of data from Phase 1, further modifications were made much faster when additional requirements were requested. This means that the framework allows retrieval and addition of data to support late design modifications even after the design project has been completed.

From the perspective of tracking and tracing, the architecture of the descriptive design framework in providing the visualisation of the entire design process in terms of their activities, decisions and the documentation involved is a critical factor in enabling the tracing and tracking features. Based on findings from the interview with the designer, the designer acknowledged that the tracking and tracing of design decisions were better with the graphical user interface. In the graphical user interface, the paths created by GFR and IFR links as well as other links in relation to time were displayed as shown in Figure 5.9. This graphical display improved his tracking and tracing of his documentation. Each sketching image, document and CAD file are linked to the appropriate decisions made via IFRs and these linkages enable the designer to know which documentation has been created to support their design decisions and they can review this documentation quickly if required to reconsidered any of their design decisions.

Other than the feedback from the interview about the tracking and tracing of the design decisions made by the descriptive design framework, the design changes made when the graphical user interface shown in Figure 5.9 also showed how the framework dealt with the design changes as mentioned earlier about the idea of using a telescopic device. This idea was later abandoned. In Figure 5.9, the information of the category type of IFR and the status of IFR can be obtained by moving the mouse near the node of each IFR. Note that the abandoned IFRs are in grey and the GFR 1, 2, 3, and 4 were merely represented by 1.0, 2.0, 3.0 and 4.0.

Later, in order to reduce cost in Phase 2 of the case study, the designer also reduced the number of cable ties used from two to one. His decision to change the usage of cable tie was captured in the database. Hence, by analysing the database, the changes of design decision on this matter can be observed. The importance of capturing such design decisions is that the existence of the four slots in the middle of the device can be traced back to this reason. To understand better, it is essential to analyse the database related to this matter.

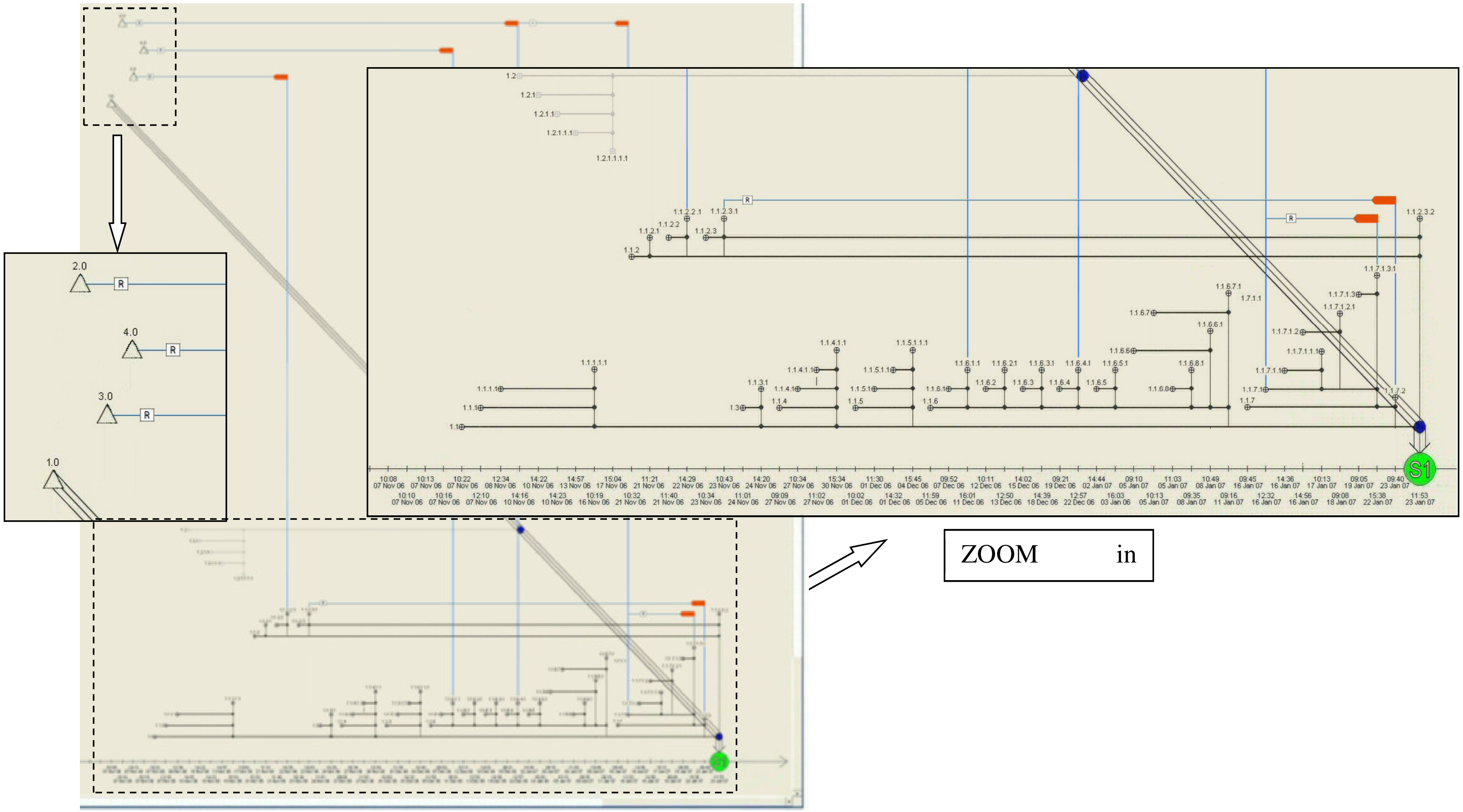


Figure 5.9 The snapshot of the graphical user interface of the descriptive design prototype tool when Phase 2 of Case Study 1 was completed with a zoom in view of the corners.

By analysing the database, the designer made a decision to reduce the number of cable ties used (IFR 1.1.7.2) as a further method to reduce cost based on the request of the customer (IFR 1.1.7). Due to the decision made earlier to use two cable ties, a two slot were created on the device for the cable tie to be secured as shown in Figure 5.10(a). When the designer decided to reduce the cable ties used from two to one, the number of slot on the device increased four as shown in Figure 5.10(b). The decision to reduce the cable tie used to one is linked to a sketch (PicDocID: 0000005). The sketch contained the design sketch of the device for using a cable tie to secure the wood plank. As shown in Figure 5.8, when the number of cable ties was reduced to one, the number of slots at the centre of the device has to be increased to four to allow the two possible orientations for the device to fit into the gap between the beams.

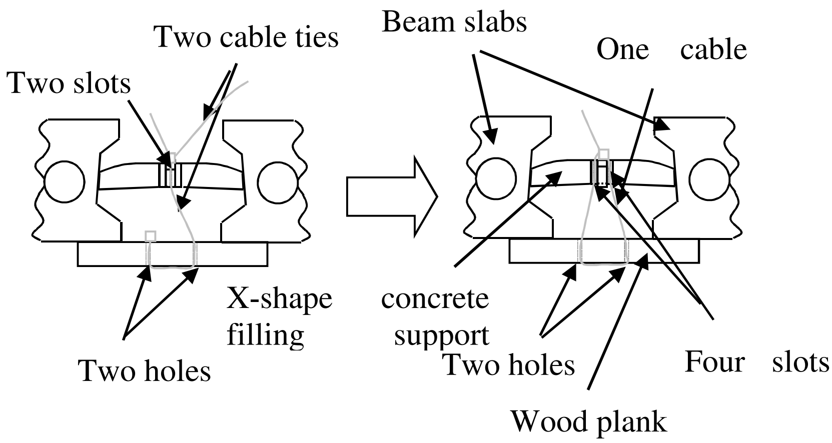
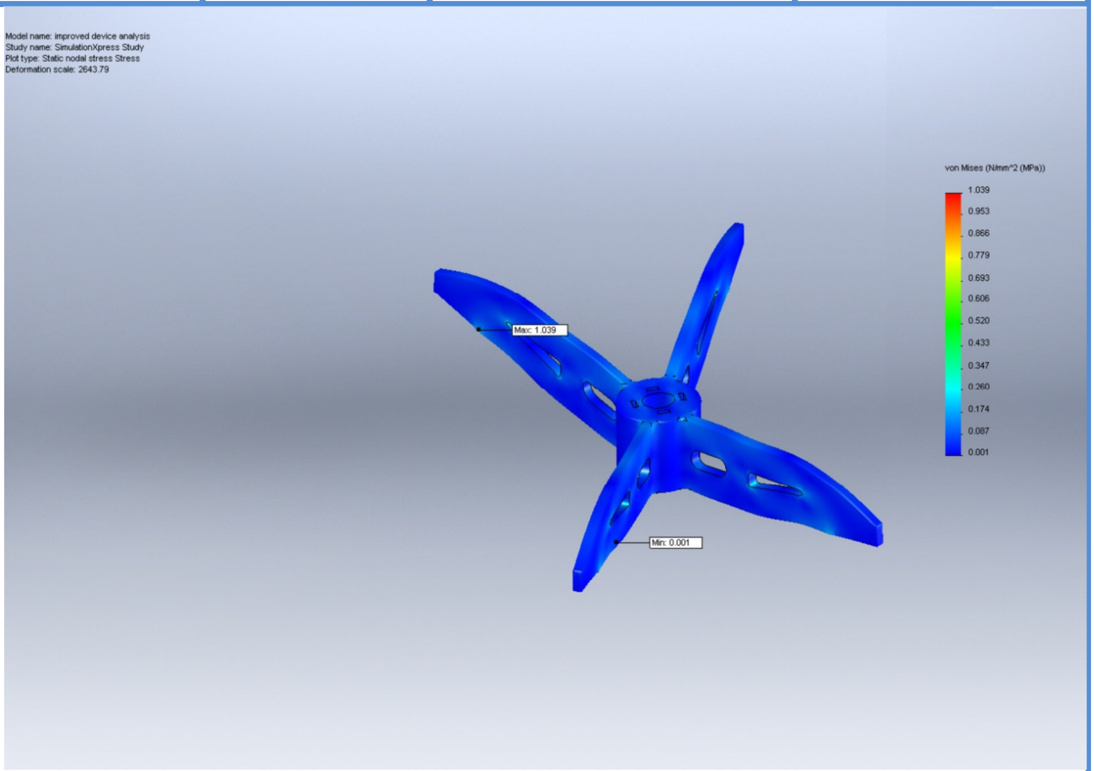


Figure 5.10 Design change involved when the number of cable ties used was reduced from two to one

The results from Phase 2 of the case study 1 also demonstrated that the descriptive framework is able to support the utilisation of computer-aided analysis (CAE) process and manages the outcomes of the CAE process to improve the device. Table 5.10 shows that the documentation of the CAE analysis results link the improved design of device (LinkID 000004 and LinkID 000005). Figure 5.11 illustrates the CAE results (von Mises stress) on the documents generated by the CAE tool.

Name	Type	Min	Max
Stress	VON: von Mises Stress	0.000658988 N/mm^2 (MPa) Node: 16454	1.03909 N/mm^2 (MPa) Node: 10898



improved device analysis-SimulationXpress Study-Stress-Stress

Figure 5.11 One of the CAE analysis results in the documentation captured by the descriptive design framework

Finally, the last aim of the Phase 2 was to find out whether the descriptive design prototype software tool with both text-based and graphical-based user interface had sufficiently offered a flexible input interface to the designer. Based on the interview, the designer thought the combination of text-based and graphical-based user interfaces perform better than just the text-based one though he also suggested further improvements to include voice recording, and others. Overall, the results of the analysis of Case study 1 for Phase 1 and Phase 2 can be summarised and tabulated as shown in Table 5.14.

Table 5.15 Summary of analysis for Case Study 1 (Phase 1)

Case Study 1	**Aims**		**Results and Findings**
Phase 1	i)	To validate that the descriptive design framework (via the prototype software tool) is able to assist and support a designer that uses his own design approach and does not use any established design methodologies	The descriptive design framework managed to assist support a designer that did not use any established design methodology in this case study to design a device for the construction industry. The designer assisted in enabling him to delay his design decisions when the information needed is not available, linking his design decisions with the relevant documentations, tracing, tracking, saving, editing, and removing of design decisions, and many others.
	ii)	To ascertain how well the descriptive design framework can capture the thoughts and ideas of a designer throughout the design process	The descriptive design framework successfully captured the thoughts and ideas of a designer in this case study from start to end. The analysis of the data captured indicated the descriptive design tool captured the design decisions, date, time and their links with other design decisions.
	iii)	To find out whether the framework is able to help the designer to decide on delay making design decision when the design information is insufficient or unavailable.	Based on data captured by the descriptive design, the designer delayed his decision to confirm the shape of the device that he was designing as he needed to consult a construction expert on this matter and the consultant was only available on 13 December 2006. He later finalised the shape after meeting with the expert.
	iv)	To show that the descriptive design framework via the prototype software tool is allowing the designer to edit, save and add design decisions throughout the design process.	The design project in case study lasted several months and within this period, the designer has edited, saved, retrieved and added design decisions many times throughout the span of the project.
	v)	To find out whether the descriptive design framework (via the prototype software tool) can perform searching of design data.	The search function in this descriptive design tool only performed a very basic search and the searching process was merely a database search.

Table 5.16 Summary of analysis for Case Study 1 (Phase 2)

Case Study 1	**Aims**		**Results and Findings**
Phase 2	i)	To attest the capability of the framework to retrieve past design data and to allow further addition of the design data to support late design modifications occasionally required in most design projects	In Phase 1, the retrieval and addition of data throughout the design process was shown to be possible. However, in this case study, the retrieval and addition of data was done after the design project has completed and this aim is to re-confirm that the data stored by the descriptive design tool is retrieval and can be modify when necessary.
	ii)	To determine whether the descriptive design framework is able to track and trace design decisions made throughout the design process	This aim is placed in this Phase 2 of the case study because of the completion of the graphical user interface. The graphical user interface provided the designer a visualisation perspective of the design process as well as its direction and progress. This allowed the designer to track and trace his design decisions.
	iii)	To demonstrate how the descriptive design framework is able to support the computer-aided engineering (CAE) analysis process and how it manages the CAE analysis report	The feature to support CAE is similar to those of CAD and documentations. The descriptive design framework enabled the designer to link CAE documentations to his design decisions and this enabled the designer to make better design decisions.
	iv)	To show and verify the advantages of a flexible input interface that includes the text-based user interface and graphical-based user interface.	In order to verify the advantages of a flexible input is not easy task. In this case study, the advantages of the input interface was verified via an interview with the designers who had used the tool for several months.

5.8 Summary

The descriptive product design methodology proposed in this thesis is able to give support to the designer without influencing their design approach. The methodology is shown to be able to accommodate and support an experienced designer via a case study in which the designer did not use any systematic methodology stated in the literature in designing. The proposed methodology also provides support that is needed by designers due to the limitations of human cognition ability, such as limited memory and human errors during the design process with design methodology-related support facilities and computational-platform-related support facilities. This descriptive design framework employs a combination of text-based and graphical user-based interfaces to ease the process of capturing the designer's ideas and thoughts. The graphical user interface of this descriptive product design tool provides an overview of all the designer's design decisions to solve a particular design problem as well the progress direction of the design process. The proposed tool also provides a comprehensive and organised visualisation to enable the designer to reflect on their past and present as well as simulate their future design decisions.

Finally, like all existing methodology, the proposed descriptive product design methodology did not assist the designer in finding an effective solution to solve a design problem. The ability to solve design problems depends highly on the creativity, knowledge and experience of the designer.

Chapter 6

Case Study 2: Designing A Conceptual End-Effector to Manipulate and Handle Human Segments for a First Aid Robot System (FAROS)

The second case study was to verify the further advantages and capabilities of the proposed descriptive design methodology from the perspective of supporting designers with an additional tool. This is an ideation tool derived based on TRIZ. The case study involved the designing of a conceptual solution for a complex robotic end- effector to manipulate and handle human segments for a first aid robot (FAROS). The case study was carried out by a student who is a novice designer and the design project was carried out over a period of more than a year. The design project involved a lot of design considerations dealing with complex manipulation, lifting and positioning of human segments such as legs, hands and head. However, this design project was only to derive the conceptual design for FAROS. In this second case study, the novice designer could call upon additional design support facilities such as TRIZ (a concept ideation support method) to help them to solve any design problem if they have difficulties in solving them. In addition to that, as the designer is a novice, the descriptive design methodology framework is also able to help a novice designer to decompose the design requirements into sub-requirements. With this case study, the implementation of the descriptive design methodology in supporting an optional design support method such as TRIZ-based ideation tool was demonstrated.

6.1 Overview of Case Study

The case study was focused on the conceptual design of an end-effector to handle an unconscious person and it is a part of a bigger project to design a First Aid

Robot System (FAROS). In the application of first aid, the position of the limbs of an unconscious patient can vary greatly depending on the event that causes the patient to become unconscious. Hence, a first aid robot system (FAROS) needs to have end-effectors that can manipulate, handle and position the body, the hands and the legs to the recovery position recommended by the first aid manual. The task of manipulating the body, hands and the legs of the patient is made more difficult by the critical requirement of avoiding injuring the patient during the manipulation, handling and positioning of the different human segments. With the effects of clothing and the differences in rigidity and shapes of different human segments and where the trunk of the body is less rigid but bigger than the hands, the manipulation task becomes very complex.

This case study was carried out to demonstrate the ability of the descriptive design methodology framework to integrate and work together with other design tools to support the designer. There are several key differences between Case Study 1 and Case Study 2. In case study 1, the descriptive design framework was validated to establish that it is able to provide design methodology related-support facilities and the computation-platform-related support facilities. It was shown that it is able to link with external tools such as CAD, Microsoft Word (documentations) and CAE tools. Such links are useful in assisting the designer in managing their data and the descriptive design methodology throughout the design process does not influence the approach and the decisions of the designer. However, there will be circumstances, especially for novice designers, when it is difficult to find the solutions for certain design problems. An ideation tool based on TRIZ would be helpful in such circumstances. Any usage of an ideation tool which recommends design solutions are a in a way influencing the decisions of a designer.

Therefore, an ideation tool should be optional and should only be considered if the designer wishes to use it. Therefore, the descriptive design is derived with the flexibility to prevent its framework from influencing or interfering with the designer’s approach or style and yet is later able to support design tools and methods that influence the designer’s decision when they wish to do so.

Case study 2 is specifically aimed at showing that the descriptive design framework is capable of supporting such an optional tool.

In addition, unlike the previous links with CAD, Microsoft Word and CAE files, the optional ideation tool was derived to work with the descriptive design prototype tool in a more integrated manner. This was because the ideation tool based on TRIZ was called upon by the descriptive design prototype tool and then the data derived by the ideation tool was stored in the descriptive design repository. Figure 6.1 illustrates the differences between the linkage of the descriptive design tool with the CAD, CAE and Microsoft Word systems when compared to the link with the ideation tool based on TRIZ. The advantage of adapting the descriptive design repository to store the data from an ideation tool is that the improving and worsening features, as well as the inventive principles related to the specific design solutions can be accumulated. These data will be useful for future research to determine the correlations with specific design solutions. Such correlations can be applied to assist the reuse of design solutions to solve new design problems for future work.

Finally, this case study was carried out based on the design approach adopted is a step-based model with design phases. Hence, this case study enabled the investigation on whether the descriptive design methodology can accommodate a step-based model design approach and how well it accommodates it.

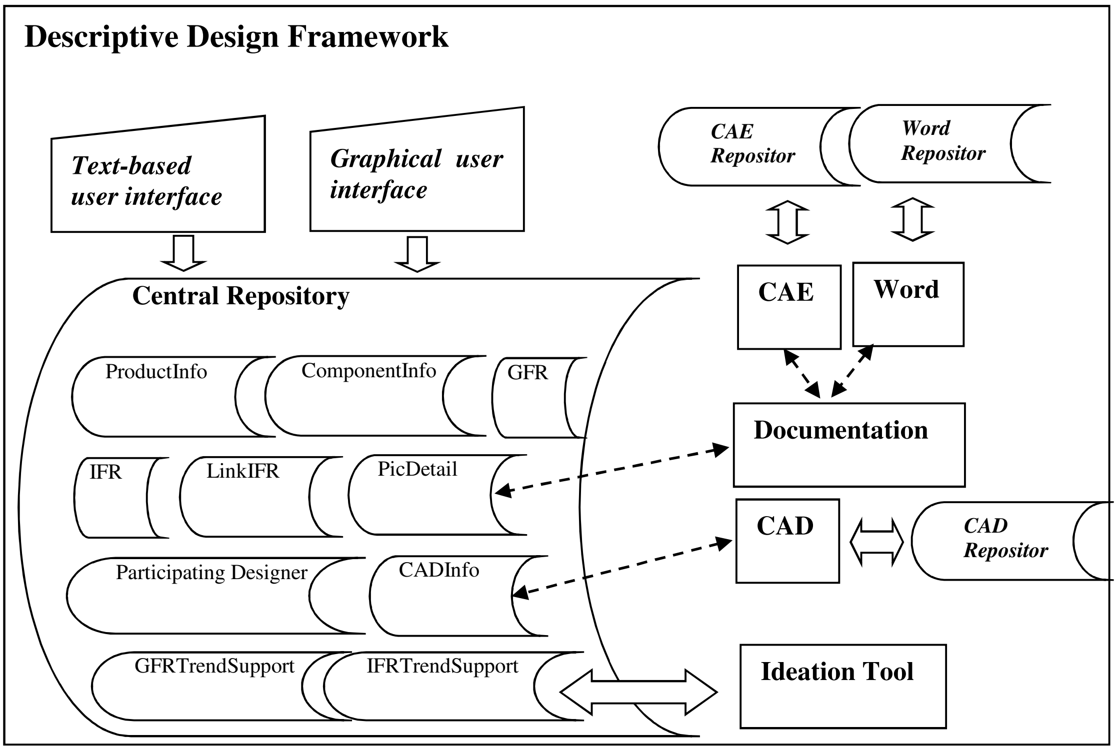


Figure 6.1 Difference between the linkages of ideation tool with descriptive design framework

6.2 Implementation of the Descriptive Design Methodology with TRIZ-based Ideation Tool in Case Study 2

This case study started with the intention of allowing a novice designer to design a robot end-effector that will be part of the FAROS project without using the descriptive design tools initially. The descriptive design methodology tool was later introduced to the novice designer to assist him in decomposing the design requirements into sub-requirements and solution ideas as the designer faces difficulty in deriving solution concepts for the FAROS end-effector. The descriptive design methodology is a methodology that captures the functional requirements, the sub-requirements, the constraints, the criteria as well as the solution ideas from a designer in the process of designing. This means it is critical that the designer has in-depth knowledge, sufficient experience and is able to come up with ideas to solve a design problem in a design task. This ability is usually lacking in a novice designer.

The descriptive design methodology is only able to assist the designer in decomposing design tasks from functional requirements and to support any other design tools via a Windows application protocol interface (API). In addition, it also provides visualisation of the design process to the designer and this helps the designer to see the directions of their progress in design. However, the descriptive design methodology is not able to provide solution concepts for designers. Hence, the descriptive design methodology tool provides a platform to integrate other design tools as well as other design methods or approach to enable better design support for the designers.

6.2.1 The aims of Case Study 2

In this case study, the implementation of the descriptive design prototype tool was first carried out to help the designer to come up with solution concepts for FAROS. When the designer faced difficulty in deriving solution concepts, an optional tool was derived to help him. In order to assist the designer to come up with solution ideas, an ideation tool adapted based on TRIZ was derived to work on top of the descriptive design methodology tool platform in this case study. The objectives of Case Study 2 were to:

i) reaffirm the advantages of and support that can be provided by the descriptive design methodology framework verified in Case Study 1

ii) assess the effectiveness of the descriptive design methodology in supporting a novice designer.

iii) evaluate the feasibility of the descriptive design methodology in supporting an ideation tool based on TRIZ.

iv) find out whether the descriptive design methodology is able to support a step-oriented based design approach (an approach that has design phases).

v) determine the effectiveness of the TRIZ-based ideation tool in supporting the designer to solve design problems.

The next section will briefly describe the ideation tool derived based on TRIZ and its role in supporting designers before further elaboration on the implementation details of the case study in section 6.3 and how a designer utilises the descriptive design tool along with the TRIZ to solve the design problem for a FAROS end-effector.

6.2.2 The Ideation Tool based on TRIZ

One the main tools of TRIZ is the contradiction matrix which is widely used by multi-national companies in deriving new products for the current competitive global market. Though the TRIZ contradiction matrix is able to assist the designer to derive new solution concepts, it is up to the decision of the designer whether they require such a tool to assist them in design. Therefore, this ideation tool based on TRIZ is an optional tool. In this case study, the designer was allowed a period of several weeks to attempt to derive a solution concept for an end-effector to be used in FAROS on his own before enabling the option of using the ideation tool based on TRIZ. The architecture of this ideation is based on TRIZ and the designer needs to specify a list of improving features and worsening features related "Introduced Functional Requirement" (IFR). This ideation tool allows the designer to provide weights to each of the worsening and improving features to enable a better search for the inventive principles to solve design problems. The weights are grouped into four, namely, 0.25, 0.5, 0.75, and 1.0 where 1.0 has the highest priority in improving the features related to the specified IFR. Figure 6.2 illustrates the user interface of the ideation tool based on TRIZ. The usage of weights in the tool is completely different from the ultimate aim of TRIZ concept to achieve an ideal solution. Though the use of weights (which means trade-off is allowed) is not in agreement with the TRIZ concept of an ideal solution, the ideation tool derived also allows the designer to disable the weightage if they prefer. By disabling the weightage, the tool would be merely a common TRIZ tool. The consideration for the use of weightage is based on the fact that the chances of finding an ideal solution may not be high in reality in many circumstances.

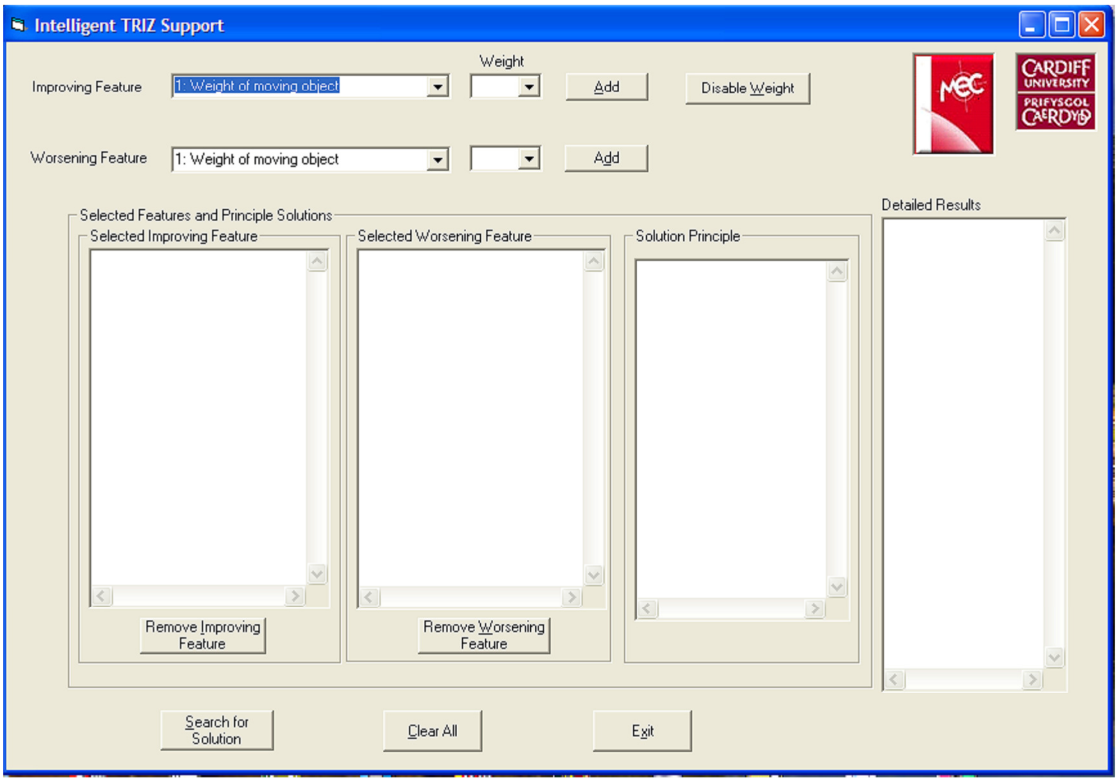


Figure 6.2 The user interface of ideation tool based on TRIZ that allows assignment of weightage

The ideation tool based on TRIZ allows the designer to choose whether they would search for a TRIZ inventive principles solution with or without weights. The improving and worsening features provided in this ideation tool are dependent on the contradiction matrix table that is used. In this case study, the updated TRIZ contradiction matrix in 2003 by Mann (2003) was used instead of the newest updated contradiction matrix of 2009 (Mann 2009). The 2003 version matrix has 48 improving and worsening features while the 2009 matrix has 50 improving and worsening features. The updated contradiction matrix of 2003 and 2009 were preferred because both the updated versions of the TRIZ contradiction matrix have a complete recommendation of inventive principles for all contradicting features except for the diagonal cells of the table (until now the TRIZ contradiction matrix has no inventive principle recommendations for the same improving feature and worsening feature).

The ideation tool based on TRIZ was developed independently from the contradiction matrix. This means that the tool itself is a program that reads from an Excel file. The contradiction matrix is an Excel file with the first row cells of improving features and first column cells of worsening features. The corresponding cells of the rows and the columns were inventive principles to solve design problems. A program was developed to read the Excel file and upon the selection of the improving features and worsening features, the corresponding inventive principles would be recommended by the matrix to solve a particular design problem. However, a designer can choose more than one improving features as well as worsening features. Each corresponding individual improving feature with an individual worsening feature will lead to the recommendation of a few inventive principles. Hence, if several improving features and worsening features are chosen, then the number of recommended inventive principles can be many. The recommended inventive principles are merely general solutions and the designer needs to translate these general solutions into specific solutions. Attempting to find specific solutions from a long list of inventive principles is a very challenging task. Therefore, it is best to rank these inventive principles to enable the designer to consider first the inventive principle that has a higher chance of solving the design problem. The assignment of weights applied in this ideation tool not only allows consideration on trade-offs but also allows the ranking of the inventive principles obtained. Table 6.1 illustrates the ranking process applied to the ideation tool based on TRIZ.

In Table 6.1, an example of how the inventive principles are determined and ranked from the requirements of a design problem to improve the shape and the adaptability/versatility of a design to the worsening feature for force or torque and loss of substance (two improving features and two worsening features) is shown.

The inventive principles proposed by the technical contradiction matrix are 14, 17, 35, 9, 2, 29, 35, 30, 3, 5, 35, 15, 17, 14, 13, 10, 3, and 15. It is noted that inventive principle 35 is recommended three times while inventive principle 3, 14, 15 and 17 are repeated twice. With the weight assignment, inventive principle 35 has a weight of (0.75*0.75) + (0.75*0.5) + (0.5*0.75) = 1.3125. However, the ranking method used will consider the highest repetition first prior to the weights. Without weights, 3, 14, 15, 17 would be equally ranked but with the weights, 14 (0.9375) and 17 (0.9375) are ranked higher than 3 (0.625) and 15 (0.625).

Table 6.1 An example of how TRIZ contradiction matrix with trade-offs is used

Weight			0.75		0.5
	Worsening / Improving	....	15: Force/ Torque	....	25: Loss of Substance
	⋮	⋮	⋮	⋮	⋮
0.75	9: Shape	....	14 17 35 9 2	....	29 35 30 3 5
	⋮	⋮	⋮	⋮	⋮
0.5	32: Adaptability /versatility	....	35 15 17 14	....	13 10 3 15

For this case study, at the time of implementation, only the updated contradiction matrix 2003 (2003) was available. The case study was carried out based on the adapted ideation tool based on TRIZ that utilised the 2003 contradiction matrix. The designer has applied the TRIZ-based ideation tool without using weightage on the features in this case study.

6.2.3 The Integration between the Descriptive Design Methodology Tool and the TRIZ-based Ideation Tool based on TRIZ

In the implementation of the descriptive design tool with the TRIZ-based ideation tool, both tools are integrated via centralised database system but using different user interfaces. The interface for the TRIZ-based ideation tool is shown in Figure 6.2. The design of the entire database system was based on Microsoft Access and the system is illustrated in Figure 6.3, which is similar to Figure 4.11 except for the addition of the TRIZ ideation database system. Note the demarcation line between the descriptive design tool and the TRIZ-ideation tool.

The descriptive design tool was integrated with the TRIZ-ideation tool and the entire system was derived using Microsoft Visual Basic 6.0 and utilising Microsoft Access database as the repository tool. The additional database tables created were used for storing the information generated by the TRIZ-based ideation tool. The integration of the descriptive design tool and the TRIZ-based ideation is designed in such a way that the designer can choose to use the ideation tool when they prefer or wishes to. If the designers do not wish to use the ideation tool, there will be no disruption to the designer in carrying out their design activities.

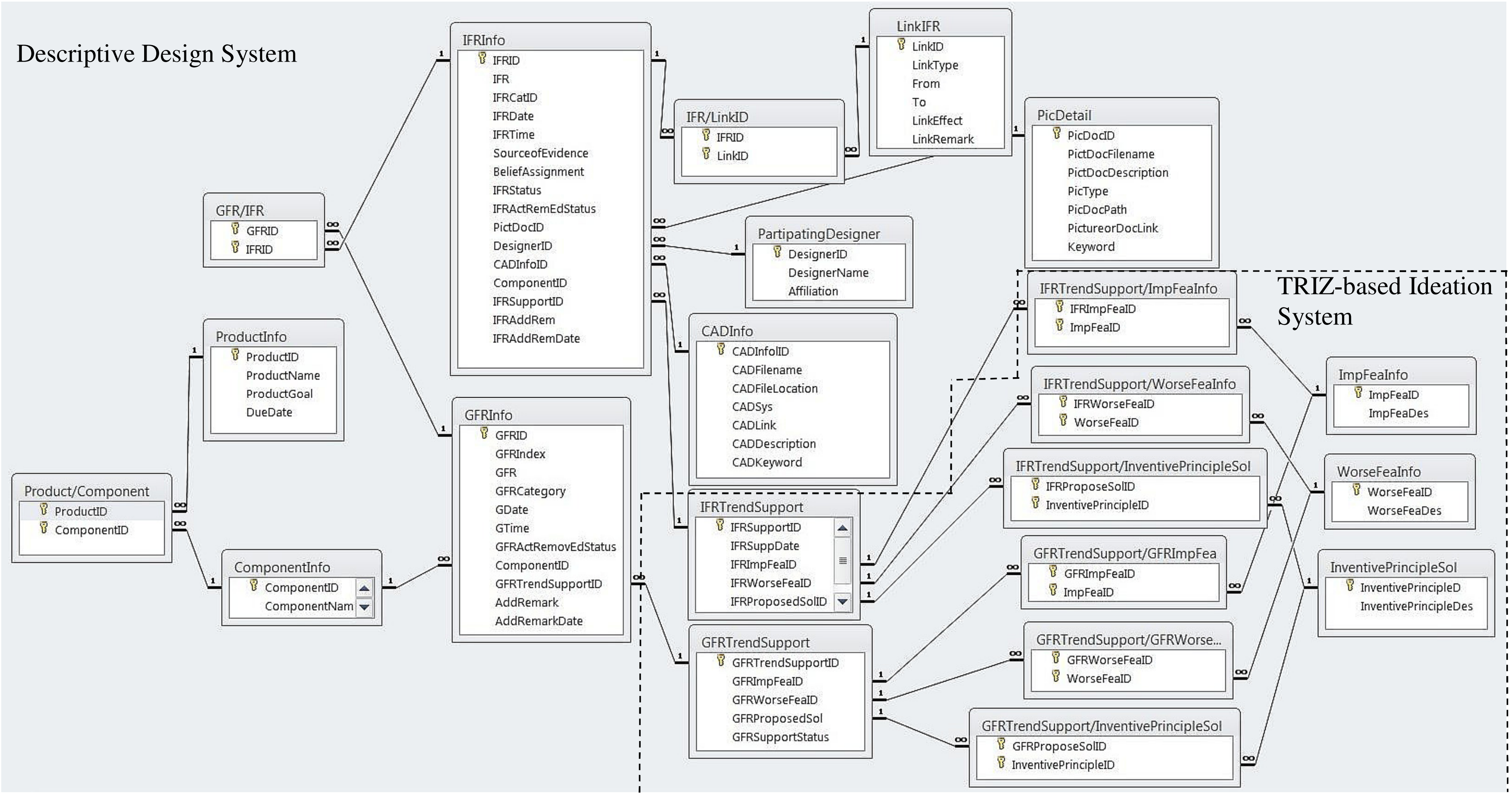


Figure 6.3 The demarcation of descriptive design system and the TRIZ-based ideation system within the entire database system

6.2.4 The details of implementation for Case Study 2

During this initial stage of the design project, the designer has tried to apply an established design methodology introduced by Pahl and Beitz (1995) to solve design problems. After observing the novice designer trying to design the conceptual end-effector for several months without success, the descriptive design tool was introduced to the designer. The designer was given a briefing and basic instructions on how to use the descriptive design prototype software tool. The descriptive design prototype tool was utilised for several months and the designer managed to decompose the main design requirements into sub-requirements, namely GFRs to IFRs. The designer continued applying the established methodology of Pahl and Beitz whilst using the descriptive design tool.

Even though the novel designer used the descriptive design prototype tool to design the conceptual end-effector, the novice designer had faced difficulties in deriving solution concepts even after decomposing the GFRs into IFRs. It is common for novice designers to have difficulties in solving design problems. An ideation concept such as TRIZ can be useful in helping a designer to come up with ideas to solve design problems, especially a novice designer. The ideation was derived to work with the descriptive design framework to provide an optional design tool for the designer to enable additional support for him to solve design problems.

Case Study 2 started with the novice designer using the descriptive design prototype tool after the design project had started a few months earlier. This was later followed by the introduction of the ideation tool derived based on TRIZ to further help the designer to come up with solution concepts. The designer was observed to use the descriptive design prototype tool to decompose GFRs into IFRs. In the process of decomposing the GFRs into IFRs, the designer managed to identify some of the further investigations needed to design the end-effector. Some of these investigations included simple experiments and determining shapes, and some measurements and weights of human segments.

As the design of an end-effector for FAROS only progressed up to the conceptual design phase, the outcome of the design project does not have any detailed model of the end-effector. Since the designer adopted a design methodology proposed by Pahl and Beitz (1995), the design approach he used was in phases, namely a step-oriented model. The designer in this project only needed to come up with the conceptual design for the end-effector for FAROS and this case study was conducted until the completion of the conceptual phase.

Finally, the case study concluded with an interview with the designer about the descriptive design framework and the TRIZ-based ideation tool. The analysis of the results obtained from the case study was based on the data captured throughout the case study and on the outcome of the interview.

In Case study 2, the designer started initially with six main design requirements or Given Functional Requirements (GFRs) as listed below:

1. Need to grasp and manipulate human segments
2. Need to support human segments
3. Need to take max load human segments
4. Need to be durable
5. Need to be cost-effective
6. Need to be easily implemented and maintained

Since the design project only proceeded until the conceptual phase, the last two GFRs were abandoned. Figure 6.4 shows a snap shot of the GFR text-based user interface with the GFRs data that the designer keyed in. The text-based user interface was slightly modified to include a button to run the TRIZ-based ideation tool so that both tools worked in an integrated manner. Note the update user interface when compared to the user interface in Case Study 1 (refer to Figure 4.5 and 4.6 of the previous chapter). The additional "TRIZ" button in the GFR user interface (refer to Figure 6.4) and "Call Ideation Tool" button in the IFR user interface as well as the "Link CAD/JPG" button to replace to the "Add DOC" button (see Figure 6.5).

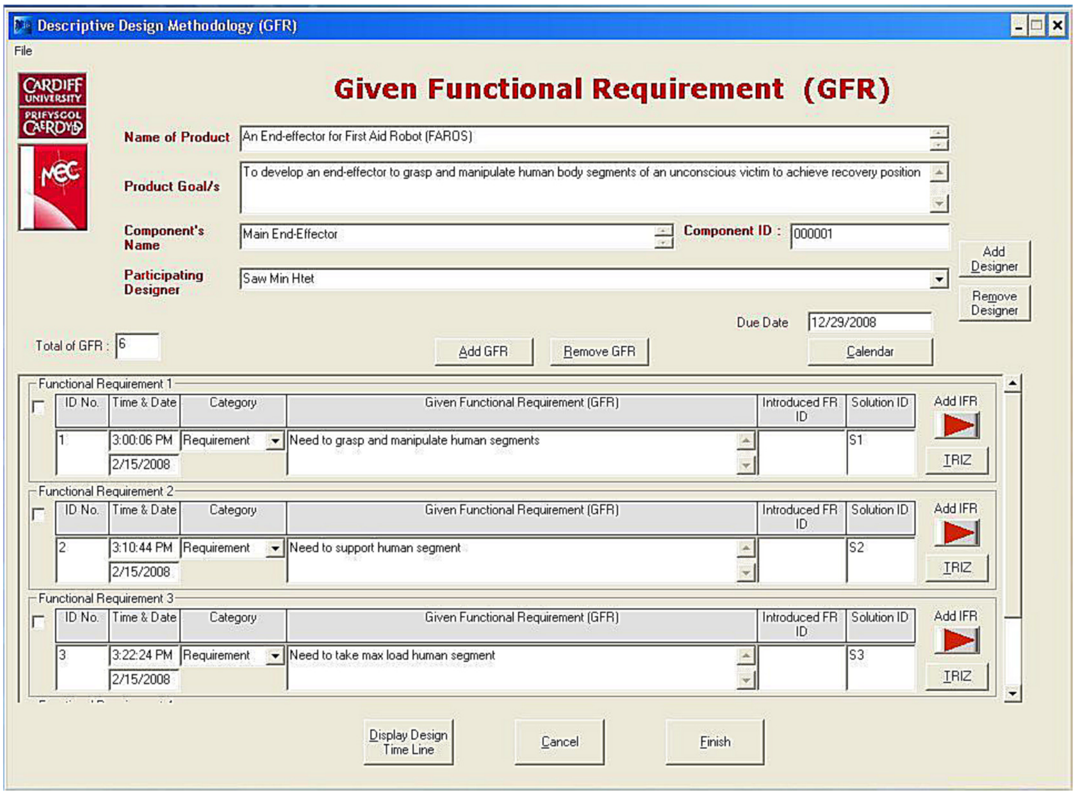


Figure 6.4 A snapshot of the modified text-based user interface of GFR for the descriptive design prototype tool that was integrated with the TRIZ-based ideation tool

In the application of TRIZ, the improving features and the worsening features should be used to represent a design problem scenario and the proposed inventive principles would be general solutions for the design problem. However, the TRIZ contradiction matrix would have been more effectively used if the contradicting matrix to solve the design problem at the root level (Mann 2002) had also been used. Though the application of contradiction can also be used to solve design problem at the top level, the number of contradiction features may be large and can lead to a high number of recommended inventive principles. A lot of recommended inventive principles may cause difficulties to a designer when deciding on which is the appropriate principle to be used. Nevertheless, applying the TRIZ contradiction matrix is still possible and may provide a novice designer with some ideas on the general solutions even though it is applied at the top level. Due to these circumstances, every GFR as well as IFR may need solving to derive solution concept. Hence, the user interface of the descriptive design tool has a button for every GFR and IFR to run the ideation tool for the designer if they wish to seek its assistance to solve a specific GFR or IFR. Figure 6.5 shows a snapshot of the modified text-based user interface of IFR for the descriptive design tool used by the designer in this case study. The snapshot was done while the designer was providing information about IFRs for the GFR1.

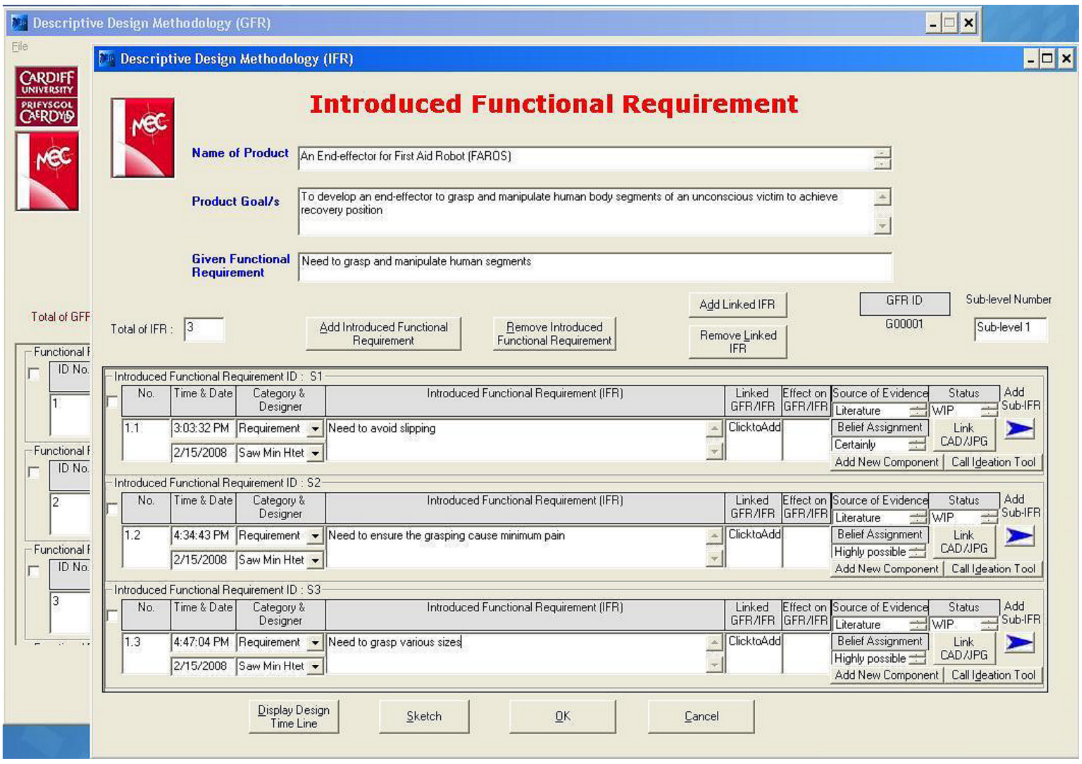


Figure 6.5 A snapshot of the modified text-based user interface of IFR for the descriptive design prototype tool that was integrated with the TRIZ-based ideation tool

The TRIZ-based ideation tool can be launched in the GFR text-based user interface by pressing the button "TRIZ" at the end of GFR row as shown in Figure 6.4. While in the IFR text-based user interface, the ideation tool can be launched by pressing the button "Call Ideation Tool" at the end of each IFR row as shown in Figure 6.5.

Upon launching the TRIZ-based ideation tool, the user interface shown in Figure 6.2 will emerge for the designer to provide inputs on the improving and worsening features as well as to decide whether to use the weightage or not. More than one improving and worsening feature can be selected from the list provided. Figure 6.6 illustrates the TRIZ-based ideation tool user interface after the designer launches the tool to find the general solution to IFR 1.7, which is to avoid injuring the unconscious victim. In selecting the improving feature for IFR 1.7, the nearest improving feature that describes avoiding injuring the unconscious victim is feature 31, which is "Other harmful effects generated by system". When improving that feature, the possible worsening features were feature 45: "Device Complexity", feature 5: "Area of moving object" and feature 46: "Control Complexity". In addition, the designer also chose not to enable the use of weightage. Figure 6.7 illustrates the inputs from the designer to find the general solution to IFR 3.1 (Need to determine the shape of end-effector to take max load) where the designer interpreted this IFR as an intention to improve the weight of moving object that can be grasped by the end-effector. For this improvement, three worsening features were chosen and there were similar to the ones for the IFR 1.7, namely features 45, 5 and 46. Lastly for the IFR 1.6.1.1.2, the improving feature chosen was feature 32: adaptability/versatility against two worsening features, namely feature 45 and 46 respectively. Figure 6.8 shows a snapshot of the user interface for the TRIZ-based ideation tool as well as the input data to solve IFR 1.6.1.1.2.

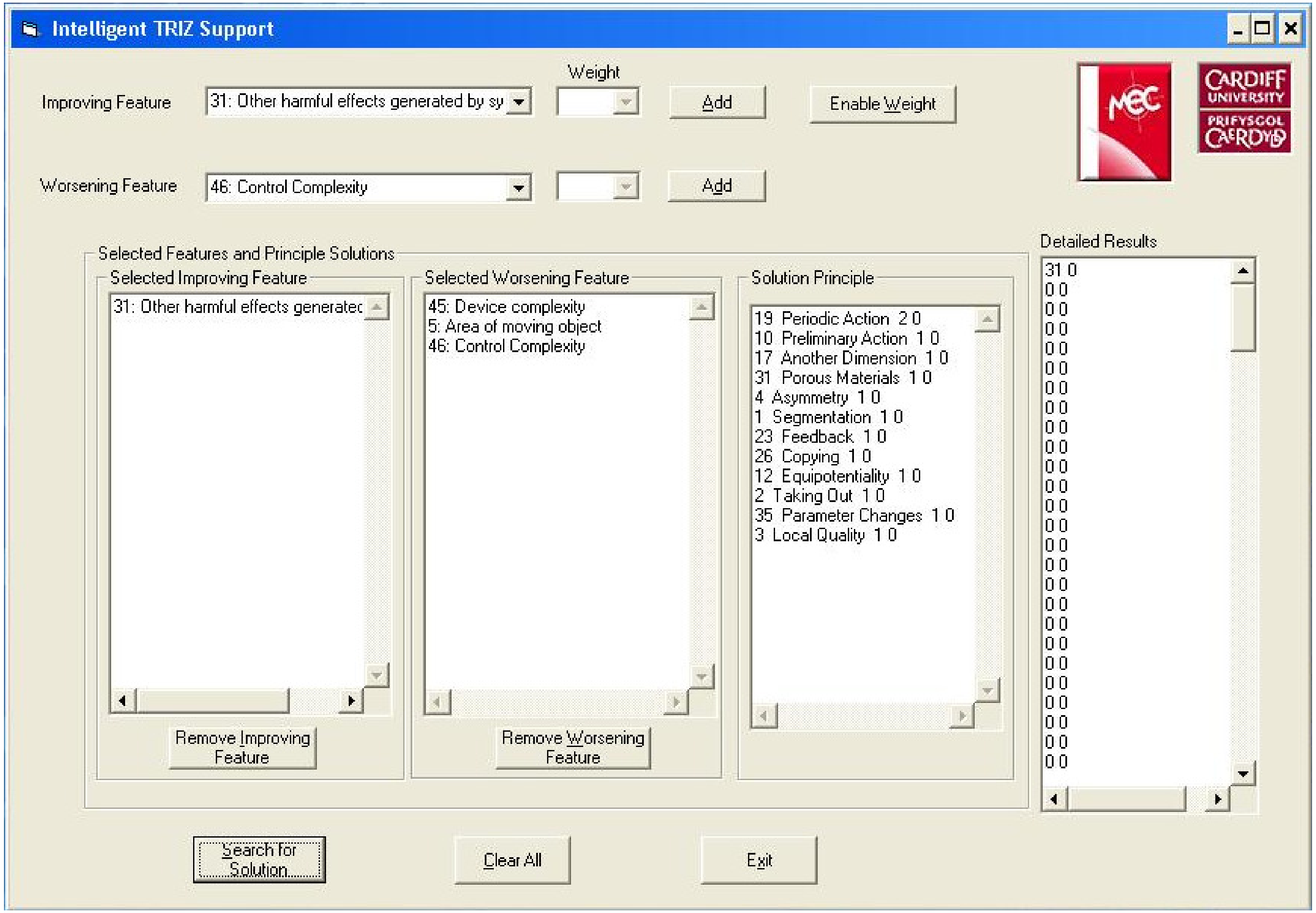


Figure 6.6 A snapshot of the TRIZ-based ideation tool with the input data completed and the results shown for the IFR 1.7

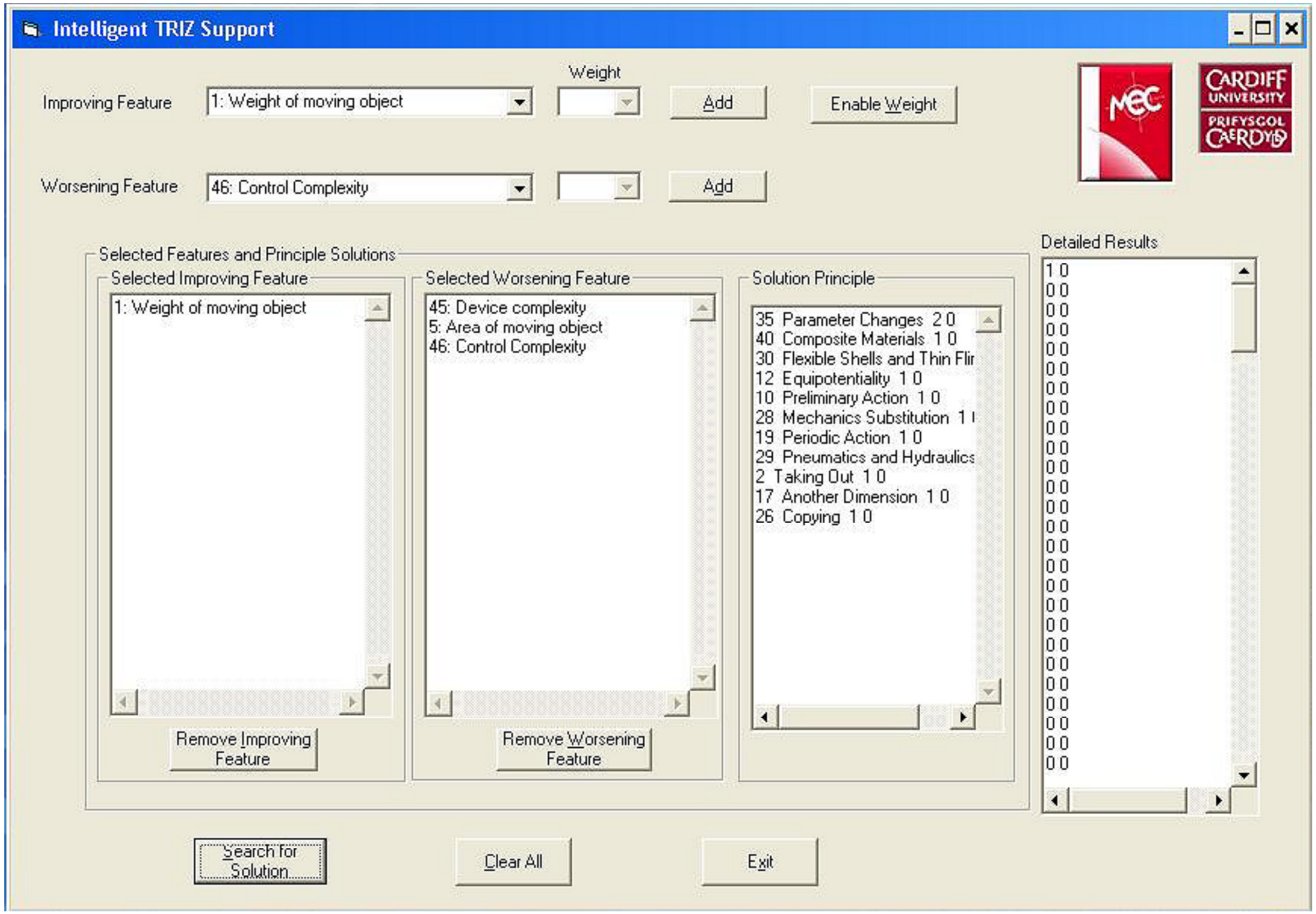


Figure 6.7 A snapshot of the TRIZ-based ideation tool with the input data completed and the results shown for the IFR 3.1

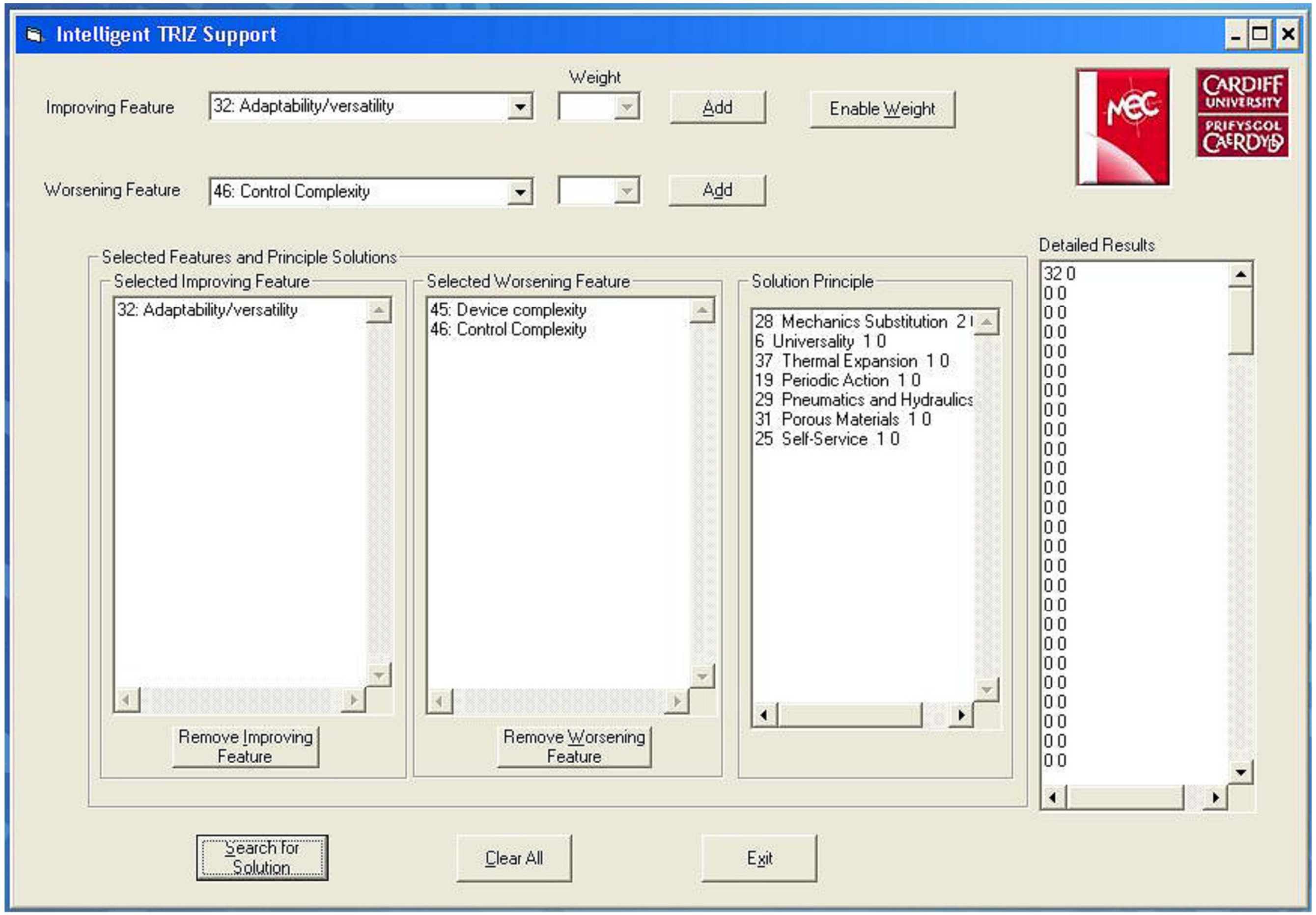


Figure 6.8 A snapshot of the TRIZ-based ideation tool with the input data completed and the results shown for the IFR 1.6.1.1.2

6.3 Results from Implementation of the Descriptive Design Methodology Software Tool with TRIZ-based Ideation Tool in Case Study 2

The results of this case study would initially looked at the data captured by the descriptive design prototype tool and later into the results generated by the TRIZ-based ideation tool stored in the centralised database system.

The thoughts and ideas of the designer were captured by the descriptive design prototype tool and the data captured are stored in the relevant repositories with several database tables similar to those shown in the previous diagram. However, due to the inclusion of data from the TRIZ-based ideation tool, the repository system design had to be changed. The changes involved the additional several tables that linked the GFRInfo table and the IFRInfo table. This is because, as mentioned in the previous section, a designer can use TRIZ at the top level of the design process i.e. during GFR. However, it is better to use TRIZ at the lower level i.e. IFR or sub-IFRs stage because the contradiction at lower level is more precisely defined and usually has lower number of contradictions. In the development of the descriptive design prototype tool with the TRIZ-based ideation tool, the tool was designed to include the capacity to capture both the results of the ideation tool at the GFR level as well as at the IFR level respectively as seen in Figure 6.3.

6.3.1 The database information captured in Case Study 2

The results of this case study will initially look at the data captured by the descriptive design prototype tool and later into the results generated by the TRIZ-based ideation tool stored in the centralised database system. Similar to the previous case study, other than observations of how the descriptive design tool helps the designer in design, looking into the data captured in this case study is also a key factor in determining the contributions of the tool in assisting the designer. The initial data captured by the descriptive design tool are the data related to the product that the designer is trying design. These data are stored in the ProductInfo table as shown in Table 6.2. Table 6.2 shows the data captured by the descriptive design tool and stored in database system which is based on Microsoft Access.

Table 6.2 A screen shot of the data captured in the ProductInfo table for the design of FAROS end-effector

Product ID	Product Name	ProductGoal	DueDate
000001	FAROS End Effector	Grasp, Handle and Manipulate human segments of unconscious victim	10/7/2008

ComponentID	Click to Add
C00001	
*	

As shown in Table 6.2, information about the name of the product to be designed, the goal of this product and the due date in which the design of this product should be completed were provided by the designer via the text-based user interface and stored in the ProductInfo table. Due to the relationship created between tables, as shown in Figure 4.11 in Chapter 4, there are links between the tables to enable an appropriate representation of the data captured. From Table 6.2, it can be seen the link between ProductID 000001 (FAROS End-Effector) and the ComponentID C00001 (Main End-effector).

In Table 6.3, the ComponentInfo table stored the data about the component name and the possible cost of the product. However, the cost of the initial end-effector for this case study was not estimated by the designer.

Table 6.3 A screen shot of the data captured in the ComponentInfo table for the design of FAROS end-effector

ComponentI	ComponentName	Component
C00001	Main End-Effector	£0.00
*		£0.00

The initial design requirements from the design specifications have six GFRs as shown in Table 6.4. As the designer was only responsible for deriving the conceptual design of the end-effector without progressing into the detail design phase, the designer decided to abandon the last two GFRs. These were given "Removed" status due to that decision and the designer only focused on deriving a conceptual end-effector based on the top four GFRs.

Table 6.4 A screen shot of the data captured in the GFRInfo table for the design of FAROS end-effector

GFRID	GFRInd	GFR	GFRCategory	Start Date	Start Time	GFRActF	Compor	GFRTrei	AddRer	AddRemarkD
G00001	1.0	Need to grasp human segment	Requirement	15/2/2008	3:00 PM	Active	C00001			
G00002	2.0	Need to support human segment	Requirement	15/2/2008	3:10 PM	Active	C00001			
G00003	3.0	Need to take max load human segment	Requirement	15/2/2008	3:22 PM	Active	C00001			
G00004	4.0	Need to be durable	Requirement	7/3/2008	2:40 PM	Active	C00001			
G00005	5.0	Need to be cost-effective	Requirement	10/3/2008	9:45 AM	Removed	C00001			
G00006	6.0	Need to be easily implemented and	Requirement	18/3/2008	11:23 AM	Removed	C00001			

The GFRInfo table has a many-to-many relationship link with the IFRInfo Table. This means a GFR can have many IFRs (sub-requirements, ideas, etc.) which will ultimately meet or satisfy the GFR and these IFRs can also be used for other GFRs if similar IFRs can be used to satisfy other GFRs. The many-to-many relationship links can be observed if the GFRs are expanded as shown in Table 6.5. However, not all GFRs have corresponding IFRs. For example, GFR 5.0 has no corresponding IFR in IFRInfo table as GFR 5.0 has no sub-requirements, ideas, etc. (IFRs) as GFR5.0 was abandoned. In Table 6.5, the GFR 1.0 (GFRID G00001) was not expanded for view in this write-up as the list of linked IFRs for GFR1.0 would be very long.

Table 6.5 A screen shot of the data captured in the GFRInfo table for the design of FAROS end-effector with the links to IFRinfo expanded

GFRID	GFRInc	GFR	GFRCategory	Start Date	Start Time	GFRActRen	Compor	GFRTren	AddRemark	AddRem
G00001	1.0	Need to grasp human segment	Requirement	15/2/2008	3:00 PM	Active	C00001			
G00002	2.0	Need to support human segment	Requirement	15/2/2008	3:10 PM	Active	C00001			

IFRID	Click to Add
2.1	
2.1.1	
2.1.1.1	
2.2	
2.2.1	
*	

GFRID	GFRInc	GFR	GFRCategory	Start Date	Start Time	GFRActRen	Compor	GFRTren	AddRemark	AddRem
G00003	3.0	Need to take max load human segment	Requirement	15/2/2008	3:22 PM	Active	C00001			
G00004	4.0	Need to be durable	Requirement	7/3/2008	2:40 PM	Active	C00001			
G00005	5.0	Need to be cost-effective	Requirement	10/3/2008	9:45 AM	Removed	C00001			

IFRID	Click to Add
5.1	
5.1.1	
5.1.1.1	
*	

GFRID	GFRInc	GFR	GFRCategory	Start Date	Start Time	GFRActRen	Compor	GFRTren	AddRemark	AddRem
G00006	6.0	Need to be easily implemented and maintained	Requirement	18/3/2008	11:23 AM	Removed	C00001			

The information on the IFRs was captured and stored in the IFRInfo table as shown in Table 6.6. As observed in Table 6.6, the snapshot of the information stored in IFRInfo table was done at the end of Case Study 2 as the "Status" field indicated "Final". Based on the information stored in the IFRInfo table, it is obvious that a number of documents were linked to several IFRs. The descriptive design methodology has assisted the novice designer to decompose the main design requirements or GFRs into IFRs and some of the IFRs indicated a need to determine certain parameters such as mass, dimensions, shapes and others prior to solving the related design problems. Hence, the designers have to perform calculations, experiments and even a literature review to determine these parameters and the documents containing these calculations, experiments and other studies that have been performed. The descriptive design methodology prototype tool stored the information in the PicDetail table as shown in Table 6.7. In this case study, all of the items stored in PicDetail table (refer to Table 6.7) are documents. No pictures or sketches were stored by the designer. This is because the designer designed the end-effector from scratch and only focused on conceptual design.

Table 6.6 A screen shot of the IFRInfo table for the design of a conceptual end-effector in the Case Study 2

IFRID	IFR	IFRCatID	Start D	Start Ti	SourceofEvidence	Beli	Statu	IFRActRe	PicDoc	Design	CADI	Compor	IFRSupportI
1.1	Need to avoid slipping	Requirement	2/15/2008	3:03 PM	Literature	1.00	Final	Active		0001		C00001	
1.1.1	Need sufficient friction	Requirement	2/15/2008	3:42 PM	Knowledge	0.50	Final	Active		0001		C00001	
1.1.2	Need sufficient grasping force	Requirement	2/15/2008	4:04 PM	Literature	0.50	Final	Active		0001		C00001	
1.2	Need to ensure the grasping cause minimum pain	Requirement	2/15/2008	4:34 PM	Literature	0.75	Final	Active	0000001	0001		C00001	
1.3	Need to grasp various sizes	Requirement	2/15/2008	4:47 PM	Literature	1.00	Final	Active		0001		C00001	
1.3.1	Need to determine the average size of human segment for grasping	Requirement	2/15/2008	4:50 PM	Knowledge/Experience	0.75	Final	Active	0000002	0001		C00001	
1.4	Need to grasp various shapes	Requirement	2/15/2008	4:53 PM	Literature	1.00	Final	Active		0001		C00001	
1.5	Need to grasp segments of various hardness	Requirement	2/15/2008	4:58 PM	Colleague	0.50	Final	Active		0001		C00001	
1.4.1	Need to determine the various shapes of the human segments	Requirement	2/16/2008	10:05 AM	Knowledge/Experience	0.00	Final	Active	0000003	0001		C00001	
1.5.1	Need to determine the maximum pressure that can be applied on various body segments	Requirement	2/21/2008	9:20 AM	Literature	0.50	Final	Active	0000004	0001		C00001	
1.5.2	Need to determine the way to grasp segments of various hardness	Requirement	2/21/2008	9:40 AM	Literature	0.50	Final	Active	0000005	0001		C00001	
2.1	Need to determine max. support human segment	Requirement	2/21/2008	4:30 PM	Literature	0.50	Final	Active		0001		C00001	
2.1.1	Need to calculate various mass of body segments	Requirement	2/21/2008	4:45 PM	Literature	1.00	Final	Active	0000006	0001		C00001	
3.1	Need to determine the shape of end-effector to take max load	Requirement	2/22/2008	11:02 AM	Knowledge/Experience	0.75	Final	Active		0001		C00001	000002
1.5.2.1	An end-effector with magnetic gel	Requirement	3/3/2008	3:43 PM	Literature	0.50	Abando	Removed	0000007	0001		C00001	

Record: 1 of 42 No Filter Search

Table 6.7 A screen shot of the PicDetail table that stored the information about the documentation for the design of a conceptual end-effector in the Case Study 2

PicDocID	PictDocFilename	PictDocDescription	PicType	PicDocPath	PictureorDocLink	Keyword
0000001	Pressure for maximum pain threshold.doc	Pain Threshold and Tolerance	Document	c:\project\faros\	Microsoft Word Document	Pressure, Pain
0000002	Survey on average size of human segments.doc	Experimental survey on sizes of human segments	Document	c:\project\faros\	Microsoft Word Document	Human segment, size
0000003	Estimation of human body segments parameters.doc	Calculation of average shape, dimensions and weight for various human segments	Document	c:\project\faros\	Microsoft Word Document	Mass, Body Segments
0000004	Calculation on max. load per area.doc	Grasping load per area	Document	c:\project\faros\	Microsoft Word Document	Grasping, Max. Load
0000005	Grasping different hardness.doc	A study of grasping object of different hardness	Document	c:\project\faros\	Microsoft Word Document	Grasping, Hardness
0000006	Weighing experiment on various human segments.doc	To determine the mass of various human segments and find out the max. mass	Document	c:\project\faros\	Microsoft Word Document	Max. mass, load
0000007	Investigations and study on different type of end-effector.doc	To qualitatvely study on these end-effectors	Document	c:\project\faros\	Microsoft Word Document	end-effector type

Among the important data stored in Case Study 2 are the linked data which implied that a suggested IFR was able to solve one or more other IFRs or GFRs. The information was stored in the LinkIFR table and this table has a junction table known as IFR/LinkIFR. As mentioned earlier in section 5.3.1, the need for the junction table is to cater to a many-to-many relationship. The information stored in the LinkIFR table is shown in Figure 6.9.

IFR/LinkIFR table

IFRID	LinkID
1.3.1.1	000001
2.1.1.1	000002
1.5.1.1	000003
1.5.1.1	000004
1.5.3.1	000005
1.1.1.1	000006
1.7.1.1.1	000007

LinkIFR table

LinkID	From	LinkType	To	LinkEffect	LinkRemark
000001	1.3.1.1	IFRtoGFR	1.0	Remark	Constraint for end-effector size
000002	2.1.1.1	IFRtoGFR	3.0	Remark	Constraint for grasping load
000003	1.5.1.1	IFRtoIFR	1.2	Remark	Solve IFR 1.2
000004	1.5.1.1	IFRtoIFR	1.1.2	Remark	Restricted by IFR 1.5.1.1
000005	1.5.3.1	IFRtoIFR	1.1	Remark	Reference for IFR 1.1
000006	1.1.1.1	IFRtoIFR	1.1	+	
000007	1.7.1.1.1	IFRtoIFR	1.1.2	Remark	Control grasping force

Figure 6.9 A screen shot of the junction table IFR/LinkIFR (top) and the LinkIFR table (bottom) that stored the information about the links between IFR and IFR or GFR for the design of a conceptual end-effector in Case Study 2

There are 7 links created by the designer and one of the links was a positive one i.e. a link that enhanced or improved the corresponding IFR, in this case study, IFR 1.1.1.1 (Use silicone rubber as inner surface for the end-effector) helps to enhance IFR 1.1 (avoidance of slipping in grasping human segments).

In addition, there were two IFR to GFR constraint links, namely GFR 4.0 (Need to be durable) and GFR 3.0 (Need to take max load human segment) were constrained by GFR 1.3.1.1 (Average size of human segments for grasp: length = 270, width = 300mm) and GFR 2.1.1.1 (The mass of various body segments determined of max. 10.36 kg for thigh) respectively. In IFR 1.3.1.1, the designer has investigated and found out the average size of human segments that are needed to be grasped. Those measurements became a reference in the form of constraint for him to estimate the possible size of the end-effector without compromising its durability. Hence, the designer has created a constraint link to the GFR 4.0 (Need to be durable).

For this case study only one designer was involved. Table 6.8 shows the information about the participating designer and this information is stored in the ParticipatingDesigner table. Similar to Case Study 1, the user interface for this case study was not enhanced to provide an input to the designer to state his affiliation though the program creates the database table with the field “Affiliations”.

Table 6.8 A screen shot of the ParticipatingDesigner table that stored the information about the designer for the design of a conceptual end-effector in the Case Study 2

DesignerID	DesignerName	Affiliation
0001	Saw Min Htet	

All data captured by the descriptive design methodology prototype tool has been presented. In this case study, there is no information stored in the CADInfo table. The designer did not proceed to the detail design phase and hence, did not produce any CAD model of the end-effector. The designer only came up with some concepts of end-effectors for the first aid robot system.

In the process of coming up with the conceptual solutions for the end-effector to grasp human segments, the designer utilised the TRIZ-based ideation tool on three different IFRs. His decision to utilise the ideation tool occurred from 8th August 2008 to 9th August 2008 as shown in Table 6.9. The designer did not use the ideation tool on any GFR. From the data captured, it is noted that the first IFR for which the designer used the ideation tool was IFR 1.7 (Need to avoid injuring the patient). Then the designer used the ideation tool on IFR 3.1 (Need to determine the shape of end-effector to take max load) before trying it on IFR 1.6.1.1.1 (Need to find a way for the end-effector to grasp at human segments with different positions). Hence, the IFRSupportID may not necessarily be in sequence relative to its respective IFR.

From the IFRInfo table, the IFRSupportID only indicates the index of the utilisation of the ideation tool. More details of the ideation tool are stored in the IFRTrendSupport table. However, this table only stores the indexes that link to the other tables which actually store the details. Table 6.10 illustrates the data captured in the IFRTrendSupport table. From the table, the index for IFRImpFeaID and WorseFeaID represents the selection of improving and worsening features made by the designer while the index of the IFRProposedSolID represents the list of solutions generated by the ideation tool.

In order to know more about what IFRImpFeaID field meant, it is important to view the data in two other tables i.e. IFRTrendSupport/ImpFeaInfo and ImpFeaInfo table. Table 6.11 shows the data captured in the IFRTrendSupport/ImpFeaInfo and this table is a junction table. It can be observed that each time the designer sought support from the ideation tool, only one improving feature was selected. For example, for IFR 1.7 (Need to avoid injuring the patient), the support sought by designer using the ideation tool was tagged as IFRSupportID 000001.

Table 6.9 Sections of the IFRInfo table that indicate the utilisation of the TRIZ-based ideation tool (note the highlighted field "IFRSupportID") to assist the designer for the design of a conceptual end-effector in the Case Study 2

IFRID	IFR	IFRCatID	Start D.	Start Ti	Sourceof	Beli	Status	IFRActRe	PicDoc	DesignerID	CAI	Compo	IFRSupport
2.1	Need to determine max. support human segment	Requirement	2/21/2008	4:30 PM	Literature	0.50	Final	Active		0001		C00001	
2.1.1	Need to calculate various mass of body segments	Requirement	2/21/2008	4:45 PM	Literature	1.00	Final	Active	0000006	0001		C00001	
3.1	Need to determine the shape of end-effector to take max load	Requirement	2/22/2008	11:02 AM	Knowledge/ Experience	0.75	Final	Active		0001		C00001	000003
1.5.2.1	An end-effector with magnetic gel	Idea	3/3/2008	3:43 PM	Literature	0.50	Abandoned	Removed	0000007	0001		C00001	
1.6.1	Need to determine the various position of body possible before put into recovery position	Requirement	4/15/2008	10:40 AM	Expert	0.75	Final	Active		0001		C00001	
1.6.1.1	Need to simulate all possible positions of human segment	Requirement	4/18/2008	10:30 AM	Colleague	0.75	Final	Active		0001		C00001	
1.7	Need to avoid injuring the patient	Requirement	4/22/2008	9:45 AM	Knowledge/ Experience	1.00	Final	Active		0001		C00001	000001
1.6.1.1.1	Need to find a way for the end-effector to grasp at human segments with different positions	Requirement	5/12/2008	11:30 AM	Knowledge/ Experience	0.75	Final	Active		0001		C00001	
2.2	Need to determine the location of grasping	Requirement	6/10/2008	2:45 PM	Knowledge/ Experience	0.75	Final	Active		0001		C00001	
1.5.2.5.1	Grasping with additional stabilising support platform	Idea	8/8/2008	11:25 AM	Knowledge	0.50	Final	Active		0001		C00001	
1.5.2.5.1.1	Additional 1 support finger for stabilising purposes	Solution	8/8/2008	4:08 PM	Knowledge	0.50	Final	Active	0000005	0001		C00001	
1.6.1.1.1.1	Need to determine the preliminary action of an end-effector to deal with possible positions	Requirement	8/8/2008	4:48 PM	Experiment	0.75	Final	Active		0001		C00001	000002
1.6.1.1.1.1.1	Use a vision system	Solution	8/11/2008	9:23 AM	Colleague	0.50	Final	Active		0001		C00001	
1.7.1	Place sensors on the end-effector	Idea	8/11/2008	9:48 AM	Experiment	0.50	Final	Active		0001		C00001	

By looking at Table 6.11, the improving feature selected by the designer is represented by IFRImpFeaID 000001, which indicated that the designer actually selected the improving feature number (ImpFeaID) 31. In the second attempt to use the ideation tool to find solution for IFR 3.1 (Need to determine the shape of end-effector to take max load), the designer selected the improving feature number 1. Similarly, in the selection of worsening features, Table 6.12 shows the details of the worsening features selected by the designer.

Table 6.10 The IFRTrendSupport table that stored the information about the details of the TRIZ-based ideation tool utilisation by designer for the design of a conceptual end-effector in the Case Study 2

IFRSupportID	IFRSuppDate	IFRImpFeaID	IFRWorseFeaID	IFRProposedSol	IFRSupportStatus
000001	8/8/2008	000001	000001	000001	☑
000002	8/8/2008	000002	000002	000002	☑
000003	12/8/2008	000003	000003	000003	☑

Table 6.11 The IFRTrendSupport/ImpFeaInfo table that stored the information about the improving features selected by designer for the design of a conceptual end-effector in the Case Study 2

IFRImpFeaID	ImpFeaID
000001	31
000002	32
000003	1

Unlike the selection of the improving feature, the designer selected three worsening features in the first two usages of the ideation tool and two worsening features for the last usage. For example, in an attempt to solve IFR 1.7 (Need to determine the shape of end-effector to take max load), the designer selected the worsening features number (WorseFeaID) 45, 46 and 5. These improving and worsening feature numbers represent the features found in the TRIZ contradiction matrix by Mann (2003) and they are stored in ImpFeaInfo table (refer to Table 6.13) and WorseFeaInfo table (refer to Table 6.14). There are 48 features for both improving and worsening features.

Table 6.12 The IFRTrendSupport/WorseFeaInfo table that stored the information about the worsening features selected by the designer for the design of a conceptual end-effector in the Case Study 2

IFRWorseFeaID	WorseFeaID
000001	45
000001	46
000001	5
000002	45
000002	46
000003	45
000003	46
000003	5

Every time the designer used the ideation tool to solve an IFR, he had to select at least one improving feature and at least one worsening feature. Upon completing the selection of improving and worsening features, the ideation tool will generate a list of inventive principles. As shown Table 6.9, in the first attempt to use the ideation tool by the designer to solve IFR 1.7 (Need to determine the shape of end-effector to take max load), a list of recommended inventive principles was generated and stored. This list was tagged with the IFRSupportID 000001in IFRInfo table. In order to find out more about IFRSupportID, it is important to look into the IFRTrendSupport table (refer to Table 6.10) where IFRProposedSolID 000001 represents the index for the list of recommended solutions. From this index IFRProposedSolID 000001, the recommended inventive principles would be known based on their InventivePrincipleID. The IFRTrendSupport/InventivePrincipleSol table and the data stored in this table are shown in Table 6.15. From Table 6.15, it can be observed that there are 12 inventive principles that have been recommended for the IFR 1.7 (Need to determine the shape of end-effector to take max load) and hence, the repetition of the IFRProposeSolID 000001 twelve times. Each repetition represented an inventive principle and in this case the first recommended inventive principle for IFR 1.7 was inventive principles number 1, while the first recommended inventive principle for IFR 3.1 was inventive principle number 10 (refer to Table 6.15). The list of inventive principles was obtained from the TRIZ inventive principles list derived by Altshuller (1997) stored in InventivePrincipleSol table (Table 6.16).

Table 6.13 The ImpFeaInfo table that stores all the improving features listed in the contradiction matrix updated by Mann (2003)

ImpFeaID	ImpFeaDes
1	1: Weight of moving object
10	10: Amount of substance
11	11: Amount of information
12	12: Duration of action of moving object
13	13: Duration of action of stationary object
14	14: Speed
15	15: Force/Torque
16	16: Energy used by moving object
17	17: Energy used by stationary object
18	18: Power
19	19: Stress/Pressure
2	2: Weight of stationary object
20	20: Strength
21	21: Stability of the object
22	22: Temperature
23	23: Illumination intensity
24	24: Function Efficiency
25	25: Loss of Substance
26	26: Loss of Time
27	27: Loss of Energy
28	28: Loss of Information
29	29: Noise
3	3: Length/Angle of moving object
30	30: Harmful Emission
31	31: Other harmful effects generated by system
32	32: Adaptability/versatility
33	33: Compatibility/Connectivity
34	34: Trainability/Operability/Controllability/Ease of operation
35	35: Reliability/Robustness
36	36: Repairability/Ease of repair
37	37: Security
38	38: Safety/Vulnerability
39	39: Aesthetics/Appearance
4	4: Length/Angle of stationary object
40	40: Other harmful effects acting on system
41	41: Manufacturability/Ease of manufacture
42	42: Manufacturing precision/Consistency
43	43: Automation/Extent of automation
44	44: Productivity
45	45: Device complexity
46	46: Control Complexity
47	47: Ability to detect/Measure/Difficulty of detecting

Table 6.14 The WorseFeaInfo table that stores all the worsening features of the updated TRIZ contradiction matrix by Mann (2003)

WorseFeaID	WorseFeaDes
1	1: Weight of moving object
10	10: Amount of substance
11	11: Amount of information
12	12: Duration of action of moving object
13	13: Duration of action of stationary object
14	14: Speed
15	15: Force/Torque
16	16: Energy used by moving object
17	17: Energy used by stationary object
18	18: Power
19	19: Stress/Pressure
2	2: Weight of stationary object
20	20: Strength
21	21: Stability of the object
22	22: Temperature
23	23: Illumination intensity
24	24: Function Efficiency
25	25: Loss of Substance
26	26: Loss of Time
27	27: Loss of Energy
28	28: Loss of Information
29	29: Noise
3	3: Length/Angle of moving object
30	30: Harmful Emission
31	31: Other harmful effects generated by system
32	32: Adaptability/versatility
33	33: Compatibility/Connectivity
35	35: Reliability/Robustness
36	36: Repairability/Ease of repair
37	37: Security
38	38: Safety/Vulnerability
39	39: Aesthetics/Appearance
4	4: Length/Angle of stationary object
40	40: Other harmful effects acting on system
41	41: Manufacturability/Ease of manufacture
42	42: Manufacturing precision/Consistency
43	43: Automation/Extent of automation
44	44: Productivity
45	45: Device complexity
46	46: Control Complexity
48	48: Measurement accuracy/Measuring Precision
5	5: Area of moving object

Table 6.15 The IFRTrendSupport/InventivePrincipleSol table stores all the links between IFRProposeSolID and the InventivePrincipleID

IFRProposeSolID	InventivePrincipleID
000001	1
000001	10
000001	12
000001	17
000001	19
000001	2
000001	23
000001	26
000001	3
000001	31
000001	35
000001	4
000002	19
000002	25
000002	28
000002	29
000002	31
000002	37
000002	6
000003	10
000003	12
000003	17
000003	19
000003	2
000003	26
000003	28
000003	29
000003	30
000003	35
000003	40

Table 6.16 The InventivePrincipleSol table that stores all 40 inventive principles based on the TRIZ contradiction table by Altshuller (1997)

InventivePrincipleD	InventivePrincipleDes
1	Segmentation
10	Preliminary Action
11	Beforehand Cushioning
12	Equipotentiality
13	"The other way round"
14	Spheroidality - Curvature
15	Dynamisation
16	Partial or Excessive Actions
17	Another Dimension
18	Mechanical Vibration
19	Periodic Action
2	Taking Out
20	Continuity of Useful Action
21	Skipping
22	"Blessing in Disguise"
23	Feedback
24	"Intermediary"
25	Self-Service
26	Copying
27	Cheap Short-Living Objects
28	Mechanics Substitution
29	Pneumatics and Hydraulics
3	Local Quality
30	Flexible Shells and Tin Flims
31	Porous Materials
32	Colour Changes
33	Homogeneity
34	Discarding and Recovering
35	Parameter Changes
36	Phase Transitions
37	Thermal Expansion
38	Strong Oxidants
39	Inert Atmosphere
4	Asymmetry
40	Composite Materials
5	Merging
6	Universality
7	"Nested Doll"
8	Anti-Weight
9	Preliminary Anti-Action

The results obtained in case study 2 were analysed in details and evaluated in the next section from the perspective of meeting the aims of the case study.

6.3.2 Results from the Interview with the Designer on the Descriptive Design Methodology in Case Study 2

This case study involved a novice designer who is learning to design using a step-base methodology that involved design phases. The designer was only required to derive a conceptual solution for an end-effector to grasp human segments in the first-aid robot system (FAROS). A similar interview conducted in Case study 1 was carried out with the designer and the questions raised were shown in Appendix 2, with an additional question about the TRIZ-based ideation tool.

The designer in Case Study 2 also found the descriptive design tool to be good and also useful. Unlike Case Study 1, the designer had read about the step-oriented design approach of Pahl and Beitz (1995) and had decided to adopt a similar approach to derive a conceptual design for the end-effector. The designer also opined that he probably will use the descriptive design software tool in future design work. Nevertheless, he thought that the decomposing and linking between GFR with IFR and IFR to sub-IFR are the core strength of the descriptive design methodology. However, he did think that the descriptive design is lacking in helping him to find solutions to design problems. However, integrating it with the TRIZ-based ideation tool is good.

From the perspective of integrating the TRIZ-based ideation tool with the descriptive design tool, the designer thought the integration was essential and he preferred to use the ideation tool more as it provided possible though general solutions to solve his design problems. Finally, the designer also felt it was not easy to select the appropriate improving and worsening features, as well as the difficulty involved in translating the recommended inventive principles into design solutions. He hoped that the TRIZ-based ideation could be improved to help the designer more in this matter.

6.4 Analysis and discussion of the Implementation Design Methodology with TRIZ-based Ideation Tool and the Results Obtained in Case Study 2

The results obtained from this case study from the perspective of descriptive design methodology framework have shown that the framework was also able to support and assist the designer that adopted a step-oriented design model. This implied that the framework was able to accommodate designers that used different strategies and approaches to design. The ability to accommodate different design approaches and strategies is crucial as it meant the methodology does not impose a prescriptive approach onto a designer. Alternatively, the framework can also accommodate other design tools such as TRIZ, which influences the decision of the designers if they prefer it. In addition, the framework is also able to capture the thoughts and ideas of the designer throughout the design process, consistent with the findings in the Case Study 1, even though it has integrated with another design tool in this case study. This is because the descriptive design framework was able to work in an integrated manner with the TRIZ-based ideation tool and yet managed to retain its capacity to track, trace, and enable the designer to delay design decisions when required, as well as other computational functions. These results confirmed that the support facilities provided by the descriptive design framework are still valid even when it integrated with another design tool.

From the data captured by the descriptive design methodology tool, the information captured by the LinkIFR table showed that the link between IFR and GFR as well as IFR and IFR were important and some of the solutions proposed by the designer became constraints or even solved others. For example, in this case study, IFR 1.3.1.1 (Average size of human segments for grasp: length = 270, width = 300mm) and IFR 2.1.1.1 (The mass of various body segment determined (max. 10.36 kg for thigh)) became a constraint to GFR 4.0 (Need to be durable) and GFR 3.0 (Need to take max load human segment) respectively. These links were created by the designer to enable him to track the constraints that he needed to consider as he progressed. However, later the designer decided to use composite material to increase the durability of the end-effector. The decision to use composite material was related to one of the recommended inventive principles proposed by the ideation tool. This matter will be further elaborated in the analysis and discussion of the TRIZ-based ideation tool below.

In Case Study 2, it may also be observed that from IFR 1.5.2.1 to IFR 1.5.2.4 (refer to Table 6.17), several ideas came up and two of them were originated by experts. The designer has given these ideas some consideration and conducted some studies on these ideas resulting in the documentation with PicDocID 0000007.

Table 6.17 The four IFRs that were studied by the designer to see whether their technology can be adapted for the FAROS end-effector

IFRID	**IFR**
1.5.2.1	An end-effector with magnetic gel
1.5.2.2	An end-effector with air muscle
1.5.2.3	An end-effector based on Bourdon Tube
1.5.2.4	A end-effector with plastering approach

From the observations on the results of the case study, the descriptive design methodology prototype tool is able to help a designer, particularly a novice one, to decompose or breakdown the requirements to sub-requirements and from task to sub-tasks. With these decompositions carried out, the designer is able to recognise the need to identify and find out the required parameters to assist decision making. For example, as shown in Table 6.18, the descriptive design framework managed to frame IFR 1.3 (need to grasp various sizes) into a need to determine the average size of the human segment for grasping and then later to obtain this information from the literature. Similarly, for IFR 1.4 (Need to grasp various shapes), the framework provided a structure for the designer to think of the possible human segments shape to grasp, which then led to the finding out of the average shapes for the human segments (refer to Table 6.19).

Table 6.18 The captured data on how the need to grasp various sizes of human segments decompose into the determination of the average size of human segments for grasping

IFRID	IFR
1.3	Need to grasp various sizes
1.3.1	Need to determine the average size of human segment for grasping
1.3.1.1	Average size of human segments for grasp: length = 270, width = 300mm

Table 6.19 Another example of the decomposition from the need to grasp various shapes of human segments to the determination of the average shape of human segments for grasping

IFRID	IFR
1.4	Need to grasp various shapes
1.4.1	Need to determine the various shapes of the human segments
1.4.1.1	Average shape of human segments are cylindrical

The results obtained from this case study also provide an indication that the descriptive design framework is able to support a novice designer to decompose design requirements and tasks into sub-requirements and sub-tasks but it has no means of helping the designer to come up with a design solution for a design problem.

From the perspective of supporting an ideation tool based on TRIZ, the descriptive design framework can be modified and adapted to capture the data generated by optional and external design tool such as the TRIZ-based ideation tool. The descriptive design framework is sufficiently versatile to work together in a synergetic way to enhance the support for the designer. This case study has shown that the TRIZ-based ideation tool can work on top of the descriptive design framework in an integrated architecture to support the designer when they wish to use it.

Unlike the previous case study where the designer was experienced and who basically did not use any design methodology or approach, in this case study, the novice designer adopted a step-oriented design model similar to the one proposed by Pahl and Beitz (1995). The designer aimed to derive a conceptual design solution for a FAROS end-effector in this case study. From this case study, the results showed that the descriptive design framework can support a step-oriented base design approach with design phases. This was because the descriptive design framework was able to continue to capture design thoughts and ideas throughout the conceptual design phase.

The last part of the analysis and discussion will focus on the effectiveness of the TRIZ-ideation tool. The results obtained from the utilisation of the ideation tool consist of three main technical contradictions that were identified by the designer. The first technical contradiction identified by the designer was when the TRIZ-based ideation tool was used for IFR 1.7 (Need to avoid injuring the patient). Then, the ideation tool was used for IFR 1.6.1.1.1.1 (Need to determine the preliminary action of an end-effector to deal with possible positions) and finally for IFR 3.1 (Need to determine the shape of end-effector to take max load).

As mentioned in previous paragraph, the tool was utilised three times by the designer at different times and dates (refer to Table 6.9). One of the key factors that showed that the TRIZ-based ideation tool is an optional tool and its utilisation is based on the preference of the designer is the time and date of its utilisation. The TRIZ-based ideation tool could be at any time and at any stage of the design process. Analysing the data collected in IFRInfo table (refer to Table 6.10), the utilisation of the ideation tool was not in accordance to the IFR time and date sequence. For example, though IFR 3.1 (Need to determine the shape of end-effector to take max load) was created earlier than IFR 1.7 (Need to avoid injuring the patient), the designer utilised the ideation tool for the latter first. In addition, all three utilisations of the ideation tool were in early August 2008 but IFR 3.1 and IFR 1.7 were created much earlier, in late February in 2008 and in late April 2008 respectively while IFR 1.6.1.1.1.1 (Need to determine the preliminary action of an end-effector to deal with possible positions) was created in early August 2008. This meant that the designer sought support from the ideation tool after months of trying to solve the three IFRs. Hence, in this case study, the ideation tool was only utilised at late stages of the design project when it was needed.

The first utilisation of the ideation tool was for IFR 1.7, namely "Need to avoid injuring the patient". IFR 1.7 was translated to an improving feature of no. 31 (Other harmful effects generated by system) and the worsening features of no. 45 (Device complexity), no. 46 (Control Complexity) and no. 5 (Area of moving object) as shown in Figure 6.10. The TRIZ-based ideation tool generated a list of solutions for IFR 1.7 (refer to Figure 6.10). The results obtained from the first utilisation of the TRIZ-based ideation tool gave some ideas of a possible solution to the designer.

Among the recommended inventive principles, the "Another Dimension" principle (inventive principle no.17) gave the designer the first idea to have a two-dimensional surface contact end-effector, namely an end-effector with two fingers (IFR 1.5.2.5). Then the "preliminary action" principle suggested to the designer the idea of adding an additional supporting finger for stabilising and slippage avoidance to enhance the grasping of the two fingers end-effector created earlier (IFR 1.5.2.5.1 to IFR 1.5.2.5.1.1). The "Preliminary action" principle also influenced the designer to first create IFR 1.6.1.1.1.1 (Need to determine the preliminary action of an end-effector to deal with possible positions) and to find what preliminary action can be used to deal with the different possible positions of the human segments when the patient is unconscious. Though the inventive principle "preliminary action" is one of the recommended inventive principles to solve IFR 1.7 and not IFR 1.6.1.1.1, nevertheless, this inventive principle suggested to him the thought of finding any preliminary action to ease the detection of the various possible positions of human segments for an unconscious patient.

IFRID	**IFR**
1.7	Need to avoid injuring the patient

Improving Feature	**Worsening Feature**
31: Other harmful effects generated by system	45:Device complexity
	46:Control Complexity
	5: Area of moving object

Inventive PrincipleID
19 Periodic Action
10 Preliminary Action
17 Another Dimension
31 Porous Materials
4 Asymmetry
1 Segmentation
23 Feedback
26 Copying
12 Equipotentiality
2 Taking Out
35 Parameter Changes
3 Local Quality

Figure 6.10 The translation of IFR 1.7 (Need to avoid injuring the patient) to improving and worsening features and the recommended inventive principles from the TRIZ-based ideation tool

With several new IFRs with solutions created after the inspiration from the recommended inventive principles, the designer has still yet to come up with a viable solution for IFR 1.7 (Need to avoid injuring the patient). However, when the designer applied the ideation tool to find out the inventive principles for IFR 1.6.1.1.1.1 (Need to determine the preliminary action of an end-effector to deal with possible positions), the new recommended inventive principles generated gave him ideas about the possible solutions to several IFRs.

The application of the ideation tool for IFR 1.6.1.1.1.1 (Need to determine the preliminary action of an end-effector to deal with possible positions) involved translation of the IFR into the improving feature no. 32 (Adaptability/Versatility) against the worsening features no. 45 (Device Complexity) and no. 46 (Control Complexity) (refer to Figure 6.11). The generated inventive principles were as shown in Figure 6.11. The top recommended inventive principle was the inventive principle no. 28 (Mechanical substitution). This inventive principle then gave the designer the idea to use sensors such as a tactile sensor, a proximity sensor and vision system to solve the problem of detecting the various positions of human segments of the patient. This led to IFR 1.6.1.1.1.1.1 (Use a vision system) and IFR 1.7.1 (Place sensors on the end-effector). However, the task of developing a vision system for FAROS was the responsibility of another designer. Hence, the designer needed to inform and approach the designer in charge of developing the vision system for discussion.

The idea of using sensors and a vision system to detect the various positions of human segments when the patient is unconscious also inspired the designer to use vision and sensor systems to avoid injuring the patient. Hence, IFR 1.7.2 (Need to use vision systems to help avoid injuring the patient) was created and again the designer had to inform the researcher that was developing the vision system for FAROS. The information that the designer needed to discuss the matter with his colleague who was responsible for the vision system of FAROS, was also captured by the descriptive design framework.

IFRID	IFR
1.6.1.1.1	Need to find a way for the end-effector to grasp at human segments with different positions

Improving Feature	Worsening Feature
32:Adaptability/versatility	45: Device complexity
	46: Control Complexity

Inventive PrincipleID
28 Mechanics Substitution
6 Universality
37 Thermal Expansion
19 Periodic Action
29 Pneumatics and
31 Porous Materials
25 Self-Service

Figure 6.11 The translation of IFR 1.6.1.1.1 (Need to determine the preliminary action of an end-effector to deal with possible positions) to improving and worsening features and the recommended inventive principles from the TRIZ-based ideation tool

In the final utilisation of the adapted TRIZ tool, the designer translated IFR3.1 (Need to determine the shape of end-effector to take max load) into improving feature no. 1 (Weight of moving object) against three worsening features no. 45 (Device complexity) and no. 5 (Area of moving object) and no. 46 (Control complexity), as shown in Figure 6.12. Though there is an improving feature for shape in the TRIZ contradiction matrix, the designer chose to use the weight of the moving object as the improving feature. The ideation tool generated a list of 11 inventive principles. Among these principles, the first is parameter change (inventive principle no. 35). This inventive principle was used by the designer to create a suitably curved flexible two-jaw end effector to handle the cylindrical shape of human segments (IFR 3.1.1). The next recommended inventive principle was “composite materials” and this principle was adopted (IFR 4.1) by the designer to improve the durability of the end-effector (GFR4.0) and to use silicone rubber as the inner surface for the end-effector to prevent slippage (IFR 1.1.1.1).

IFRID	IFR
3.1	Need to determine the shape of end-effector to take max load

Improving Feature	Worsening Feature
1:Weight of moving object	45: Device complexity
	5: Area of moving object
	46: Control Complexity

Inventive PrincipleID
35 Parameter Changes
40 Composite Materials
30 Flexible Shells and
12 Equipotentiality
10 Preliminary Action
28 Mechanics
19 Periodic Action
29 Pneumatics and
2 Taking Out
17 Another Dimension
26 Copying

Figure 6.12 The translation of IFR 3.1 (Need to determine the shape of end-effector to take max load) to improving and worsening features and the recommended inventive principles from the TRIZ-based ideation tool

With these three trials of the ideation tool, the designer managed to derive a conceptual solution for the end-effector of FAROS. The effectiveness of the inventive principles in supporting a designer to derive conceptual design solutions is obvious. However, the designer did raise the point that it is not easy to translate design problems into improving and worsening features as well as interpreting the recommended inventive principles. There are also issues related to the recommended inventive principles, which suggest that the top recommended inventive principles may not provide an appropriate design solution. For example, the top recommended inventive principle for IFR 1.7 was periodic action (inventive principle no. 19) but the designer could not come up with any solution based on it. Finally, Figure 6.13 illustrates the graphical user interface display of the information captured in Case

Study 2. Overall the analysis and discussions of this case study can be summarised as shown in Table 6.20.

Table 6.20 Summary of analysis of Case Study 2

	Aims of Case Study 2	**Results and Findings**
i.	Reaffirming the advantages of and support that can be provided by the descriptive design methodology framework verified in Case Study 1	The outcome of this case study was consistent with the outcome of the previous case study. The framework managed to capture the GFRs, the IFRs and linked them to their relevant the documentations if available as what happened in the previous case study.
ii.	To assess the effectiveness of the descriptive design methodology in supporting a novice designer.	The framework was found to help the designer to decompose design requirements into sub-requirements and others.
iii.	To evaluate the feasibility of the descriptive design methodology to support an ideation tool based on TRIZ.	The descriptive design methodology was found to work in complementary way with the TRIZ-based ideation tool. The descriptive design methodology was able to help the designer to decompose design requirements and guide the designer to sub-requirements and ideas at lower level. After that, the TRIZ-ideation tool was then able to further help the designer to come up with conceptual solutions.
iv.	To find out whether the descriptive design methodology is able to support a step-oriented based design approach (an approach that has design phases).	The data captured by the descriptive design tool indicated that the information captured was less detailed and rather conceptual. There were no link to CAD model created and the analysis on the data showed that the designer was trying to come up with a working concept for the end-effector of FAROS.

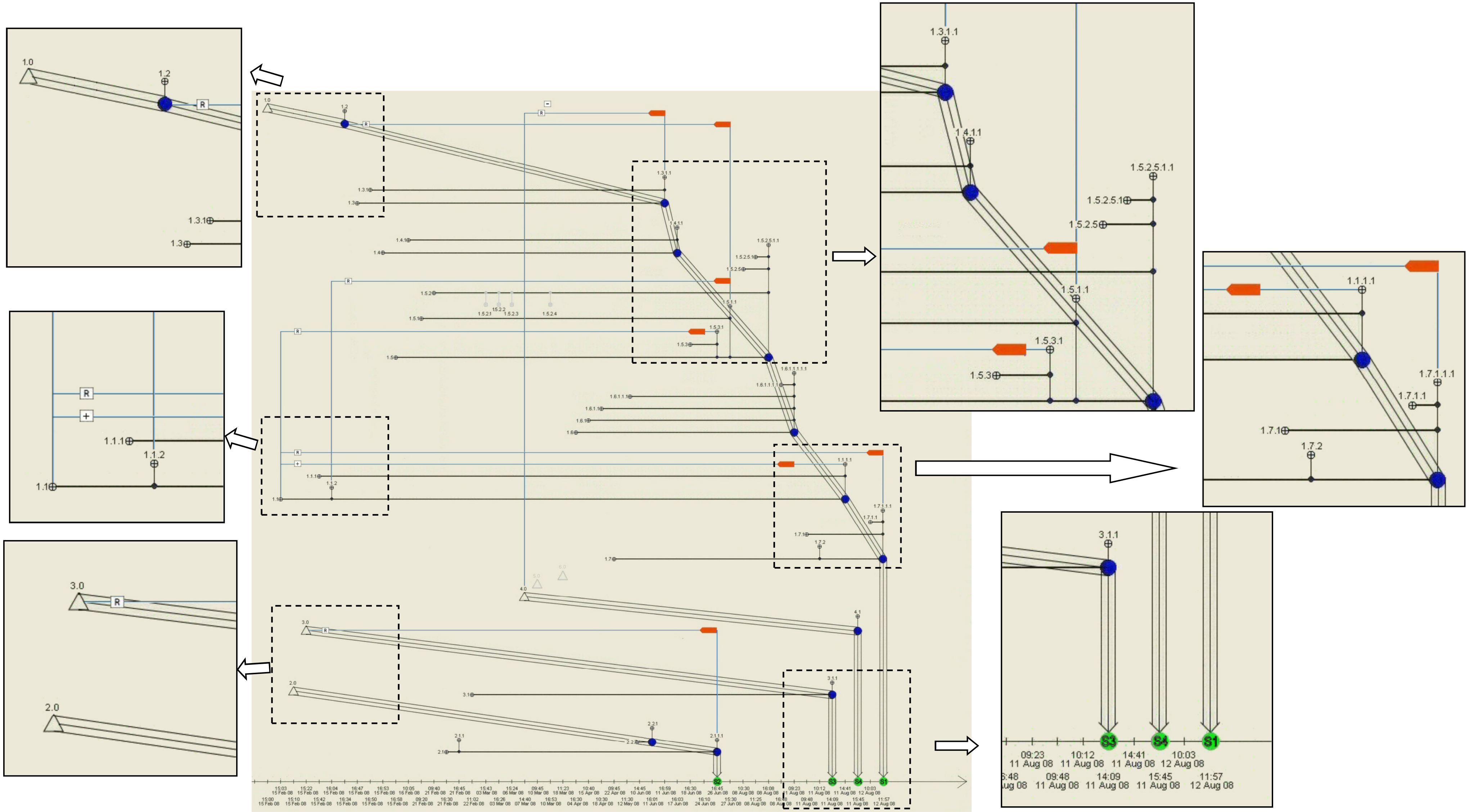


Figure 6.13 The snapshot of the graphical user interface of the descriptive design prototype tool when Case study 2 was completed with zoom in views

6.5 Summary

The analysis and discussion of the application of the descriptive design methodology with an optional ideation tool show that it was flexible enough to be integrated with other design tools to enhance the support of the designer and reaffirmed that it was able to accommodate different design approaches. The results demonstrated that the descriptive design framework was able to capture the thoughts and ideas of the designer in a structured manner. It was again able to show that the descriptive design framework can manage linked files such as sketches and calculations.

The contribution of the inventive principle, "preliminary action" was significant. This contribution not only helped the designer to come up with the idea to utilise the vision system to assist in preventing injury to the patient (IFR 1.7) but also to utilise the vision system to detect the positions and locations of human segments (IFR 1.6.1.1.2) and lastly the addition of a support finger to prevent slippage (IFR 1.5.2.5.1.1).

Chapter 7
Conclusions, Contributions and Further Work

In this chapter, the findings and contribution of this research work will be reviewed and reflected upon from the perspective of achieving the stated research objectives and on the potential future work that can be endeavoured.

7.1 Conclusions

Engineering design methodology was established more than two decades ago and one of its key aims is to support designers to enable the design process to be carried out in a systematic manner. Within these two decades, a huge amount of literature and research work on design methodologies has been published but there were implications from surveys and studies that designers are still not adopting these design methodologies in their design work. There were several possible reasons behind the designers' lack of interest in adopting these design methodologies, particularly among the experienced designers. Is it because experienced designers do not need any support? Literature findings show that even experienced designers do indeed need support and help in design. This research work has reviewed and analysed a number of established design methodologies and design support facilities to derive a descriptive design methodology framework to support designers. The framework was then applied in two case studies to verify its effectiveness.

7.2 Contributions

The findings and the contribution of this research work in respect of the objectives set earlier are presented in this chapter as follows:

1. *To explore, review and compare established design methodologies from the perspective of supporting designer and their influences on the designers approach and strategy.*

From the perspective of decision-making, design methodologies can be categorised into three types: the normative, the prescriptive and the descriptive. Most of the established methodologies are of the prescriptive type. This type of design methodology attempts to prescribe a design method to the designers in the form of guidelines and offers a breakdown of the design process into design phases. By assuming that a design process can be managed better when it is performed in phases, the design process will be more systematic. The prescriptive design methodologies also assume that the process of design is one of searching for many design solutions within the solution design space and then selecting the best one by using various design analysis techniques. Empirical studies have showed that experienced designers work in different manners and that one of their strategies is to improve their initial design solution to meet design requirements.

Most of the descriptive types of design methodologies are used for the purpose of studying design. Among the popular descriptive design methodologies are the protocol analysis and the logbook approaches. Descriptive design methodologies lack the appropriate structure to support designers sufficiently and the methodology captures a huge amount of data some irrelevant to the design project. Finally, the normative design methodologies are types that require designers to quantify design parameters based on utility values, mathematical models or to follow certain axioms and design to achieve the best utility values or adhere to axioms or mathematical models. One of the normative design methodologies, the axiomatic design method, is found to bias the designer's decisions. The findings from the literature suggested there is a need for a design methodology that allows designers to design with their own approaches and strategies.

In addition, one of the key findings of the exploration, reviewing and comparing various established design methodologies was that these methodologies can be grouped according to their orientation, namely, the step-oriented type and the function-oriented type. Both models have their strengths and deficiencies but the step-oriented model is a type of design model that breaks down the design process into design phases. However, apparently most of the established design methodologies are step-oriented but when compared to the function-oriented types, they were found to be more focused on trying to achieve the design requirements.

Within the three types of design methodologies, only the descriptive type actually describes what the designer is doing in design irrespective of their approaches and strategies. Nevertheless, most descriptive types are not applied to support designers and are lacking in structure to do so. Hence, one of the key challenges in this research is to derive a novel descriptive design with a function-oriented basis that has crucial design support facilities for a designer. This leads to the research of the next objective: what are the crucial design support facilities that can be provided to a designer?

For this objective, the key findings and contributions can be summarised as below:

i) Designers were found to unwilling to adopt the design approaches and strategies suggested by the established design methodologies even though they needed design support and help. Hence there is a need for a design methodology to support designers without influencing their approaches and strategies.

ii) Function-oriented design models such as the axiomatic design are more focused on helping the designer achieve the design requirements and also provide some level of traceability.

iii). A descriptive design methodology is one that does not influence the approach and the strategy of a designer but it lacks the structure to support a designer. Hence, there is a need to derive a descriptive design with function-oriented basis that can support designers. The next objective will investigate the type of design support facilities that are crucial and can be incorporated into a design methodology.

2. *To determine and evaluate the crucial design support facilities, namely the design methodology-related support and the computational-platform-related support, that can be applied in a design methodology to help designers*

There are main design support facilities identified: design-methodology-related support, computation-platform-related support, concept selection support and concept ideation support.

For design-methodology-related support, the support facilities identified are:

i) to record and capture ideas and thoughts

ii) to trace and track their ideas and thoughts

iii) to decide when to delay design decisions when there is insufficient information

iv) to add, edit or remove design decisions anytime

v) to provide an indication of the effects of any change made in the past, or present

vi) to enable the re-use of past ideas, solutions and information

For computation-platform-related support, the support facilities identified are:

i) the saving of all records

ii) the searching for and visualisation of design decisions throughout the design process

iii) a flexible interface to capture ideas, thoughts, and others.

For concept selection support and ideation support, there are established tools to support designers and it is important for a design methodology to work in a synergised manner to support the designer better. Among the more widely used concept selection approaches is the "satisficing" method, in which the designer need only determine in sequence whether a concept satisfies the requirements or not. For an ideation tool, the established technical contradiction matrix of TRIZ is an example of a tool that can be used to support designers.

With the identification of the support facilities, the evaluation showed that design-methodology-related support and the computational-platform-related support are among the crucial support facilities needed, while any novel design methodology framework needs to be able to work with other established and optional tools such as "satificing" and the technical contradiction matrix tool of TRIZ.

3. *To determine and link common characteristics of design tasks with the aims of a designer to determine the critical design parameters for the purpose of deriving a novel design methodology*

The key challenge to conceptualise and derive a novel descriptive design methodology that can support designers is to determine what is the design parameters needed to enable the facilitation of all the support identified in the previous objective. In order to determine these design parameters, there is a need to link the common characteristics of design tasks with the aims of a designer.

The findings from linking the common characteristics of design tasks with the aims of a designer are as follows:

i) Design parameters that are important to capture are knowledge, information, time, communication, requirement, presentation and human memory, change and source of information.

ii) The knowledge parameter can be translated into the ideas and thoughts of designers as they are generated from the knowledge and experience of the designer.

From these important design parameters, the novel descriptive design methodology that supports designers can be conceptualised.

4. *To propose and derive a novel design methodology that can provide these key design support facilities without influencing the designer's approach and strategy.*

Although the important design parameters that need to be captured determined the crucial design support facilities identified and should be a descriptive type with function-oriented basis, the task of deriving a representation framework is still daunting. The novel descriptive design methodology framework that provides these key design support facilities is based on a new dynamic graph diagram that was inspired by the fishbone diagram of Ishikawa. The new representation framework of the descriptive design has a time line (x-axis) and will depict the flow of the design process (refer to Figure 4.3).

5. *To integrate an optional design tool based on TRIZ with the novel descriptive design methodology to evaluate its ability to work with established design tools.*

In order to demonstrate the ability of the new descriptive design framework to support other design tools so that they are able to work together in a synergised manner, an ideation tool based on the contradiction matrix of TRIZ was developed. The novice designer faced difficulties in deriving a conceptual solution for the end-effector of FAROS, even with descriptive design methodology. The descriptive design methodology assisted the designer to decompose the design requirements into sub-requirements, criteria and others but has no means to help the designer to solve it. The TRIZ-based ideation tool was able to work together in a complementary way to help the designer solve the problem. The ideation tool shared the database system with the descriptive design framework. Additional adaptations were also made to the contradiction matrix of TRIZ to include the adding of weightage to the improving and worsening features and an accumulative method to rank the recommended inventive principles. This accumulation method is based on the number of repetitions for a recommended inventive principle among all the recommendations obtained when every individual improving feature and the worsening feature were matched.

6. *To assess and verify the effectiveness of the novel design methodology to two case studies.*

In Case Study 1 the descriptive design framework was used in a project to design a device to support concrete loading in between beams in the construction industry. Case study 1 (Phase 1) managed to show the ability of the descriptive design framework in capturing the ideas and thoughts of the designers and in managing the information such as documentation, CAD data and scanned pictures of sketches. The descriptive design framework also proved that it could link different IFRs and GFRs as well as enable the designers to made design changes such as removing design decisions. In addition, the framework also managed to allow the designer to delay their decision until sufficient design information was available.

In Phase 2 of Case study 1, the descriptive design was able to accept additional design requirements and design improvements at the late stage of the design process and to capture computer-aided design analysis results documentation at late stages that affected the final design. The descriptive design approach was also able to retrieve the data from Phase 1 for re-use and modifications. In this project, the designer was an experienced one who did not use any design methods in his work.

For Case Study 2, the descriptive design framework was applied together with a TRIZ-based ideation tool in a student project. The project was to design a conceptual end-effector for a first-aid rescue robot (FAROS). The designer was a novice and the descriptive design framework assisted him in decomposing the design requirements into sub-requirements and into solutions. In addition, Case Study 2 also managed to show the descriptive design framework work together with an ideation tool that was based on the TRIZ contradiction matrix. The ideation tool was utilised as an optional tool to help the designer when they have difficulties in trying to solve the design problem and at a very late stage of the design process. In this case study, the descriptive design framework was able to support and work together with other design tool, such as the ideation tool that was based on the TRIZ contradiction matrix. The ideation tool was based on an improved method of ranking inventive principles. Finally, the designer applied the descriptive design framework to a conceptual design phase that was based on a step-oriented design model and hence showed that the descriptive design framework can also support other design models.

7.3 Future work

This research has contributed to the domain of engineering design in proposing a descriptive design framework to support designers. The framework was derived with a functional basis and within the scope of a Ph.D. research. Within this scope, only functional requirements were considered. In reality, product design involves requirements that cover aesthetics, affectivity, ergonomics and branding. These requirements are also crucial to the success of a product. Further research work that involves consideration of these requirements will enable the effectiveness of the proposed framework to support designer to be better evaluated.

As noted in the graphical representation of the descriptive design framework in Figure 4.3, the research work on the descriptive design framework only covers the top part of the graphical representation. This top part (above the x-axis) is utilised to capture design activities controlled by the designers. However, the bottom part of the framework representation (below the x-axis) is allocated for the design activities beyond the control of the designers, namely factors such as legislation, safety regulations and rules set by the company or the government. These factors affect design decisions but are beyond the scope of this research work. With the recent significant worldwide focus on green, environmentally friendly and energy-efficient products, designers have to give serious consideration to these requirements that are beyond their control. Hence, future work may include consideration of these factors.

Though the descriptive design framework can capture the name of the designer as well as the source and of the designers that contributed to the design decisions, thoughts and ideas, the information captured on communication between designers, suppliers, and other stakeholders is very limited. The importance of communication between designers, with suppliers and other stakeholders are crucial and there is a need for further expansion of the current descriptive design framework to accommodate support facilities focusing on facilitating discussions, negotiations and group decision-making.

This research only demonstrated the descriptive design framework working together with the TRIZ-based ideation tool. There are a lot of established design tools proposed by researcher to help designers. Future research work can integrate more established design tools with the descriptive design framework to provide more design support and help to the designer.

Finally, the descriptive design framework shown is able to manage documentation and CAD files. The framework acted as a kind of macro manager of the design process. However, future work to apply the descriptive design framework at the micro level, namely to utilise the descriptive framework to represent a computer-aided design (CAD) model of a product, will be challenging task for the research community. The integration of the descriptive methodology with a CAD model to form a design representation of the geometric modelling will be a significant research contribution in the area of design methodology. This is because CAD only captures the data about the product's geometry as well the modelling process involved in generating the model. The design intent and reasons behind why the geometric model is created in a particular shape and assembled in a specific way are not captured.

Appendix 1: Feedback form for the descriptive design software tool

Case Study 1

Particulars of the designer: Date: ____________

1. Name:
   ____
2. Position:
   ____

Questions:

1. What do you think of the descriptive design software tool help in assisting and supporting your design work?
   a. Excellent b. Good c. Useful d. Not very good e. Poor

2. Do you use any design methodologies in your design work?
   a. Yes b. No c. Not sure
   If **yes**, please name which methodology
   ____

3. Do you think you will use the descriptive design software tool in your future design work?
   a. Yes b. Probably c. No
   If **No**, please give reason
   ____

4. In your opinion, what do you think are the core strengths of the descriptive design software tool?
   a. ____
   b. ____
   c. ____

5. In your opinion, what do you think are the core improvements that can be done on the descriptive design software tool?
   a. ____
   b. ____
   c. ____

6. In your opinion, are there any deficiencies about the descriptive design software tool?
   a. ____
   b. ____
   c. ____

Appendix 2: Feedback form for the descriptive design software tool

Case Study 2

Particulars of the designer: Date: ___________________

1. Name:
   ____
2. Position:
   ____

Questions:

3. What do you think of the descriptive design software tool help in assisting and supporting your design work?
b. Excellent b. Good c. Useful d. Not very good e. Poor

4. Do you use any design methodologies in your design work?
b. Yes b. No c. Not sure
   If **Yes**, please name which methodology
   ____

5. Do you think you will use the descriptive design software tool in your future design work?
b. Yes b. Probably c. No
   If **No**, please give reason
   ____

6. In your opinion, what do you think are the core strengths of the descriptive design software tool?
   a. ____
   b. ____
   c. ____

7. In your opinion, what do you think are the core improvements that can be done on the descriptive design software tool?
   a. ____
   b. ____
   c. ____

8. In your opinion, are there any deficiencies about the descriptive design software tool?
   a. ____
   b. ____
   c. ____

9. What are your opinions on the integration between descriptive design software tool and the TRIZ-based ideation tool?
   a. ____
   b. ____

Thank you for your participation.

References

Agarwal, M. and Cagan, J. 1998. A blend of different tastes: the language of coffeemakers. *Environment and Planning B: Planning and Design*.25(2), pp. 205-226.

Ahmed, S. and Hansen, C. T. 2002. An analysis of design decision-making in industrial practice. In: Marjanovic, D. ed. *Proc. 7th Int. Design Conf. 2002*, Dubrovnik, Croatia. pp. 145-150.

Ahmed, S. and Wallace, K. M. 2004. Identifying and supporting the knowledge needs of novice designers within the aerospace industry. *Journal of Engineering Design*.15(5), pp. 475-492.

Ahmed, S.,Wallace, K. M. and Blessing, L. T. 2003. Understanding the differences between how novice and experienced designers approach design tasks. *Research in Engineering Design*.14(1), pp. 1-11.

Akin, Ö. and Lin, C. 1996. Design protocol data and novel design decision. In: Cross, N. et al. eds. *Analysing design activities*. John Wiley & Sons, p. 463.

Altshuller, G. 1997. *40 principles: TRIZ keys to innovation*. Worcester, MA: Technical Innovation Center.

Badke-Schaub, P. 2003. Strategies of experts in engineering design: between innovation and routine behaviour. In: Cross, N. and Edmonds, E. eds. *Expertise in Design: Design Thinking Research Symposium 6*. Sydney, Australia: Creativity & Cognition Studios, pp. pp.51-68.

Baron, J. 2004. Normative models of judgment and decision making. In: Koehler, D.J. and Harvey, N. eds. *Blackwell Handbook Of Judgment And Decision Making (Blackwell Handbooks of Experimental Psychology)*. London: Blackwell, pp. 19-36.

Baron, R. S.,Kerr, N. L. and Miller, N. 2003. *Group process, group decision and group action*. 2nd ed. Buckingham, UK: Open University Press.

Barron, F. H. and Barrett, B. E. 1996. The efficacy of SMARTER - Simple multi-attribute rating technique extended to ranking. *Acta Psychologica*.93(1-3), pp. 23-26.

Beach, L. R. 1990. *Image theory : Decision making in personal and organizational contexts*. John Wiley & Sons.

Beach, L. R. and Mitchell, T. R. 1978. A contingency model for the selection of decision strategies. *Academy of Management Review*.3(3), pp. 439-449.

Beach, L. R. and Mitchell, T. R. 1987a. Image theory: Principles, goals and plans in decision making. *Acta Psychologica*.66(3), pp. 201-220.

Beach, L. R. and Mitchell, T. R. 1987b. More on image theory : A reply to Montgomery and Vlek. *Acta Psychologica*.66(3), pp. 231-235.

Beach, L. R. and Strom, E. 1989. A toadstool among the mushrooms: screening decisions and image theory's compatibility test. *Acta Psychologica*.72(1), pp. l-12.

Belaziz, M.,Bouras, A. and Brun, J. M. 2000. Morphological analysis for product design. *Computer-Aided Design*.32(5-6), pp. 377-388.

Belton, V. 1986. A comparison of the analytic hierarchy process and simple multi-attribute value function. *European journal of operational research*.26(1), pp. 7-21.

Belton, V. and Gear, T. 1983. On a short-coming in Saaty's Method of Analytic Hierarchies. *OMEGA (International Journal of Management Science)*.11(3), pp. 228-230.

Belton, V. and Goodwin, P. 1996. Remarks on the application of the analytic hierarchy process to judgmental forecasting. *International Journal of Forecasting*.12(1), pp. 155-161.

Bender, B. and Blessing, L. T. M. 2003. Re-interpretation of conceptualisation - A contribution to the advance of design theory. In: Lindemann, U. ed. *Human Behaviour in Designn*. Berlin: Springer-Verlag, pp. 10-25.

Bender, B.,Kammerer, U.,Pietzcker, F.,Blessing, L. and Hacker, W. 2001. A systematic approach to observation, analysis and categorisation of design heuristics. *Int. Conf. Engineering Design (ICED 01)*, Glasgow. August 21-23.

Bender, B.,Kammerer, U.,Pietzcker, F.,Blessing, L. and Hacker, W. 2002a. Task design and task analysis for empirical studies into design activity. *Proc. 7th Int. Design Conf.. 2002*, Dubrovnik, Croatia. pp. 119-124.

Bender, B.,Kammerer, U.,Pietzcker, F.,Hacker, W. and Blessing, L. T. M. 2002b. A software tool for observation and analysis of design activity. *Proc. 7th Int. Design Conf. 2002*, Dubrovnik, Croatia. 14-17 May, pp. 113-118

Bentley, P. J. 2000. Exploring component-based representations - The secret of creativity by evolution? *International conference on adaptive computing in design; Evolutionary design and manufacture*. Springer, pp. 161-174.

Blessing, L. T. M.,Chakrabarti, A. and Wallace, K. M. 1998. An overview of descriptive studies in relation to a general design research methodology. In: Frankenberger, E. et al. eds. *Designers - The Key to Successful Product Development*. London: Springer, pp. 42-56.

Bracewell, R. H.,Ahmed, S. and Wallace, K. M. 2004. DRED and design folders, a way of capturing, storing, and passing on, knowledge generated during design projects. *Proc. ASME 2004 Int. Design Eng. Technical Conf. & Computers and Information in Eng. Conf.(IDETC/DAC 2009)*, Salt Lake City, Utah, USA.

Brereton, M. F.,Cannon, D. M.,Mabogunje, A. and Leifer, L. J. 1996. Collaboration in design teams: How social interaction shapes the product. In: Cross, N. et al. eds. *Analysing design activity*. Chichester, England: Wiley, pp. 319-341.

Bryant, C. R.,Stone, R. B.,McAdams, D. A.,Kurtoglu, T. and Campbell, M. I. 2005. Concept generation from the functional basis of design. *Int. Conf. Eng. on Design (ICED 05)*, Melbourne, Australia, 2005

Buchanan, J.,Dillon, S. and Corner, J. 1999. A comparison of descriptive and prescriptive decision making theories. *Proc. - Decision Sciences Institute, 5th Int. Conf. : Integrating Technology and Human Decisions 1999.*, Athens, Greece. 4-7 July, 1999, pp. 1307-1309.

Buzan, T. 2005. *The ultimate book of mind maps : unlock your creativity, boost your memory, change your life*. London: Thorsons.

Cantamessa, M. 2003. An empirical perspective upon design research. *Journal of engineering design*.14(1), pp. 1-15.

Case, K.,Graham, I. and Wood, R. 2004. Shape modification using genetic algorithms. *Proceedings of the Insitution of Mechanical Engineers Part B, Journal of Engineering Manufacture*.218(7), pp. 827-832.

Cash, P.,Hicks, B. and Culley, S. 2010. An information requirement strategy for capturing and analysing design activity and behaviour. *Proc. 11th Int. Design Conf. 2010*, Dubrovnik, Croatia,17-20 May 2010.

Chakrabarti, A. and Bligh, T. P. 2001. A scheme for functional reasoning in conceptual design. *Design Studies*.22, pp. 493-517.

Chakrabarti, A.,Morgenstern, S. and Knaab, H. 2004. Identification and application of requirements and their impact on the design process: a protocol study. *Research in Engineering Design*.15(1), pp. 22-39.

Chamberlain, M. K.,Williams, C. B.,Cowan, F. S. and Mistree, F. 2001. Orchestrating learning in a graduate engineering design course. *Proc. ASME 2001 Design Eng. Technical Conf. (IDETC/DTM 2001)*, Pittsburgh, Pennysylvania, U.S.A.,September 9-12, 2001.

Chase, V. M.,Hertwig, R. and Gigerenzer, G. 1998. Visions of rationality. *Trends in cognitive sciences*.2(6), pp. 206-214.

Cohen, M. D.,March, J. G. and Olsen, J. P. 1972. A garbage can model of organizational choice. *Administrative Science Quarterly*.17(1), pp. 1-25.

Cooke, J. A.,McMahon, C. A. and North, M. R. 2003. Sources of error in the design process. In: Gogu, G. et al. eds. *Recent Advances in Integrated Design and Manufacturing in Mechanical Engineering*. Kluwer Academic Publishers, pp. 421-430.

Cosmides, L. and Toby, J. 1996. Are humans good intuitive statisticians after all? Rethinking some conclusions from the literature on judgment under uncertainty. *Cognition*.58, pp. 1-73.

Court, A. W.,Ullman, D. G. and Culley, S. J. 1998. A comparison between the provision of information to engineering designers in the UK and the USA. *International Journal of Information Management*.18(6), pp. 409-425.

Crilly, N.,Good, D.,Matravers, D. and Clarkson, P. J. 2008. Design as communication: exploring the validity and utility of relating intention to interpretation. *Design Studies*.29(5), pp. 425-457.

Cross, N. 1994. *Engineering design methods; strategies for product design*. 2nd ed. Chichester, England: John Wiley & Sons, Ltd.

Cross, N. 2000. *Engineering design methods; strategies for product design*. 3rd. ed. Chichester: Wiley.

Cross, N.,Christiaans, H. and Dorst, K. eds. 1996. *Analysing design activity.* Chichester, New York: Wiley.

Davis, J. G.,Subrahmanian, E.,Konda, S.,Grange, H.,Collins, M. and Westerberg, A. W. 2001. Creating shared information spaces to support collaborative design work. *Information Systems Frontiers*.3(3), pp. 377-392.

de Bono, E. 1977. ***Lateral thinking : a textbook of creativity***. Harmondsworth: Penguin.

Deng, Y. M. 2002. Function and behavior representation in conceptual mechanical design. *Artificial Intelligence For Engineering Design Analysis and Manufacturing (AIEDAM)*.16(5), pp. 343-362.

Dietz, T. 2003. What is a good decision? Criteria for environmental decision making. *Human Ecology Review*.10(1).

Dunegan, K. J. 1993. Framing, cognitive modes and Image Theory : Toward an understanding of a glass half full. *Journal of Applied Psychology*.78(3), pp. 491-503.

Dyer, J. S. 1990. Remarks on the analytic hierarchy process. *Management Science*.36(3), pp. 249-258.

Dym, C. L. and Little, P. 2000. *Engineering design : A project-based introduction*. John Wiley & Son, Inc.

Eckert, C.,Kelly, I. and Stacey, M. 1999. Interactive generative systems for conceptual design: An empirical perspective. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*.13(4), pp. 303-320.

Eder, W. E. 1995. Viewpoint : Engineering design- art, science and relationships. *Design Studies*.16(1), pp. 117-127.

Eder, W. E. 1998. Design modeling - a design science approach (and why does industry not use it). *Journal of engineering design*.9(4), pp. 355-371.

Eder, W. E. 2009. Why systematic design engineering? *Proc. ASME 2009 Int. Design Eng. Technical Conf. & Computers and Information in Eng. Conf. (IDETC/CIE 2009)*, San Diego, U.S.A. Pinehurst Technologies, Inc.

Edwards, W. and Barron, F. H. 1994. SMARTS and SMARTER: Improved simple methods for multiattribute utility measurement. *Organizational Behavior and Human Decision Processes*.60(3), pp. 306-325.

Edwards, W. and Hutton, B. F. 1994. SMARTS and SMARTER: Improved Simple Methods for Multiattribute Utility Measurement. *Organizational Behavior and Human Decision Processes*.60(3), pp. 306-325.

Eris, O. 2004. *Effective inquiry for innovative engineering design*. Kluwer Academic.

Evbuomwan, N. F. O.,Sivaloganathan, S. and Jebb, A. 1996. A survey of design philosophies, models, methods and systems. *Journal of Engineering Manufacture*.210(4), pp. 301-320.

Fargnoli, M.,Rovida, E. and Troisi, R. 2006. The morphological matrix: Tool for the development of innovative design solutions *ICAD 2006*, Firenze, Italy. pp. 13-16 June 2006

Fernandes, R. P.,Grosse, I. R.,Krishnamurty, S.,Witherell, P. and Wileden, J. C. 2011. Semantic methods supporting engineering design innovation. *Advanced Engineering Informatics*.25(2), pp. 185-192.

Fey, V. and Rivin, E. 2005. *Innovation on demand; New product development using TRIZ*. Cambridge: Cambridge Univ. Press.

Finger, S. and Dixon, J. R. 1989a. A review of research in mechanical engineering design. Part 1 : Descriptive, Prescriptive, and Computer-based Models of Design Processes. *Research in engineering design*.1(1), pp. 51-67.

Finger, S. and Dixon, J. R. 1989b. A review of research in mechanical engineering design. Part II : Representation, Analysis, and Design for the life cycle. *Research in engineering design*.1(2), pp. 121-137.

Finkelstein, L. and Finkelstein, A. C. W. 1983. Review of design methodology. *IEE Proceedings Part A (Physical Science, Measurement and Instrumentation, Management and Education, and Reviews)*, London. The IEE, pp. 213-222.

French, M. J. 1971. *Engineering design : The conceptual stage*. London: Heinemann Educational Books.

Fricke, G. 1996. Successful individual approaches in engineering design. *Research in engineering design*.8(3), pp. 151-165.

Galle, P. and Bela Kovacs, L. 1996. Replication protocol analysis: a method for the study of real-world design thinking. *Design Studies*.17(2), pp. 181-200.

Galle, P. and Kovács, L. B. 1992. Introspective observations of sketch design. *Design Studies*.13(3), pp. 229-272.

Girod, M.,Elliot, A. C.,Burns, N. D. and Wright, I. C. 2003. Decision making in conceptual engineering design : an empirical investigation. *Proc. of Inst. of Mech. Eng. Part B : Journal of Engineering Manufacture*.217(9), pp. 1215-1228.

Goel, A. K.,Rugaber, S. and Vattam, S. 2009. Structure, behavior, and function of complex systems: The structure, behavior, and function modeling language. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*.23(1), pp. 23-35.

Goldenberg, J. and Mazursky, D. 2002. *Creativity in product innovation*. New York: Cambridge Univ. Press.

Goldenberg, J.,Mazursky, D. and Solomon, S. 1999a. Templates of original innovation: Projecting original incremental innovations from intrinsic information. *Technological Forecasting and Social Change*.61(1), pp. 1-12.

Goldenberg, J.,Mazursky, D. and Solomon, S. 1999b. Toward Identifying the Inventive Templates of New Products: A Channeled Ideation Approach. *Journal of Marketing Research (JMR)*.36(2), pp. 200-210.

González, D. 2001. *The art of solving problems: Comparing the similarities and differences between creative problem solving (CPS), lateral thinking and synectics.* State University of New York - Buffalo State College.

Goodwin, P. and Wright, G. 2004. *Decision analysis for management judgment*. 3rd. ed. Wiley.

Gordon, W. J. J. 1961. *Synectics: the development of creative capacity*. New York: Harper & Row.

Graham, I. J.,Case, K. and Wood, R. L. 2001. Genetic algorithms in computer-aided design. *Journal of materials processing technology*.117(1-2), pp. 216-221.

Grebici, K.,Aurisicchio, M. and Bracewell, R. 2009. Guiding engineering design activities through a question based approach. *Proc. ASME 2009 Int. Design Eng. Technical Conf. & Computers and Information in Eng. Conf. (IDETC/CIE 2009)*, San Diego, U.S.A. Pinehurst Technologies, Inc.

Günther, J. and Ehrlenspiel, K. 1999. Comparing designers from practice and designers with systematic design education. *Design Studies*.20, pp. 439-451.

Hansen, C. T. and Ahmed, S. 2002. An analysis of design decision-making in industrial practice. *Proc. 7th Int. Design Conf. 2002*, Dubrovnik, Croatia.

Harker, P. T. 1989. The art and science of decision making: The analytic hierarchy process. In: Golden, B.L., Wasil, E.A., and Harker, P.T. ed. *The analytic hierarchy process: applications and studies*. Berlin: Springer-Verlag.

Harker, P. T. and Vargas, L. G. 1990. Reply to remarks on the analytic hierarchy process by J.S. Dyer. *Management Science*.36(3), pp. 260-273.

Harte, J. M.,Westenberg, M. R. R. and van Someren, M. 1994. Process models of decision making. *Acta Psychologica*.87, pp. 95-120.

Hatamura, Y. ed. 2006. *Decision-making in engineering design; Theory and practice.*Translated by Iino, K., London: Springer.

Hazelrigg, G. A. 1996. *Systems engineering : An approach to information-based design*. Upper Saddle River: Prentice Hall.

Hazelrigg, G. A. 1998. A framework for decision based engineering design. *ASME Journal of Mechanical Design*.120, pp. 653-659.

Heiser, J. and Tversky, B. 2005. Arrows in comprehending and producing mechanical diagrams. *Cognitive Science*.30(3), pp. 581-592.

Heisig, P.,Caldwell, N. H. M.,Grebici, K. and Clarkson, P. J. 2010. Exploring knowledge and information needs in engineering from the past and for the future – results from a survey. *Design Studies*.31(5), pp. 499-532.

Hicks, B. J.,Culley, S. J.,Allen, R. D. and Mullineux, G. 2002. A framework for the requirements of capturing, storing and reusing information and knowledge in engineering design. *International Journal of Information Management*.22(4), pp. 263-280.

Hideaki, T.,Yoshioka, M.,Tomiyama, T. and Shimomura, Y. 1996. Analysis of design protocol by functional evolution process model. In: Cross, N. et al. eds. *Analysing Design Activity*. Chichester: Wiley, pp. 187-294.

Hirschi, N. W. and Frey, D. D. 2002. Cognition and complexity: An experiment on the effect of coupling in parameter design. *Research in engineering design*.13(3), pp. 123-131.

Hirtz, J.,Stone, R. B.,Daniel A. McAdams,Szykman, S. and Wood, K. L. 2002. A functional basis for engineering design: Reconciling and evolving previous efforts. *Research in Engineering Design*.13(2), pp. 65-82.

Ho, C.-H. 2001. Some phenomena of problem decomposition strategy for design thinking: differences between novices and experts. *Design Studies*.22(1), pp. 27-45.

Horowitz, R. and Maimon, O. 1997. Creative Design Methodology and SIT Method. *Proc. ASME 1997 Design Eng. Technical Conf. (DETC/DTM 1997)*, Sacramento, California,14-17 Sept.,1997.

Horváth, I. 2000. Conceptual design : inside and outside. In: Rohatyñski, R. ed. *Engineering design in integrated product development*. 12 - 14 October 2000.

Huang, C.-C. 2004. A Multi-agent approach to collaborative design of modular products. *Concurrent Engineering: Research and Applications*.12(1).

Huang, G. Q. 2002. Web-based support for collaborative product design review. *Computers in industry*.48(1), pp. 71-88.

Huang, G. Q. and Mak, K. L. 1999. Web-based collaborative conceptual design. *Journal of Engineering Design*.10(2), pp. 183-193.

Hubka, V. and Eder, W. E. 1995. *Design science; Introduction to the needs, scope and organization of engineering design knowledge*. Springer-Verlag.

Jin, Y. and Zhou, W. 1999. Agent-based knowledge management for collaborative engineering. *Proc. ASME 1999 Design Eng. Technical Conf. (DETC/CIE 1999)*, Las Vegas, Nevada, U.S.A,12-16 Sept.,1999.

Johnson, A. L. 1991. Designing by functions. *Design Studies*.12(1).

Jones, J. C. 1970. *Design methods; seeds of human futures*. Wiley-Interscience.

Kavakli, M. and Gero, J. S. 2002. The structure of concurrent cognitive actions: a case study on novice and expert designers. *Design Studies*.23(1), pp. 25-40.

Kavakli, M. and Gero, J. S. 2003. Difference between expert and novice designers: an experimental study. In: Lindemann, U. ed. *Human Behaviour in Design*.

Keeney, R. L. 1992. *Value-focused thinking; A path to creative decisionmaking*. Harvard University Press.

Kirschman, C. F.,Fadel, G. M. and Jara–Almonte, C. C. 1996. Classifying functions for mechanical design. *Proc. 1996 ASME Design Eng. Technical Conf. & Computers in Eng. Conf. (DETC/DTM 1996)*, Irvine, California, U.S.A. August 18-22, 1996.

Klein, G. A. 1989. Recognition-primed decisions. In: Rouse, W.B. ed. *Advances in Man-Machine Systems Research*. Vol. 5. Greenwich: JAI Press, pp. 47-92.

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

Kroes, P. 2002. Design methodology and the nature of technical artefacts. *Design Studies*.23(3), pp. 287-302.

Lahti, A.,Mäntylä, M. and Ranta, M. 1996. Capturing and deploying design decisions. In: Pratt, M.J. et al. eds., *TC5/WG5.2 International Workshop on Geometric Modeling in Computer-Aided Design*, Airlie, Virginia, USA. Chapman & Hall, pp. 3-16.

Larkin, J. H. and Simon, H. A. 1987. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*.11(1), pp. 65-100.

Lee, M. D. and Dry, M. J. 2006. Decision making and confidence given uncertain advice. *Cognitive Science: A Multidisciplinary Journal*.30(6), pp. 1081-1095.

Liikkanen, L. A. and Perttula, M. 2009. Exploring problem decomposition in conceptual design among novice designers. *Design Studies* 30(1), pp. 38-59.

Linstone, H. A. and Turoff, M. eds. 1975. *The Delphi method: techniques and applications.* London: Addison-Wesley.

Liu, H.,Gao, L. and Liu, X. 2005. Generative design in an agent based collaborative design system. In: Shen, W. et al. eds., *Computer Supported Cooperative Work in Design I: 8th International Conference, CSCWD 2004*, Xiamen, China, . 26-28 May , 2004, Springer, pp. 105-116.

Long, M. W. and Narciso, J. D. 1999. *Probabilistic design methodology for composite aircraft structures*.Springfield, Virginia: U.S. Department of Transportation, Federal Aviation Administration. DOT/FAA/AR-99/2.

Louridas, P. 1999. Design as bricolage: anthropology meets design thinking. *Design Studies*.20(6), pp. 517-535.

Lu, H.-P.,Yu, H.-P. and Lu, S. K. 2001. The effects of cognitive style and model type on DSS acceptance: An empirical study. *European Journal of Operational Research*.131, pp. 649-663.

Lu, S. C.-Y. and Cai, J. 2000. *A generic collaborative engineering design process model for conflict management*.Los Angeles, U.S.A.: Department of Aerospace and

Mechanical Engineering, University of Southern California. IMPACT WORKING PAPER.

Mann, D. 2002. *Hands-on systematic innovation* Leper, Belgium: CREAX Press.

Mann, D.,Dewulf, S.,Zlotin, B. and Zusman, A. 2003. *Matrix 2003; Updating the TRIZ contradiction matrix*. Leper, Belgium: CREAX Press.

Mann, D. L. 2009. *Matrix 2010; Re-updating the TRIZ contradiction matrix*. Devon, UK: IFR Press.

Markus, D. 2011. Relationships between TRIZ and classical design methodology. *Procedia Engineering*.9, pp. 512-527.

McAlpine, H.,Hicks, B. J. and Culley, S. J. 2009. Improving re-use of informal information through the creation of an engineering electronic logbook (EEL) - a demonstrator. *Proc. ASME 2009 Int. Design Eng. Technical Conf. & Computers and Information in Eng. Conf. (IDETC/CIE 2009)*, San Diego, U.S.A. Pinehurst Technologies, Inc.

McCormack, J. P.,Cagan, J. and Vogel, C. M. 2004. Speaking the Buick language: capturing , understanding, and exploring brand identity with a shape grammar. *Design studies*.25(1), pp. 1-29.

McDonnell, J. 1997. Descriptive models for interpreting design. *Design Studies*.18(4), pp. 457-473.

Mehalik, M. M. and Schunn, C. D. 2006. What constitutes good design? A review of empirical studies of the design process. *International Journal of Engineering Education*. 22(3).

Montgomery, H. 1987. Image theory and dominance search theory: How is decision making actually done? : A comment on `Image theory: principles, goals, and plans in decision making' by Beach and Mitchell. *Acta Psychologica*.66(3), pp. 221-224.

Murray, T. J. 1979. Delphi methodologies: A review and critique. *Urban Systems*.4(2), pp. 153-158.

Nakagawa, T.,Kosha, H. and Mihara, Y. 2002. Reorganizing TRIZ solution generation methods into simple five in USIT. *The European TRIZ Association (ETRIA) World Conference: TRIZ Future 2002*, Strasbourg, France,6-8 Nov., 2002.

Nakakoji, K.,Yamamoto, Y. and Ohira, M. 1999. A framework that supports collective creativity in design using visual image. *Creativity & Cognition*, Loughborough, UK. ACM.

Noel, F. and Brissaud, D. 2003. Dynamic data sharing in a collaborative design environment. *Int. Journal Comp. Integrated Manufacturing*.16(7-8), pp. 546-556.

Nolan, V. 2003. Whatever happened to synectics? *Creativity and innovation management*.12(1), pp. 24-27.

NRC 1991. *"Improving engineering design: Designing for competitive advantage*. In: National Research Council Report. Washington, DC: National Academy Press.

Olewnik, A. T. and Lewis, K. 2005. On validating engineering design decision support tools. *Concurrent Engineeing: Research and Applications*.13(2), pp. 111-122.

Ong, S. K.,Xu, Q. L. and Nee, A. Y. C. 2008. *Design reuse in product development modeling, analysis and optimization*. Singapore: World Scientific Publishing.

Opricovic, S. and Tzeng, G.-H. 2004. Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS. *European Journal of Operational Research*.156(2), pp. 445–455.

Orloff, M. A. 2006. *Inventive thinking through TRIZ; A practical guide*. Berlin: Springer.

Oxman, R. 1995. Observing the observers: research issues in analysing design activity. *Design Studies*.16(2), pp. 275-283.

Pahl, G. and Beitz, W. 1995. *Engineering design; A systematic approach*. 2nd. ed. London: Springer-Verlag.

Parent, A. 1997. Analysing design-oriented dialogues: a case study in conceptual data modelling. *Design Studies*.18(1), pp. 43-66.

Pedersen, K.,Emblemsvåg, J.,Bailey, R.,Allen, J. K. and Mistree, F. 2000. Validating design methods and research: The validation square. *Proc. ASME 2000 Design Eng. Technical Conf. (DETC 2000)*, Baltimore, Maryland. 10-14 Sept.

Pedgley, O. 2007. Capturing and analysing own design activity. *Design Studies*.28(5), pp. 463-483.

Pham, D. T.,Ang, M. C.,Ng, K. W.,Otri, S. and Darwish, A. H. 2008. Generating branded product concepts: Comparing the Bees Algorithm and an evolutionary algorithm. *The Proc. 4th Int. Virtual Conf. on Innovative Production Machines and Systems (I*PROMS 2008).* Whittles Publishing, pp. 398-403.

Pham, D. T.,Dimov, S. and Ng, K. W. 2007. A descriptive design tool to capture a designer's thought. *Proc. of the 4th Int. Conf. on Digital Enterprise Technology (DET 2007)*, Bath, United Kingdom,19-21 September.

Pham, D. T. and Liu, H. 2006. A new TRIZ-based approach to design concept generation. *Int. Conf. on Manufacturing Research*, Liverpool John Moores University, Liverpool,5-7 Sept. 2006.

Pham, D. T. and Liu, H. 2009. An I-CHING-TRIZ-inspired approach to design concept generation. *Proc. ASME 2009 Int. Design Eng. Technical Conf. & Computers and Information in Eng. Conf. (IDETC/CIE 2009)*, San Diego, U.S.A. Pinehurst Technologies, Inc.

Phillips, L. D. 1984. A theory of requisite decision models. *Acta Psychologica*.56(1-3), pp. 29-48.

Pugh, S. 1991. *Total design; Integrated methods for successful product engineering*. Prentice Hall, Inc.

Quinn, J. B. 1980. *Strategies for change : logical incrementalism*. New York, McGraw-Hill: Irwin.

Rantanen, K. and Domb, E. 2008. *Simplified TRIZ*. New York: Auerbach Pub.

Rawlinson, J. G. 1981. *Introduction to creative thinking and brainstorming*. Rev. & updated ed. London: British Institute of Management Foundation.

Regli, W. C.,Hu, X.,Atwood, M. and Sun, W. 2000. A survey of design rationale systems: Approaches, representation, capture and retrieval. *Engineering with Computers*.16, pp. 209-235.

Reymen, I.,Hammer, D.,Kroes, P.,van Aken, J.,Dorst, C.,Bax, M. and Basten, T. 2006. A domain-independent descriptive design model and its application to structured reflection on design processes. *Research in Engineering Design*.16(4), pp. 147-173.

Reymen, I. M. M. J. and Hammer, D. K. 2000. Design method supporting regular reflection on design situations. In: Horváth, I. and Vergeest, J. eds., *Proc. 3rd. Int.*

Symposium on Tools and Methods of Competitive Engineering, Delft, Netherlands. April 18-21, 2000, Delft University Press, pp. 325-338.

Reymen, I. M. M. J. and Hammer, D. K. 2002. Structured reflection for improving design processes. In: Marjanovic ed. *Proc. 7th Int. Design Conf. 2002*, Dubrovnik, Croatia. May 14 - 17, 2002, pp. 887-892.

Reymen, I. M. M. J. and Melby, E. 2001. *Improving design processes through structured reflection: A protoype software tool*.Eindhoven: Stan Ackermans Institute, Center for Technological Design, Technische Universiteit Eindhoven. SAI Report 2001/2.

Roozenburg, N. F. M. and Eekels, J. 1995. *Product design : Fundamentals and methods*. John Wiley & Sons, Inc.

Rosenbaum, J. 2001. Practical creativity: Lateral thinking techniques applied to television production education. *International Journal of Engineering Education*.17(1), pp. 17-23.

Rosenthal, S. and Finger, S. 2006. Design collaboration in a distributed environment. *Proc. 36th ASEE/IEEE Frontiers in Education Conference (Session M2G-13)*, San Diego, California, U.S.A. 28 – 31 October, 2006.

Roy, U. and Kodkani, S. S. 2000. Collaborative product conceptualization tool using web technology. *Computers in industry*.41, pp. 195-209.

Saaty, T. L. 1994. Highlights and critical points in the theory and application of the analytic hierarchy process. *European Journal of Operational Research*.74(3), pp. 426-447.

Saaty, T. L. 2008. Decision making with the analytic hierarchy process. *International Journal Services Sciences*.1(1), pp. 83-98.

Savage, L. J. 1954. *The Foundations of Statistics*. New York: Wiley.

Savransky, S. D. 2001. *Engineering of creativity*. Boca Raton, U.S.A: CRC Press.

Schön, D. A. 1983. *The reflective practitioner: How professionals think in action*. New York, U.S.A.: Basic Books.

Schütze, M.,Sachse, P. and Römer, A. 2003. Support value of sketching in the design process. *Research in Engineering Design*.14(2), pp. 89-97.

Schwenk, C. R. 1984. Cognitive simplification processes in strategic decision-making. *Strategic Management Journal*.5(2), pp. 111-128.

Scott, M. J. and Antonsson, E. K. 1996. Formalism for negotiation in engineering design. *Proc. 1996 ASME Design Eng. Technical Conf. & Computers in Eng. Conf. (DETC/DTM 1996)*, Irvine, California, U.S.A.,18-22 August 1996.

See, T.-K. and Lewis, K. 2006. A formal approach to handling conflicts in multiattribute group decision making. *Journal of Mechanical Design*.128(4).

Seidl, C. and Traub, S. 1998. A new test of image theory. *Organizational Behavior and Human Decision Processes*.75(2), pp. 93-116.

Shafir, E.,Simonson, I. and Tversky, A. 1993. Reason-based choice. *Cognition*.49(1-2), pp. 11-36.

Shah, J. J.,Lacroix, Z.,Summers, J. D.,Vargas-Hernández, N. and Zhao, Z. 2001. Comparative study of representation structures for modeling function and behavior of mechanical devices. *Proc. ASME 2001 Design Eng. Technical Conf. & Computers and Information in Eng. Conf. (DETC/CIE 2001)*, Pittsburgh, Pennsylvania, U.S.A.,9-12 September 2001.

Shea, K.,Cagan, J. and Fenves, S. J. 1997. A shape annealing approach to optimal truss design with dynamic grouping of members.119(3), pp. 388-394.

Siddall, J. N. 1972. *Analytical decision-making In engineering design*. Englewood Cliffs: Prentice-Hall.

Simoff, S. J. and Maher, M. L. 2000. Analysing participation in collaborative design environments. *Design Studies*.21(2), pp. 119-144.

Simon, H. A. 1982. *Models of bounded rationality. Vol. 1, Economic analysis and public policy*. Cambridge, Massachusetts: MIT Press.

Sivaloganathan, S.,Shahin, T. M. M.,Cross, M. and Lawrence, M. 2000. A hybrid systematic and conventional approach for the design and development of a product: a case study. *Design Studies*.21(1), pp. 59-74.

Slocum, N. 2005. Participatory methods toolkit; a practitioner's manual; Method: Delphi. In: Foundation, S.S.v.a.H.L.K.B. ed. King Baudouin Foundation and the Flemish Institute for Science and Technology Assessment (viWTA).

Stauffer, L. A. and Ullman, D. G. 1988. A comparison of the results of empirical studies into the mechanical design process. *Design Studies*.9(2), pp. 107-114.

Stauffer, L. A. and Ullman, D. G. 1991. Fundamental Processes of Mechanical Designers Based on Empirical Data. *Journal of engineering design*.2(2), pp. 113-125.

Stempfle, J. and Badke-Schaub, P. 2002. Thinking in design teams - an analysis of team communication. *Design Studies*.23, pp. 473-496.

Stone, R. B. and Wood, K. L. 2000. Development of a functional basis for design. *Journal of Mechanical Design*.122, pp. 359-370.

Suh, N. P. 1990. *The principles of design*. New York: Oxford University Press.

Suh, N. P. 2001. *Axiomatic design; Advances and applications*. New York: Oxford University Press.

Szykman, S.,Racz, J. W. and Sriram, R. D. 1999. The representation of function in computer-based design. *Proc. 1999 ASME Design Eng. Technical Conf. (DETC/DTM 1999)*, Las Vegas, Nevada, U.S.A. 12-15 September , 1999, pp. 233-246.

Thomas, J.,Sen, C.,Mocko, G. M.,Summers, J. D. and Fadel, G. M. 2009. Investigation of the interpretability of three function structure representations: A user study. *Proc. ASME 2009 Int. Design Eng. Technical Conf. & Computers and Information in Eng. Conf. (IDETC/CIE 2009)*, San Diego, U.S.A. Pinehurst Technologies, Inc.

Thompson, S. C. and Paredis, C. J. J. 2009. A process-centric problem formulation for decision-based design. *Proc. ASME 2009 Int. Design Eng. Technical Conf. & Computers and Information in Eng. Conf. (IDETC/CIE 2009)*, San Diego, U.S.A. Pinehurst Technologies, Inc.

Thurston, D. L. 2001. Real and misconceived limitation to decision based design with utility analysis. *Journal of Mechanical Design*.123(2), pp. 176-182.

Thurston, D. L. L., A. Ragavan, R. 1993. Multiattribute decision methods for engineering product design. *Proc. NSF design and manufacturing systems conference*. pp. 697-702.

Tomiyama, T.,Gu, P.,Jin, Y.,Lutters, D.,Kind, C. and Kimura, F. 2009. Design methodologies: Industrial and educational applications. *CIRP Annals - Manufacturing Technology 2009*.58(2), pp. 543-565.

Tversky, A. 1972. Choice by elimination. *Journal of Mathematical Psychology*. 9(4), pp. 341-367.

Ullman, D. G. 1995. A taxonomy for classifying engineering design problems and support systems. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AI EDAM)*.9(5), pp. pg. 427-438.

Ullman, D. G. 1997. *The mechanical design process*. 2nd ed. Singapore: McGraw-Hill.

Ullman, D. G.,Herling, D. and Sinton, A. 1996. Analysis of protocol data to identify product information evolution and decision making process. In: Cross, N. et al. eds. *Analysing Design Activity*. John Wiley & Sons Inc. pp. 169-185

Ulrich, K. T. and Eppinger, S. D. 2000. *Product design and development*. 2nd. ed. Boston: Irwin/Mc Graw-Hill.

Urban, S. D.,Ayyaswamy, K.,Fu, L.,Shah, J. J. and Liang, J. 1999. Integrated product data environment: data sharing across diverse engineering applications. *Int. Journal Computer Integrated Manufacturing*.12(6), pp. 525-540.

Valkenburg, R. and Dorst, K. 1998. The reflective practice of design teams. *Design Studies*.19(3), pp. 249-271.

Van Wie, M.,Bryant, C. R.,Bohm, M. R.,Daniel A. McAdams and Stone, R. B. 2005. A model of function-based representations. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*.19, pp. 89-111.

Vlek, C. 1984. What constitutes 'a good decision'? A Panel Discussion among Ward Edwards, Istvh Kiss, Giandomenico Majone and Masanao Toda. *Acta Psychologica*.56(1-3), pp. 5-27.

Vlek, C. 1987. Towards a dynamic structural theory of decision behavior? A comment on ‘Image Theory: principles, goals, and plans in decision making’ by Beach and Mitchell. *Acta Psychologica*.66(3), pp. 225-230.

Von der Weth, R. 1999. Design instinct? - the development of individual strategies. *Design Studies*.20, pp. 453-463.

Von Der Weth, R. and Frankenberger, E. 1995. Strategies, competence and style--Problem solving in engineering design. *Learning and Instruction*.5(4), pp. 357-383.

von Neumann, J. and Morgenstern, O. 1953. *The theory of games and economic behavior*. 3rd. ed. Princeton, New Jersey: Princeton University Press.

Wright, G. and Goodwin, P. 1999. Rethinking value elicitation for personal consequential decisions. *Journal of Multi-Criteria Decision Analysis*.8(1), pp. 3-10.

Yang, M. C. 2003. Concept generation and sketching: correlations and design outcome. *Proc. ASME 2003 Design Eng. Technical Conf. & Computers and Information in Eng. Conf. (DETC/DTM 2003)*, San Diego, U.S.A. Pinehurst Technologies, Inc.