

Modelling Learning in Design

by

SIM SIANG KOK

A thesis submitted to the University of Strathclyde

for the degree of

Doctor of Philosophy

CAD Centre

Department of Design, Manufacture and Engineering Management

University of Strathclyde

Glasgow, Scotland, UK.

July 2000

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by the University of Strathclyde Regulation 3.49. Due acknowledgement must be made of the use of any material contained in, or derived from, this thesis.

Abstract

Learning in design is a phenomenon that exists in the domain of design. That the phenomenon exists is evident from the many systems built based on Artificial Intelligence and machine learning techniques in an attempt to support designers at work. However, the development of these systems may not be well founded on a holistic basis as evident from the literature review. Hence, this thesis contributes to the knowledge of modelling learning in design and has resulted in a novel and comprehensive model.

To model learning in design, formalisms for design and learning activities were derived based on the corpus of research work published. To explain the inextricable link between learning and design, three hypotheses were raised. These hypotheses posited that the cognitive activities of design and learning are linked at the epistemic level, teleological level and temporally. The formalisms and hypotheses were substantiated through a protocol analysis of a recording of a designer at work. The model based on the formalisms and hypotheses was presented for peer critique at one of the Machine Learning in Design (MLinD) workshops. Based on the outcome the protocol analysis and the peer review, the model was further evolved, resulting in the model called Model-LinD.

Model-LinD features learning in design as the transformation of input knowledge into output knowledge that is activated by rationale trigger(s) and through the interactions between the design and learning activities that are determined by the epistemic, teleological and temporal links. The evolved model was evaluated against a set of evaluation criteria, which showed that it is indeed a model capable of explaining the phenomenon of learning in design. Hence, this is claimed as the main contribution of this thesis

In order to derive a theory of learning in design from Model-LinD, the strengths and weaknesses of deriving Model-LinD have been discussed leading to some suggestions for further work.

To Rina, Calvin and Debbie

Acknowledgements

The completion of this thesis would not have been possible had it not been for many who in the course of the last five years have in many ways, some more than others but nonetheless important in contributing towards the process, the inspiration, the exhortation and the encouragement that one who embarked on a PhD journey would required. Pursuing a PhD programme on a full time basis is considered by many a daunting task; pursuing it part-time would be unimaginable had it not been for the many who encouraged me to surmount the 'unimaginable' task. To mention all of them would be too lengthy here. However, I would like to express from the bottom of my heart my thankfulness and appreciation to the following whose deeds and kindness I would not be able to repay.

First of all, I would like to thank Dr Alex H B Duffy, my PhD supervisor for his devotion, professionalism, astute guidance, counselling at times especially when the going was tough without which the journey would be trodden with a great burden, but lightened because of the manner in which he had supervised me as his student, in the most excellent way. His searching questions, critical but meticulous reading of my drafts had been instrumental in the shaping of my thoughts as I wrote the thesis.

I would like to thank Dr Kenneth MacCallum whom I met some ten years ago while he was in Singapore and through his kind gesture and hospitality enabled me to spend a period of eight months in 1994 at the CAD Centre of which he was the then Director.

I would like to thank Mr Brain Morrison, Managing Director of Vosper International Limited (VIL) in Scotland and Mr Anthony Gould, Senior Designer of VIL for the opportunity and assistance in the recording of the protocol without which the research work in this thesis would not have been possible.

I would like to thank The Dean and Head of Division of the Systems Engineering and Management Division of the School of Mechanical and Production Engineering for their encouragement during the course of undertaking the PhD programme, ever mindful that I should have a balanced workload so that my contribution towards my academic duties are adequately discharged and yet it in no way hindered my progress in the PhD work. Most of all, they showed great compassion during my period of recovery from the bypass surgery in the midst of the PhD programme.

I would like to thank Professor Robert Gay of the School of Electrical and Electronics Engineering who acted as the local supervisor, for his encouragement and the reading of my drafts.

I would like to thank my many church friends and in particular, Dr George Foong, Dr Lim Yew Cheng and Dr Goh Kee Tai for their constant encouragement especially during the period when the going was hard.

I would like to thank Mr Chong Thiam Teck who so kindly read through the draft thesis at very short notice and into the wee hours of the morning.

Last but not least, I would like to thank my dear wife, Rina and my two beloved children, Calvin and Debbie for their encouragement and their quiet assurance that ensures that the completion of this PhD. To my wife especially, who had to endure many quite evenings by herself while I was busy at completing the thesis.

To the living and true God who graciously preserved my life over two episodes of heart attack, and for His grace and wisdom that enable me to complete this thesis which serves as a testimony of his providence, protection, grace and wisdom.

TABLE OF CONTENTS

<i>Table of Contents</i>	i
<i>List of Figures</i>	v
<i>List of Tables</i>	vii

Chapter 1

Introduction	1
1.1 The role of learning in design practice.....	2
1.1.1 The role of sharing/learning of knowledge in concurrent engineering	3
1.1.2 The harness of knowledge for reuse.....	5
1.1.3 The use of Information Technology to support design and learning.....	6
1.2 The need for a theory or model of learning in design	8
1.3 Defining learning	9
1.3.1 What is learning?	10
1.3.2 Learning as storing of knowledge	12
1.4 Aims and Objectives	13
1.5 Structure of the thesis.....	14

Chapter 2

A Review of Theories and Models of Learning in Design	16
2.1 Criteria for evaluating theories, models and systems	16
2.1.1 Explanative capability	18
2.1.2 Predictability	19
2.1.3 Operationalisation	20
2.1.4 Comprehensiveness.....	21
2.1.5 Applicability	22
2.2 Evaluation of design theories	22
2.2.1 Explanative capability.....	23
2.2.2 Predictability	31
2.2.3 Operationalisation	33
2.2.4 Comprehensibility	34
2.2.5 Applicability	35
2.3 Evaluation of Models of design and learning and related systems.....	36
2.3.1 Models of design.....	36
2.3.2 Exploration-based model	39
2.3.3 Design Reuse Model	44
2.3.4 The Design/Learning Loop	47
2.4 Machine Learning in Design (MLinD) Systems	51
2.5 Summary	52

Chapter 3

Research methodology	55
3.1 A model of a world view.....	55
3.1.1 Ontology	56
3.1.2 Epistemology	56
3.1.3 Methodology	57
3.2 Research perspectives	57
3.2.1 Scientism.....	58
3.2.2 Practicism.....	58
3.2.3 Evolution of Research	59
3.3 Choice of Research methodology	61
3.3.1 Ontology of learning in design.....	62
3.3.2 Epistemology of learning in design.....	62
3.3.3 Methodology	63
3.4 Summary	65

Chapter 4

Learning in Design 67

- 4.1 The context of learning 67
 - 4.1.1 The situatedness of learning in design 68
 - 4.1.2 Learning experiential knowledge 69
 - 4.1.3 Summary 74
- 4.2 The requirements of a model of learning in design 74
 - 4.2.1 The nature of the envisaged model of learning in design 74
 - 4.2.2 Modelling of learning in design 76
 - 4.2.3 Summary 77

Chapter 5

A Formalism for Generic Design Activities 79

- 5.1 Generic design activities 80
 - 5.1.1 Granularity of design activities 80
 - 5.1.2 Identification of generic design activities 81
 - 5.1.3 Criteria for classification of design activities 82
- 5.2 A Formalism for Generic Design Activities 88
 - 5.2.1 Input Knowledge 91
 - 5.2.2 Goal of Design Activity 92
 - 5.2.3 Output knowledge 92
 - 5.2.4 Interactions of design activities 92
- 5.3 Knowledge increment in design definition activities 93
 - 5.3.1 Abstracting 93
 - 5.3.2 Association 94
 - 5.3.3 Decomposition 95
 - 5.3.4 Generating 99
 - 5.3.5 Detailing 104
 - 5.3.6 Defining 105
 - 5.3.7 Standardising 105
 - 5.3.8 Structuring/Integrating 106
 - 5.3.9 Synthesis 107
 - 5.3.10 Summary 112
- 5.4 Knowledge increment in design evaluation activities 113
 - 5.4.1 Analysis 113
 - 5.4.2 Modelling 115
 - 5.4.3 Evaluation 118
 - 5.4.4 Decision making 121
 - 5.4.5 Testing/Experimenting 124
 - 5.4.6 Summary 125
- 5.5 Knowledge increment in Design management activities 126
 - 5.5.1 Constraining 126
 - 5.5.2 Information gathering 127
 - 5.5.3 Identifying 127
 - 5.5.4 Exploring 128
 - 5.5.5 Searching 129
 - 5.5.6 Resolving 129
 - 5.5.7 Selecting 130
 - 5.5.8 Prioritising 130
 - 5.5.9 Planning 131
 - 5.5.10 Scheduling 131
 - 5.5.11 Summary 132
- 5.6 Summary 133

Chapter 6

A formalism for the learning activity 134

6.1	Defining a Foundation of Design Learning.....	135
6.1.1	Elements of a foundation for machine learning in design.....	136
6.1.2	Types of design knowledge learnt.....	137
6.1.3	The learning goal	137
6.1.4	Types of knowledge change.....	138
6.1.5	Knowledge Transformers.....	139
6.1.6	The learning triggers	140
6.1.7	Current work of Machine Learning in Design	141
6.2	Types of Design Knowledge Learnt	142
6.2.1	Product Design Knowledge.....	142
6.2.2	Design Process Knowledge.....	148
6.2.3	Summary of the types of knowledge learnt.....	151
6.3	Knowledge Transformers in MLinD Systems	152
6.3.1	Group rationalisation (or Clustering)/Decomposition (Ungroup).....	153
6.3.2	Similarity/Dissimilarity comparison	155
6.3.3	Association/Disassociation	156
6.3.4	Derivation/Randomisation	156
6.3.5	Generalisation/Specialisation.....	157
6.3.6	Abstraction/detailing.....	160
6.3.7	Explanation/Discovery.....	162
6.4	Types of Triggers for Machine Learning.....	165
6.4.1	Retrospective triggers	165
6.4.2	In-situ triggers	165
6.4.3	Provisional triggers	167
6.5	Summary	169

Chapter 7

A Model of Learning in Design..... 171

7.1	Epistemic link between design and learning activities.....	171
7.2	Temporal Link between design and learning activities.....	173
7.2.1	Retrospective learning.....	173
7.2.2	In-situ Learning.....	176
7.3.2	Provisional Learning.....	178
7.3	Teleological Link through design and learning goals	179
7.4	Summary	180

Chapter 8

Protocol analysis of the model of learning in design 181

8.1	Protocol Analysis	181
8.1.1	Model of Information Processing.....	183
8.1.2	Method of Protocol Analysis	185
8.2	Case Study	186
8.3	Segmentation and Coding	187
8.3.1	Coding Scheme for Design and Learning Activities.....	188
8.3.2	Segmentation into design activities.....	190
8.4	Results of Protocol Analysis	194
8.4.1	Identifying elements in the design activity	195
8.4.2	Identifying elements in the learning activities	199
8.4.3	Goal interaction in design and learning activities	201
8.4.4	Knowledge change in design activities	206
8.4.5	Triggers of learning.....	220
8.5	Summary of results of protocol analysis	230

Chapter 9

Evaluation of the model of learning in design 231

9.1	Evaluation through peer critique.....	231
9.2	Protocol analysis	234

9.2.1	Elements of a design activity	234
9.2.2	Elements of a learning activity	236
9.3	The evolution of the model of learning in design	239
9.3.1	Interaction between design and learning activities.....	239
9.3.2	The evolved model of learning in design.....	243
9.3.3	Backus-Naur Form of the model of learning in design	247
9.4	Evaluation of the evolved model of learning in design.....	249
9.4.1	Explanative capability.....	249
9.4.2	Predictive capability.....	252
9.4.3	Operationalisation	253
9.5	Summary	254
 Chapter 10		
Discussion and future work.....		255
10.1	Research approach	255
10.2	Results and solutions.....	258
10.3	Future work.....	259
10.3.1	Enhancing explanative and predictive capability	261
10.3.2	Model of learning in team design.....	262
10.3.3	Issues and differences between human learning and machine learning	263
 Chapter 11		
Conclusion.....		266
References		275
 Appendix 1.....A1		
Appendix 2.....A42		
Appendix 3.....A47		

LIST OF FIGURES

Figure 2.1: An exploration-based model of design.....	41
Figure 2.2: A design reuse model.....	45
Figure 2.3: Design/Learning loop.....	48
Figure 2.4: PERSPECT's activity diagram: Coupling design and learning	49
Figure 3.1: The research methodology for developing a model or theory of learning in design.....	66
Figure 4.1: The unit of experience.....	70
Figure 4.2: Mapping between tacit knowledge and explicit knowledge in design..	72
Figure 4.3: A model of learning.....	75
Figure 4.4: A model of learning in the context of design situation.....	76
Figure 5.1: Gero's model of design activities.....	89
Figure 5.2: Formalism for a design activity.....	91
Figure 5.3: Hierarchy of Design Definition Activities.....	113
Figure 5.4: Hierarchy of Design Evaluation Activities.....	126
Figure 5.5: Hierarchy of Design Management Activities for Design Solution.....	132
Figure 5.6: Hierarchy of Design Management Activities for design process....	132
Figure 6.1: Elements of learning.....	136
Figure 7.1: Retrospective Learning in Design.....	176
Figure 7.2: In-situ Learning.....	177
Figure 7.3: Provisional Learning.....	179
Figure 9.1: Original model of the learning activity but redrawn to highlight differences.....	232
Figure 9.2: Peer evolved model of the learning activity.....	233
Figure 9.3: The mapping between knowledge transformers and design activities..	241
Figure 9.4: Learning goal precedes design goal.....	242
Figure 9.5: Design goal precedes learning goal.... ..	243

Figure 9.6: Temporal learning triggers activating the design and learning activities..... 243

Figure 9.7: An evolved model of learning in design, Model-LinD..... 246

Figure 9.8: Backus-Naur Form representation of the model of learning in design.248

Figure 10.1: Research framework for future work.....261

Figure 11.1: Summary of the work..... .274

LIST OF TABLES

Table 1.1: Definitions of learning.....	11
Table 2.1: Key phases in various models of the design process.....	37
Table 2.2: Evaluation of theories.....	53
Table 2.3: Evaluation of models of design and/or learning.....	53
Table 5.1: Identification and classification of generic design activities from sources on design processes.....	85
Table 5.2: Description of design information/knowledge	86
Table 5.3 Abstracting – details of the elements of the activity and knowledge change	94
Table 5.4: Associating – details of the elements of the activity and knowledge change	95
Table 5.5: Decomposing – details of the elements of the activity and knowledge change	98
Table 5.6: Decomposing of tasks/activities – details of the elements of the activity and knowledge change.....	99
Table 5.7: Generating – details of the elements of the activity and knowledge change.	104
Table 5.8: Detailing – details of the elements of the design activity and knowledge change	104
Table 5.9: Defining – details of the elements of the design activity and knowledge change	105
Table 5.10: Standardising - details of the elements of the design activity and knowledge change.....	106
Table 5.11: Structuring - details of the elements of the design activity and knowledge change	107
Table 5.12: Synthesising- details of the elements of the design activity and knowledge change.....	112
Table 5.13: Analysis - details of the elements of the design activity and knowledge change	115
Table 5.14: Modelling- details of the elements of the design activity and knowledge change	118
Table 5.15: Evaluating - details of the elements of the design activity and knowledge change	121

Table 5.16: Decision-making - details of the elements of the design activity and knowledge change.....	124
Table 5.17: Testing/Experimenting - details of the elements of the design activity and knowledge change.....	125
Table 5.18: Constraining - details of the elements of the design activity and knowledge change.....	127
Table 5.19: Information gathering - details of the elements of the design activity and knowledge change.....	127
Table 5.20: Identifying - details of the elements of the design activity and knowledge change	128
Table 5.21: Exploring - details of the elements of the design activity and knowledge change	129
Table 5.22: Searching - details of the elements of the design activity and knowledge change	129
Table 5.23: Resolving - details of the elements of the design activity and knowledge change	130
Table 5.24: Selecting - details of the elements of the design activity and knowledge change	130
Table 5.25: Prioritising - details of the elements of the design activity and knowledge change	131
Table 5.26: Planning - details of the elements of the design activity and knowledge change	131
Table 5.27: Scheduling - details of the elements of the design activity and knowledge change	132
 Table 6.1: Elements of machine learning in design.....	 137
Table 6.2: Relationships between input knowledge, output knowledge and learning goal.....	152
Table 6.3: Knowledge transformers used in various MLinD systems and the related machine learning or other methods.....	164
Table 6.4: Learning Triggers in relation to knowledge learnt and knowledge transformer involved.....	169
 Table 8.1: Codes for Knowledge Transformers.....	 189
Table 8.2: Key words and/or criteria used in identifying design definition activity.....	191
Table 8.3: Key words and/or criteria used in identifying design evaluation activity.....	192
Table 8.4: Key words and/or criteria used in identifying design management activity.....	193

Table 8.5: Coding the segment of protocol into elements of the design and learning activity.....	198
Table 8.6: Interaction between Design and Learning Goals for Design Definition Activities.....	203
Table 8.7: Interaction between Design Goal and Learning Goal for Design Evaluation Activities.....	204
Table 8.8: Interaction between Design Goal and Learning Goal for Design Management Activities.....	205
Table 8.9: Knowledge Change and knowledge transformers for Design Definition Activities.....	209
Table 8.10: Knowledge Change and knowledge transformers for Design Evaluation Activities.....	212
Table 8.11: Knowledge Change and knowledge transformer for Design Management Activities.....	217
Table 8.12: Mapping between design definition activities and knowledge transformers.....	219
Table 8.13: Mapping between design evaluation activities and knowledge transformers.....	220
Table 8.14: Mapping between design management activities and knowledge transformers.....	220
Table 8.15: Protocol segment of knowledge learnt in retrospect and applied in design definition activities.....	221
Table 8.16: Protocol segment of knowledge learnt in retrospect and applied in design evaluation activities.....	222
Table 8.17: Protocol segment of knowledge learnt in retrospect and applied in design management activities.....	223
Table 8.18: Protocol segments and coding of examples of in-situ learning.....	225
Table 8.19: Protocol segment and coding of examples of provisional learning.....	226
Table 8.20: Protocol segment and inference of examples of rationale triggers.....	229
Table 9.1: Design definition activities identified in the protocol analysis	235
Table 9.2: Design evaluation activities identified in the protocol analysis	235
Table 9.3: Design management activities identified in the protocol analysis.....	235
Table 9.4: A comparison of the knowledge transformers postulated in the model of learning and those inferred in the protocol analysis.	238
Table 9.5: Mapping between knowledge transformers and categories of design activities inferred in the protocol analysis.	241

Table 9.6: Evaluation of the evolved model of learning in design using RD.....251

Chapter 1

Introduction

Design is a practice mediated by scientific, engineering and technological knowledge aimed at transformation of a set of needs into an artefact [1]. Science and design are closely interwoven in that the purpose of design is the creation of a product based on scientific principles. Willem [2] aptly describes this intimate relationship:

“Though the activities of design and science have little in common, science in the form of science knowledge is intimately involved in design. Science knowledge is part of the fabric with which designers design. Consider for example, the popular blow-dryer for drying hair. Principles of electricity are applied in connecting the wall outlet, switch, heating element and motor in a circuit, while the motor operation is a demonstration of electromagnetism. Fluid dynamics is involved in the fan design, and principles of convective heat-transfer guide the shaping and placement of the heating coil. Metallurgy was significant in the development of the nichrome alloy of heating coils, while polymer chemistry was involved in the development of the plastic case material. Human factor principles are applied in developing the overall geometry and case design, as are principles from art concerning aesthetics. The blow-dryer is an example of science and design inseparably joined: they coexist in the same space, science made visible through design”.

But the knowledge of science, engineering and technology continues to grow at an unprecedented pace [3] through research and development. The application of new knowledge to design practice is achieved through designers’ effort of learning and development throughout their training and professional careers. As part of their professional training, designers receive both formal education in the area of design in colleges/universities as well as “on-the-job” practical training. The learning process of designers however does not stop the moment they leave the formal system of education. Designers continue to learn as they become involved in design projects. As long as the designers continue in their profession, they will continue to gain a wealth of design knowledge and experience. Without learning, designers cannot

learn from past experience and harness new knowledge in the development of better or new products for the service of mankind. Hence, there appears to be an underlying human phenomenon that describes the cognitive process of learning of designers at work that needs to be addressed in order that the benefits of learning in the design of products can be fully utilised. In this thesis, it is suggested that the underlying phenomenon is that of learning in design.

1.1 The role of learning in design practice

Designers work in the context of market competition. Due to the sophistication of customer demand and market forces, enterprises involved in the design, manufacturing and distribution of products have to strike a right balance amongst various factors. These factors are affordability of the product pricing policy, flexibility in the variety of product range and types, and the quality of the products. Enterprises now have to achieve both economy of scale and economy of scope coupled with the promotion of quality in their products. In the phenomena of economy of scale and economy of scope there is a rising number of product units coupled with a rising number of product models. This has given rise to mass customisation and product model variety. Pine [4] suggests that mass customisation is a new way of viewing business competition, one that makes the identification and fulfillment of the wants and needs of individual customers paramount without sacrificing efficiency, effectiveness or low costs. The impact of mass customisation to product design and the design process cannot be ignored. Whitney [5] reports an extensive case study of an enterprise focusing its product design process on supporting high-volume, high variety, unpredictable demand fluctuations and high quality automated assembly. The case study shows that the enterprise was able to develop products for the high-volume, Just-In-Time (JIT) model-mix business environment by exploiting the design process and the assembly process of the products. Hence the need to satisfy the “voice of the customers” has caused enterprises to adopt customer-driven design and production resulting in an increasing number of product models or a high-variety model mix.

In an effort to remain competitive in the face of increasing product varieties, the goals in product development are shorter time to market [113, 7], high quality standards at reduced product costs [8]. Pressures to achieve these goals have resulted in significant changes in the way that engineering design is practiced [9]. Hence, enterprises have adopted different strategies for product development to achieve these goals. The main strategies are as follows:

- The sharing and learning of information/knowledge through concurrent engineering to facilitate and expediate product development.
- The harness of knowledge for reuse in current and future designs.
- The use of information technology to facilitate product development.

1.1.1 The role of sharing/learning of knowledge in concurrent engineering

The use of flexible organisational structures for product development involves matrix organisations and cross-functional project teams. The principal concept behind these flexible organisational structures is the grouping of people from several departments to work on products/projects to achieve a common goal. The notion of working cooperatively and concurrently on product development gives rise to the practice of concurrent engineering (CE) or simultaneous engineering. The traditional sequential design-and-manufacture process is being superseded by a parallel activity of concurrent engineering [8]. The success of flexible organisational structures is dependent on the sharing of information/knowledge and experiences. Neergaard [10] observes that through this process, people learn about other areas of knowledge, thereby preventing excessive specialisation and as a result, adaptability and flexibility can be expected to be facilitated.

Concurrent engineering aims at eliminating unnecessary changes [11, 12, 8] and redesigns from a product development process (PDP), and at achieving better product quality [12, 13]. To achieve CE, it is critical to facilitate teamwork and collaboration by improving communication among various participants in the development process and by allowing them to share information and knowledge [12]. In a competitive market, adopting the practice of CE by itself may not lead to success in the market place. As argued by Siemieniuch and Sinclair [13], the

adoption of concurrent engineering as a design practice does not guarantee success in the market place since competitors can equally adopt the same practice. Many proponents of CE [12, 13] have advocated that the success of CE is dependent on the sharing of knowledge and information and experiences. For example, Siemieniuch and Sinclair point to the need of restructuring and redistribution of knowledge among the people, and between people and IT applications and the necessity for a disparate group of people (comprising of researchers, designers, engineers and supplier's representatives), with their different concerns working cooperatively as a team. The knowledge that a member in a CE team possesses includes knowledge of design (i.e. past and current designs, and successful and unsuccessful designs), knowledge of how, individually to use particular design methods and techniques and knowledge of how to use these techniques effectively within the context of the organisation's design operations and procedures [13]. For the restructuring and redistribution of knowledge to be efficient and effective, the taxonomy of knowledge must be kept relevant and up to date, since concurrent engineering as a design practice occurs in an environment where the knowledge, the people and the applications are subject to change. Furthermore, difficulties in CE are encountered when large multi-disciplinary projects are involved where the team members work at different levels of detail, using different models and different knowledge [11]. There is therefore a need for computational support for concurrent engineering by developing tools that help team members to share knowledge, and to keep track of each others' needs, constraints, decisions and assumptions [14]. Recognising this need has led to a specification for a knowledge-based support in a concurrent engineering environment [11] and the identification of Artificial Intelligence (AI) approaches to model and support concurrent design of complex artefacts [15]. Douglas and Brown [16] identified twelve issues in developing tools to support CE. Three of these issues that point to some aspects of learning in design are:

- A catalogue contains descriptions of parts that can be used in the design. These parts can be components or partial designs. They are the means of relating past experience into the current design, and speeding the design process.

- An accumulated database contains knowledge about downstream activities accumulated along with the design.
- A design history captures knowledge about alternatives and decisions made, and the rationale behind those decisions. This is useful in determining if the best design was chosen, for credit/blame assignment, for learning and for use in restoring the design if a design decision has been retracted.

These three issues emphasise the need of knowledge accumulation in CE before they can be applied in a design or downstream activities. Knowledge is accumulated during the design process about all relevant aspects of the life-cycle of the product. This view of CE as knowledge accumulation can be seen to underlie all CE activities. The knowledge accumulation view of CE produces an emphasis on learning as much as possible about all aspects during the design and not just focusing on the design. Hence this view that is promoted by Douglas and Brown describes the phenomenon of learning of not just a single designer or agent but that of a team of designers. This view emphasises the role of knowledge in the design of systems for supporting CE and a clear understanding of the phenomenon will support an effective and efficient design of systems to support CE.

1.1.2 The harness of knowledge for reuse

Another strategy that enterprises can and do pursue to reduce the product development cycle(s) is to utilise past design knowledge (in the form of design descriptive knowledge/information and design process knowledge). Reusing an existing design (i.e. design reuse) is also seen as an obvious way of reducing effort and risk (and therefore time to market and cost) – not just in the design activity, but also in downstream functions. By applying an existing design, resources consumed in original design and analysis could be avoided [17]. Design reuse can only be possible if there is a conscious effort to learn from past designs and their corresponding design processes. That is, there ought to be the learning of knowledge before that knowledge can be reused or transferred to another design. Busby [18] considers design transfer as an important, practical outcome in a design activity. Without design reuse or transfer, one could not progress from elementary to complex

designs. Without design transfer, most designs would be excessively risky if they were start from scratch and would cost too much. Hence there are benefits of design reuse [17] relating to time, cost, quality and performance [19].

1.1.3 The use of Information Technology to support design and learning

The increase in the use of computers and software to support design and drawing office activities has also helped in reducing the time to market. Computer-Aided Design (CAD) systems have been developed to support design representation and modelling (e.g. wire frame modeling, boundary representation, solid modeling), Computer Aided Engineering (CAE) software for simulation and design analysis (e.g. finite element analysis (FEM), computational fluid dynamics (CFD)) [20], and Computer Aided Drafting tools for drawing office activities. Software tools to promote Computer Supported Cooperative Work (CSCW) in organisation(s) have emerged in recent years [21, 22].

In an empirical study to identify product development and design practices, policies and tools of 31 companies (Navy contractors), Stevenson and Dooley [23] found the use of computerised design tools¹ and design via appropriate physical sciences² as the two most prevalent “best” practices. This finding does not appear surprising as the use of these tools can be used to reduce design cycle time [23] and hence the time to market. However, Stevenson and Dooley concluded that there was a lack of best practices reported at the stages of new product strategy, design organization, idea generation and screening, preliminary assessment and conceptualisations as the reported practices tended to primarily address the later stages of “detailed” design. While this may indicate a lack of emphasis in the initial stages of product development process (PDP) [23], the underlying reason may well be that these design tools lack the required capabilities of an Intelligent Design Assistant (IDA) system to act as an active partner of a designer in the initial stages of PDP [24]. As an active partner to the designer, Duffy and Duffy [24] posit that the IDA system

¹ These computerised design tools include those that fall under the acronyms of CAD, CAM, CAE, AI etc.

² These include thermal analysis, finite element analysis, aero-science analysis, graphical structural analysis, fluid dynamics analysis and digital circuit analysis.

should be capable of jointly supporting design and learning assistance. Whether, the IDA is supporting a designer working individually or as a team member in CE, the dual roles as a design assistant and learning assistant will lead to a reduction in design cycle time and the time to market.

MacCallum [25] in providing answers to the question he raised ‘Does Intelligent CAD exist?’ identified three major components of an intelligent system that will deliver the type of behaviour that will promote interactive design in an Intelligent Design Assistant (IDA) system [26]. The three components are the model of the artifact in the system, the resource of knowledge that promote implicit inferencing by the system in the course of design dialogue with the user and the ability of the system to learn.

As a result, there has been numerous attempts at building Artificial Intelligence (AI) systems of machine learning in design of which case-based design (i.e. learning cases of past designs for reuse as a specific example) [27]. This growing interest has led to several Machine Learning in Design (MLinD) workshops which have resulted in special issues of the Journal of Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM) [28, 29, 30]. The research work recorded therein centres on the application of machine learning techniques to capture knowledge learnt of the design or the design process. These developed MLinD systems were able to solve some of the knowledge acquisition problems associated with building knowledge-based systems to automate or support the design process. These have been developed not only to record design history, design decisions made and states of design [31], but they are able to utilise past design knowledge, modify past design knowledge and acquire/generate new design knowledge [24]. Most of these systems embody some machine learning algorithms and techniques as part of their automatic learning of design knowledge. Because these systems are developed without a clear understanding of the phenomena of learning in design they are limited in their effectiveness as design support systems. These systems have been developed on *an ad hoc* basis without a formalism as a foundation [32] (i.e. a model or theory). Hence Sim and Duffy [32] advocate the need to put the study of MLinD

research and the development of MLinD systems to support designers on a formal basis.

1.2 The need for a theory or model of learning in design

For the strategies discussed above to be efficient and effective in design and product development, the impact of the role of learning in the organisations that practise concurrent engineering, in design reuse and IT in particular, Machine Learning in Design cannot be ignored. Zuboff [33, p395], in her book *In the Age of Smart Machine* explains that information technology has altered basic assumptions about the relationship between work and learning:

“Learning is no longer a separate activity that occurs either before one enters the workplace or in remote classroom settings. Nor is it an activity preserved for a managerial group. The behaviour that defines learning and the behaviour that defines being productive are one and the same. Learning is not something that requires time out from being engaged in productive activity; learning is the heart of productive activity. To put it simply, learning is the new form of labour.”

The need for design reuse as a research topic was first recognised in the report to the design research community in 1986 entitled, “Goals and priorities for research in engineering design” [34]. Three of the major themes on the generic research needs identified were:

- To gain a basic understanding of the design process, so as to develop a theory in design.
- The field of design theory and methodology needs to address the early, conceptual stages of design.
- The development of methodologies to reuse past design efforts (e.g. results of past decisions) on current design.

While the need to develop a design theory and methodology for design reuse was identified, there was no identification of the important role that learning plays in the design process and hence the need for developing a theory of learning in design was

not raised in the report. The focus of the design research community has been to develop theories of design. This may explain why the review in Chapter 2 of this thesis shows that while there is no lack of theories and models of design, there is no theory or model of learning in design.

It was Persidis and Duffy [35] who first posited the hypothesis that learning is inextricably linked to design. This inextricable link between learning and design points to a phenomenon that exists in and outwith a design process, a phenomenon in which the designer learns design knowledge as he/she interacts with the evolving design and interacts with other members in the design team. The hypothesis raised in this thesis is that there is currently no model or theory of learning in design; this is the reason why systems to support designers at work are developed on an *ad hoc* basis. Further, while there are theories on organisational learning [10, 36], there is no theory of learning in team design. Hence, there is a need to study the phenomenon of learning in design in a design practice (i.e. design reality) with the objective to develop a model or theory that can describe the phenomenon. Consequently this is the research task presented in this thesis; that is, the development of a model or theory of learning in design. It is envisaged that a model or theory will provide a foundation on which to develop an IDA or design support systems.

1.3 Defining learning

Learning is such a pervasive human activity that it has captivated the interest of philosophers, psychologists, scientists, teachers, students and many others. The fundamental question of what is learning continues to be the ground of many research programs. Several researchers have asked epistemological questions on learning in the areas of pedagogy [37], learning in design [35, 112, 39] and machine learning in design [40]. These questions are:

- What is learned?
- How does learning occur?
- What is the object (goal) of learning?
- When does learning occur?

The issues related to these questions will be discussed in Chapter 6 in the context of deriving a formalism for learning in design. However, because learning has been researched by different group of researchers, they may have different definitions of what constitutes learning. It is therefore important, in the context of this thesis to examine and define what is meant by learning in relation to learning in design and therefore the first of these questions is discussed here.

1.3.1 What is learning?

Learning and memory have been pursued as two separate areas of study by researchers in psychology [37]. Learning has been associated with behaviourist approaches to psychology while memory has been associated with cognitive approaches to psychology. But learning and memory are inextricably linked as learning seen as a result of a process must be stored in a depository called memory. Here learning is considered as the gain of knowledge that must be recorded in memory.

Learning as gaining knowledge

Learning may connote different meanings to different groups of researchers. Depending on the perspective of their background disciplines each group of researchers may hold certain definitions of what learning is. As a result it is often difficult to define learning [41] and hence have a consensus of what constitutes learning. Nevertheless, several definitions are given in Table 1.1 below. The purpose of tabulating these definitions is to summarise succinctly the epistemological understanding that different groups of researchers have of learning. The researchers in psychology view learning as a relatively permanent change in behaviour as a result of experience. Most learning theorists look upon learning as a process that mediates behaviour [41]. For them, learning is something that occurs as the result of certain experiences and precedes changes in behaviour. The researchers in artificial intelligence and, in particular, machine learning view learning of a computer program as improvement in performance of a task. But none of these definitions view learning as the gaining of knowledge of (a subject matter), especially through experience or through being taught.

From the pedagogic viewpoint, learning can be seen as the gaining of knowledge on the part of the learner (i.e. the student). Marton and Booth [42] view learning as progressing from acquisition of factual knowledge and integrating that knowledge into existing pre-stored knowledge. To quote Marton and Booth:

“Learning – in the sense of gaining knowledge about the world – is frequently seen as a progression that starts with acquiring some basic facts (where facts are seen as pieces of valid, elementary knowledge) and goes on through building ever more complex and advanced forms of knowledge out of, or on the ground of simpler forms. In our view, learning proceeds, as a rule, from an undifferentiated and poorly integrated understanding of the whole to an increased differentiation and integration of the world and its parts. Thus learning does not proceed as much from parts to wholes as from whole to parts, and from wholes to wholes. To put it simply, in order to learn about something you have to have some idea of what it is you are learning about.”

Table 1.1: Definitions of learning

Author	Definition of Learning	Discipline
Kimble (1961)	As a relatively permanent change in behavioural potentiality that occurs as a result of reinforced practice [43].	Psychology
Klein (1987)	As a relatively permanent change in the ability to exhibit a behaviour; this change occurs as the result of successful and unsuccessful experience [44].	Psychology
Anderson (1995)	The process by which relatively permanent changes occur in behavioural potential as a result of experience [37].	Psychology
Simon (1983)	Any change in a system that allows it to perform better the second time on repetition of the same task or that allows it to perform tasks that it could not address before [45].	Machine learning
Mitchell (1997)	A computer program is said to learn from experience with respect to some classes of tasks and performance measure, if its performance at these tasks, as measured by performance, improves with experience [46].	Artificial Intelligence

Kemp [3] views learning as that part of human behaviour concerned with the acquisition of knowledge (or concepts). To Persidis and Duffy [35], learning is the changing of the human’s state of knowledge that can have a direct influence on the ability of human to solve problems. So the concept of learning by Marton and Booth, Kemp and Persidis and Duffy suggests that learning involves some form of acquisition of new knowledge that is integrated into the existing pre-stored knowledge called memory.

1.3.2 Learning as storing of knowledge

Anderson [37] defines memory as follows:

“Memory is the relatively permanent record of experience that underlies learning.”

The use of the term “record” implies that there is some mental change that embodies the learning experience. This mental change is the creation of a memory record.

There has been growing evidence for two types of storage in human memory, short-term memory and long-term memory [37]. Short-term (ST) memory can be considered as a temporary storage system that can hold a small amount of information. Long-term (LT) memory can be considered as a more permanent repository of knowledge. This structure of memory that consists of ST and LT memories forms the basis of a theory of human memory by Atkinson and Shiffrin [47]. This model however is superseded by more current theories such as SAM by Gillund and Shiffrin [48].

In recent years, the physiological basis of learning at the neuron level is an active area of research that will shed more light on how the brain learns and records what is learned in memory [49, 50]. There have been some experimental studies of the brain activity of novices and experts to whom were given the same set of design problems to solve [51]. Based on the results of neurophysiological research, some basic assumptions were made [52]. According to these assumptions, activity in the frontal regions of the brain is a sign of planning, reasoning and decision-making. The left hemisphere specialises in the processing of abstract concepts and language, and contains our verbal memory. The right hemisphere is specialised in processing visual information and contains our visual memory. The study by Goker [51] was able to substantiate the observations made during the behavioural experiments, that with growing experience (i.e. comparing the brain activity of novices and experts), the usage of visual information replaces abstract functional models. But for the purpose of understanding learning as a gain of knowledge and how this knowledge is stored, it suffices to rest on Schank’s theory of dynamic memory. Schank [53] believed that knowledge structures in the memory change when new experiences are encountered or when a present structure is no longer applicable.

This statement presupposes that there is a depository for knowledge and that there are mechanisms by which the current state of knowledge can be compared/manipulated/changed to the new state. Without memory, information/knowledge learned cannot be compared, stored and retrieved for subsequent use. Kerr [54] therefore concluded that memory can be considered to be the foundation upon which learning can take place.

1.4 Aims and Objectives

The overall aim of this research is to develop a model of learning in design that will serve as a basis to formulate a theory of learning in design. To achieve this aim the following objectives were identified:

- There exists the phenomenon of learning in design as a cognitive process in design practice and hence raise a hypothesis that there is no model or theory to explain the phenomenon.
- To show that the hypothesis is indeed true that there is no model or theory of learning in design and hence identify the need for a model or theory of learning in design.
- To adopt an appropriate research methodology by which a model of learning in design can be derived and developed.
- To describe what is learning in design and hence establish the requirements of a model of learning in design and the approaches that are required to model learning in design.
- To identify and classify the generic design activities and represent these activities using a formalism so that knowledge change in each of these design activities can be identified. The knowledge change is seen as contributing to the learning process.
- To identify and justify basic elements that will represent a learning activity in order to explain what is learnt, why learning occurs, how learning takes place, and when does learning occur in the context of design.

- To hypothesise on the manner in which design and learning as cognitive processes are inextricably linked and hence propose a model of learning in design.
- To conduct and analyse a protocol recording of a designer at work as one of the basis to evaluate the proposed model of learning.
- To evaluate and evolve the proposed model of learning in design based on the results of the protocol analysis and through peer critique by presenting the model to the research community of learning in design.
- To identify avenues of future work in order that a more generic model or theory of learning in design can be achieved as a basis to develop and implement design support system(s).

1.5 Structure of the thesis

This thesis consists of eleven chapters in three parts. Part 1 (Chapters 1 to 4) seeks to define the research problem as the need to develop a model or theory of learning in design and how the research problem is to be solved through an appropriate research methodology. Part 2 (Chapters 5 to 7) presents a proposed solution to the research problem in a model of learning in design. Part 3 (Chapters 8 to 11) presents the evaluation of the proposed model and the refinements made to the model as a result of peer evaluation and that which is revealed in a protocol recording and analysis.

In Chapter 1, it has been shown that there is a design reality (i.e. a phenomenon) of learning in design that must be studied and comprehended before a formalism (i.e. a model or theory) can be proposed or developed. The model or theory is seen as a foundation by which an effective and efficient IDA can be developed to support designers. In Chapter 2, a critical review of existing theories, models of design or learning and of systems implemented based on some of these models supports the hypothesis raised in Chapter 1 that there is no model or theory of learning in design and hence reaffirms the research problem of the need to develop such a model. Chapter 3 reviews research methodologies so that an appropriate methodology could

be chosen for the research problem. Chapter 4 defines what is the phenomenon of learning in design, the nature and manner in which model of learning in design is sought.

Chapter 5 reviews current literature in design research and practice in order to identify and classify generic design activities for which a formalism for a generic design activity is proposed. This formalism is used to study the knowledge change that occurs during a design activity. In Chapter 6, a similar review of current literature of research work in machine learning, machine learning in design (MLinD) and cognitive science was undertaken resulting in the definition of a formalism for the learning activity that occurs during or outwith the design activity. The elements of the formalism were substantiated through a critique of MLinD systems. By synthesising the formalisms representing the cognitive activities of design and learning, a model of learning in design was derived. Hence Chapter 7 presents the proposed model of learning in design and presents three hypotheses to show the nature of the inextricable links that may exist between the cognitive activities of design and learning.

In order to evaluate the proposed model, a protocol recording of a designer at work was analysed. Chapter 8 serves as the repository of the results of the analysis. As part of the research methodology, the model of learning in design was evaluated through a peer review and the results of the protocol analysis. This led to a refinement of the model and a representation of the model in Backus-Naur Form (BNF) which can be implemented as a computational model of learning in design for further evaluation. Hence Chapter 9 describes the evaluation and the refinement of the model of learning in design. Chapter 10 discusses the strengths and weaknesses of the research methodology in the light of the proposed solution to the research problem established in Part 1. Chapter 10 also presents some avenues by which the model of learning in design can be developed into a theory of learning in design.

Chapter 11 summarises the key contributions and conclusions; that is, a development of a model of learning in design.

Chapter 2

A Review of Theories and Models of Learning in Design

Design and learning are two distinctly different types of human cognitive activities. While designing requires the application of knowledge, learning results in the acquisition of new knowledge. According to Duffy and Duffy [24] designers learn when they encounter knowledge that is sufficiently different from their present state of knowledge. Although design and learning are two distinct cognitive activities they are inextricably linked [24, 35]. To date the design research community has developed many models and theories of design to describe and prescribe what is design and what it ought to be. In Chapter 1 the research problem that there is no model or theory to describe the phenomenon of learning in design was identified. The purpose of Chapter 2 is to present a critical review of the existing models of design and/or learning, and the theories of design to affirm or reject the identified research problem. The critical review is based on a set of criteria given in Section 2. While the review shows that there are several theories of design and many models of the design process, there is a dearth of models of learning in design, let alone theories of learning in design. This finding is important as it therefore suggests that there is a need to develop a model of learning in design that may lay the foundation for a theory of learning in design. Such a theory could then be used to develop effective IDA or design support systems. Hence the purpose of this review is important in establishing this need to develop a model of learning in design which is the thesis of this research work.

2.1 Criteria for evaluating theories, models and systems

Whilst there is no lack of theories or models of design, there are however only a few models that describe some aspects of learning in design. The purpose of the critical review of theories, models and systems related to design and learning is twofold. Firstly it is to establish if there is currently a theory or model of learning in design. Secondly if there are theories or models of learning in design, what aspects of the phenomena of learning in design that these theories or models portray and how

generic are they? Before defining what are the criteria used to evaluate these theories, models and systems, there is a need to distinguish between theories, models and systems.

Theories and models play a central role in scientific endeavours. Theories and models can be seen as different abstractions used by scientists and researchers to understand and explain natural phenomena or human behaviour phenomena. They do not represent reality completely but as an idealisation through which knowledge of the phenomena in terms of laws, hypotheses and axioms can be investigated, verified and further developed.

Theories are general statements which make no reference to and do not depend upon particular instances of the phenomenon they are supposed to be theories about [55]. According to Braithwaite [56], a scientific theory is a deductive system consisting of certain initial hypotheses at the summit and empirically testable generalisations at the base. The deductive structure of a theory is shown explicitly by expressing the theory by means of a formal axiomatic system or calculus. The formal axiomatic system consists of a sequence of sentences (or formulae) in which the initial hypotheses are represented by sentences called axioms and what are deduced from these hypotheses by sentences which are called theorems of the calculus.

Models of nature, including cognitive models, are abstract organisational ideas derived from inferences based on observations [55, 57]. One of the purposes of building models is to make observations more comprehensible. It provides researchers with a structure within which specific hypotheses can be tested, and enables them to predict events on the basis of the model. Although models do not actually constitute a theory, a theory can emerge when there are feasible explanations as to why a model behaves as it does [58]. In other words it is an acceptable scientific approach to hypothesise a model and then collect and analyse experimental data to ascertain how robust the model is and the extent of its validity. In summary, a model is based on inferences drawn from observations. Its purpose is to provide an understandable representation of the character of the observation and in developing

hypotheses, and aid in making predictions [57].

Design and learning as cognitive activities demonstrate human intelligence behaviour. To study some kind or aspect of intelligent behaviour, Artificial Intelligence (AI) researchers attempt to build experimental systems which embody their formal theories of intelligent behaviour being studied [59]. These experimental systems are computer programs. The methodology generally adopted in attempting to achieve this aim has three parts [59]:

1. The development of a theory or model of the design process.
2. The development of a software system architecture which embodies this model.
3. The implementation of this architecture in the form of computer programs which we can then experiment with and test.

Hence systems can be considered as a means to evaluate theories or models. If Smithers considers models as particularisation and specialisation of a theory [55], then systems may be considered as further particularisation and specialisation of models and theories. Hence it is proposed that not all computational systems on design and learning will be evaluated unless they are developed in order to evaluate some theory or model of learning in design.

It is proposed that a set of criteria be used to evaluate the current theories, models and systems in design and/or learning. The criteria consists of the following:

- Explanative capability,
- Predictability,
- Operationalisation,
- Comprehensiveness,
- Applicability.

The rationale for this set of criteria in evaluating theories and models are discussed below.

2.1.1 Explanative capability

In general, scientific theories or models offer an understanding and provide the basis of explanations of the phenomena we observe. They are necessarily abstract statements that identify general properties, characteristics, and underlying processes

and mechanisms for all possible instances of the phenomena [55]. Explanations can be either based on evidence or based on a set of deductions derived from the tenets of the theory or models. However explanations may turn out to be inaccurate, inadequate or irrelevant. Explanations must be at a level of abstraction higher than the data or generalization they explain. Dixon [58] draws a distinction between descriptive theories and prescriptions. A descriptive theory defines a system and explains why it behaves as it does. A prescription states how a system, or at least part of the system, should behave according to the author of the prescription. Hence, a model or theory is said to have explanative capability if it is able to explain every observable instances of the phenomenon through its systems of abstract statements. A model or theory cannot be considered as one if it does not have an explanative capability. Hence explanative capability is considered here as the criterion to evaluate whether a model or theory qualifies to be as a model or theory of a phenomenon.

2.1.2 Predictability

The concepts of prediction and predictability are central to scientific thinking and are of practical importance [60] for this is ultimately the means by which science gives an understanding of nature and other people [61]. For models or theories to exhibit predictability, they are characterised by a consistent set of statements and an inferential structure that can be used for prediction of events or conditions on the basis of assumed existing conditions [62]. Michalski [63] and Shanahan [64] show that the inference process of abduction is explanation and the inference process of deduction is prediction.

The most convincing demonstration of the validity of a scientific hypothesis is the successful prediction of a previously unobserved or unrecorded phenomenon. One example is Mendeleev's model [65]. Mendeleev's model serves to illustrate the hypothesis testing and predictive capability of the model. It classifies the structure of elements into a periodic table. On the basis of the arrangements of the elements in it, rather than perform exhaustive experiments involving chemical combinations, scientists can accurately hypothesise chemical combinations and displacement laws. Furthermore, elements not yet discovered can be predicted and their characteristics

inferred, even though no physical proof of their properties exist. But prediction is not the necessary and sufficient condition for scientific explanation [61]. There are, for example, perfectly sound ‘historical’ sciences such as geology, palaeontology and cosmology, whose phenomena cannot strictly be predicted since they all occurred in the past. However theories about these phenomena may be satisfactorily validated by the discovery of ‘predicted’ evidence after they were originally proposed.

Lighthill [66] demonstrates that for a wide class of simple systems that satisfy the equations of Newtonian dynamics and these equations can be solved precisely (with known initial conditions), predictability is impossible beyond a certain definite time horizon. Thus, it can be argued that a theory or model need not necessarily have the predictive capability as long as it has the explanative capability. Predictability is considered as an important criterion as it has been argued that it is not necessary that a model or theory has predictive capability as long as it has explanative capability.

2.1.3 Operationalisation

To construct effective explanations, all the terms and concepts used to form a theory must be operationalised [55]. These operationalisations must make it possible in practice, to identify and classify, unambiguously, particular examples and states of the phenomena covered by the theory. What are the observables of the phenomena for which the theory is trying to explain must be established first. Before operationalisation can occur, all terms and concepts must be subjected to an operational definition.

An operational definition of a term is one which defines the term in connection to some physical or computational operation used to measure or quantify the defined entity. Not all terms or concepts can be measured. Hence Bridgman [67] distinguishes between several kinds of operation that may be invoked in specifying meanings of scientific terms. The principal ones are what Bridgman calls *instrument* operations – these consist of the use of various devices of observation and measurement and what Hempel [68] calls *symbolic* operations which include paper and pencil operations, verbal operations, mental experiments, the techniques of

mathematical and logical inference and the like. To safeguard the objectivity of science, all operations invoked in this kind of definition are required to be inter-subjective in the sense that different observers must be able to perform “the same operation” with reasonable agreement in their results [68].

An operational definition is meant to give experiential meaning to the term it introduces because it enables us to decide on the applicability of that term to a given case by observing the response the case shows under specifiable test conditions.

Smithers [55] observed that the process of developing and testing theories in science is essentially the development and use of terms and concepts that can be operationalised adequately to form effective explanations. Hence an effective theory or theory of learning in design in particular must be founded on a set of operations that can give a reasonable agreement in the result.

2.1.4 Comprehensiveness

Miller [69] observes that the goal of science is the development of both special theories of limited scope and general theories that unify or integrate special theories and cover broader spheres of knowledge. It is usually necessary to start with special theories that deal with a limited set of phenomena. Middle-range theories concerned with a greater number of phenomena come later. Ultimately a body of research based upon these theories lead to general theories that include a major segment of the total subject-matter of a field or of several fields.

Designing is a particularly sophisticated kind of behaviour, drawing on numerous human cognitive capacities in complicated ways [55]. Here, in evaluating the various theories and models, it is important that the theory or model in question is able to describe the phenomena of learning in design. So a design theory will be considered as a special theory since it describes a single perspective of the cognitive behaviour of the designer. A theory or model of learning in design that describes both the phenomena of designing and learning and the coupling between them will be considered as a middle-range theory to use the classification by Miller. But one that is able to describe design as a social process involving not just one designer but

a team in which both cognitive activities of designing and learning in a corporate manner would be considered as a general theory. So, in evaluating the comprehensiveness of a theory or model describing the cognitive phenomena of design, these qualitative terms of special, middle-range or general will be used to describe its comprehensiveness.

Whilst ultimately the aim is to develop general theories, Dasgupta [70] argues that a grand theory of design is pointless if it is so general that no one knows how to relate it to specific problems. Thus he concludes that our search for a design theory must attend to both the theoretical need for generality and the practical quest for domain-specificity.

In terms of evaluating how generic a theory is, comprehensiveness as a criterion of evaluation should be ranked the highest. But unless a model or theory can satisfy the criterion of explanative capability, it will not be considered for its genericity. However, here, the measure of comprehensiveness of a model or theory is not in explaining the phenomenon of design but that of learning in design. So even if a model or theory qualifies as one in explaining the phenomenon of design, it still lacks comprehensiveness if it cannot explain the phenomenon of learning in design.

2.1.5 Applicability

The measure of applicability of a theory or model of design is the extent to which it has been applied to design of different artifacts or different types of design processes (i.e. architectural, engineering, or software). It can be described as artifact independent or domain independent.

2.2 Evaluation of design theories

Using the set of criteria given in Section 2.1, the five theories to be evaluated are:

- General Design Theory
- Axiomatic Theory of Design
- Theory of Technical Systems
- Theory of Plausible Design

- Inferential Design Theory

The theories selected here are not meant to be exhaustive but rather they are considered as theories indicative of the domains of engineering design (i.e. mechanical, machine design, system design, computer architecture design).

2.2.1 Explanative capability

General Design Theory

Models or theories that display explanative capability are characterised by a system of abstract statements that consist of axioms and theorems or some formal expressions such as inference equation. Hence, the General Design theory (GDT) and Axiomatic Theory of Design (ATD) use a system of axioms and theorems to explain the phenomenon of design.

The original GDT developed by Yoshikawa [71] uses three axioms³ to describe design as the mapping from the function space (the entity concept set) to the attribute space (the design topology) in an ideal world. That is, design is described as a mapping from the conceptual world to the logical world, ignoring the influence of physical laws that exist in the real world. In the ideal knowledge scenario, design solutions can be obtained immediately once the design specifications are completely described. The most significant result of having the ideal knowledge, which can be proven from the three axioms, is that design, as a mapping from the function space to the attribute space, immediately terminates when the specification is described. Thus there is no iteration in the design process and the knowledge change is the design specification derived from the functional specification as a one to one mapping.

To remove the restriction imposed by the definition of ideal knowledge, Tomiyama and Yoshikawa [72] extended the GDT in three additional theorems to take into account the physical laws of nature and its influence on the design solution. These

³ Axiom 1 (Axiom of recognition). *Any entity can be recognised or described by attributes and/or other abstract concepts.*

Axiom 2 (Axiom of correspondence). *The entity set S' and the set of entity concepts (ideal) S have one-to-one correspondence.*

Axiom 3 (Axiom of operation). *The set of abstract concepts is a topology of the set of entity concepts.*

theorems state that the feasible entity concept is to be selected from the topology of the physical law concepts, that every attribute has a value if it can be measured and that design solution(s) can be obtained as a convergence process given the design specification [73]. In the convergence process, two theorems describe the design solution in terms of the concept of the meta-model, which is described in terms of attributes that can be measured by physical laws. Thus in the real world, the system of three axioms and three theorems explain design as a stepwise evolutionary transformation process from the design specification to the design solution. There is an implicit knowledge increment as the design is evolved, but GDT does not explain this implicit knowledge increment as learning *per se*.

The GDT as expounded by Yoshikawa and Tomiyama is able to explain design as a mapping from a functional space to an attribute space via the meta-model (physical phenomena) space. To test the theoretical results of GDT, Takeda et al. [74] conducted an experimental investigation of design processes that led to the formulation of the cognitive design process model [75]. While GDT is able to explain what design is it fails to explain the phenomena of designing as cognitive actions taken by the designer. Recognising the inadequacy to explain the interaction between the evolving design through actions of the designer, a computable design process model was developed by Takeda et al. [76]. The model formalises design processes as combinations of three types of reasoning i.e., deduction, abduction and circumscription. Hence the inability of GDT to explain the design phenomena necessitated in the formulation of the cognitive design process model and the need to formalise it in a computable design process model. However, the extended GDT and its subsequent evolution do not address or explain the phenomena of learning in design.

Axiomatic Theory of Design

Similar in nature to the extended GDT which describes design as a mapping from the functional space to the attribute space, Suh [77] defines design and its evolution to a final solution as a mapping process between the Functional Requirements (FRs) and the Design Parameters (DPs). According to Suh, FRs are defined as a minimum set

of independent requirements that completely characterises the functional needs of the design in the functional domain whereas DPs are the key variables that characterise the physical entity to fulfill the FRs. Suh defines two axioms to state the relationships between FRs and DPs. The mapping between the FRs in the functional domain and the DPs in the physical domain may be expressed as a design equation:

$$\{\text{FRs}\} = [\text{DM}] \{\text{DPs}\}$$

where $\{\text{FRs}\}$ represents the vector of FRs, $[\text{DM}]$ is the design matrix, and $\{\text{DPs}\}$ is the vector of the design parameters.

The nature of the design matrix determines the three groups of design: uncoupled, coupled and decoupled designs. An uncoupled design satisfies Axiom 1, whereas a coupled renders some of the functions dependent on other functions, and thus violates Axiom 1. Several corollaries⁴ and many theorems [77, 78] can be proven from these two axioms. These corollaries are seen as more useful in making specific design decisions, since they can be applied to actual situations more readily than can the original axioms [77, 79]. These corollaries and theorems are statements of design process knowledge as to how FRs, DPs and the DM can be manipulated in order to arrive at the uncoupled design that satisfies Axiom 1.

Axiom 2 states that, among all the designs that satisfy the Independence Axiom (Axiom 1), the one with minimum information content is the best design. Hence Axiom 2 provides the rationale for the selection of the design.

A good design theory must describe what constitutes a good design, how a good design can be derived and explain why a selected design is an optimum design. Suh's ATD is able to provide answers to these fundamental questions that one would seek in the design. Through its system of axioms, corollaries and theorems, a designer can explain the evolution of the design solution until the final solution is derived. As far as describing design as a phenomenon, ATD has explanative

Axiom 1: The Independence Axiom - *Maintain the independence of FRs*

Axiom 2: The Information Axiom - *Minimise the information content of the design*

⁴ Corollaries are propositions that follow from axioms or other propositions that have been proven.

capability, but it fails to describe the phenomenon of learning in design.

Theory of Technical Systems

The theory of Technical Systems has been perceived by Hubka and Eder [80] as a general theory about artefacts that should cover their components and relationships, the life stages of an individual artefact, the development of artefacts over longer time periods, the general properties of such artefacts and their internal structure. Thus their theory of technical systems [81] embodies statements and propositions regarding nine aspects of the design of an artefact. They are the transformation systems, technical process, technical systems (TS)⁵, classification of TS, properties of the TS, evaluation of TS, representation of TS, origination and operations of TS and the developments in TS in the course of time. There are a total of 61 statements and 9 propositions. These statements are not axiomatic in nature from which theorems can be derived.

A transformation system is basically a system of inputs, operators to achieve the transformation expressed as outputs in terms of operands (e.g. materials, energy, information) to satisfy a fairly well-defined purpose. A technical process is a special form of transformation process which allows auxiliary, regulating and controlling, material and energy converting, connecting and supporting processes in the time domain. Ehrlenspiel [82] gives a succinct definition of what Hubka and Eder meant by a technical system as:

“A system consists of a set of elements (subsystems) that possess properties and that are connected to each other by relationships. A system is separated from its surroundings by a system boundary and relates to its environment, for instance by input and output quantities. The function of a system can be described by those differences between its input and output quantities that correspond to its purpose”.

The Theory of Technical Systems as envisaged by Hubka and Eder is to show the overall structure of knowledge about processes and systems, including the role of information, management, goal-setting and realisation and the social and physical

⁵ The term “technical system” is used to represent all types of man-made artifacts, including technical

environment. Many branches of engineering (e.g. theory of machine systems) can be subsumed under its general umbrella since it treats all basic aspects from a sufficiently high level of abstraction. It embodies general descriptive statements of what is a technical system or a design. As such it does not explain the design process. As a theory of design its explanative capability covers the design but not the design process. Its explanative capability in explaining the phenomenon of learning in design is non-existent.

Theory of Plausible Design

The Theory of Plausible Design (TPD) promoted by Dasgupta [70] seeks to explain both the design process and the artefact evolving towards the design solution. The design process is evolutionary in that the design and/or the requirements are continually modified so as to attain a state of mutual adaptation. The evolving design is a collection of tentative hypotheses such that evidence in favour of the hypotheses establishes the plausibility of the design which may be revised if necessary in the light of new evidence.

A central notion in TPD is that a design (of some target artefact) is a specification of various features (e.g. function, behaviour, structure, performance, reliability, aesthetic etc.) expressed as constraints. A design at any particular abstraction level is, then, a specification of any organised, mutually consistent collection of constraints which a future implementation must satisfy. Given a constraint, its plausibility refers to the extent to which the designer believes that an implementation of the artifact satisfies the constraint. At any stage of a TPD based design process, constraint C can be in exactly one of four plausibility states namely assumed, validated, refuted and undetermined⁶.

In the course of the design process each new constraint is introduced into the design.

products and processes.

⁶ Assumed: when there is no evidence against C's plausibility.

Validated: when there is significant evidence in *favour of* and no *evidence against* C's plausibility.

Refuted: when there is evidence against C's plausibility.

Undetermined: when it is not known what counts (or may count) as evidence for or against C's plausibility.

Thus the design process can be expressed as a plausibility statement, defined as a 5-tuple:

$$S = \{C, A, R, V, P\}$$

Where:

- C is a specific constraint the plausibility of which is being (or is to be) determined.
- A is the knowledge base used to assist in establishing C's plausibility. This includes facts, theories, principles, tools, research results, as well as other constraints belonging to the current design.
- R is a logical relationship or assertion of which implies the plausibility of C. R is specified in predicate calculus [83].
- V denotes the means of verification (that is, the nature of the evidence) used to establish C's plausibility.
- P is the plausibility state in which C is being (or is to be) placed.

The aim of the plausibility driven design process [70] is such that it seeks to change the state of constraints from the plausibility state of 'assumed' to 'validated' or from 'undetermined' to 'assumed' or 'validated state'. This may involve the refinement or partitioning of the original constraint C_j into a set of new constraints according to some laws of partitioning such that some relationship between the constraints can be used to demonstrate the desired plausibility for C_j . In the process of attempting to change the plausibility state of a constraint C_j from 'assumed' to 'validated', it may happen that the evidence E_k is such that C_j is found to be refuted. In that case, C_j , must be eliminated from the design and if, some other C_i is dependent on C_j , then C_i 's plausibility state will need to be revised in the light of the new state of affairs. Thus TPD features an evolutionary design process which interleaves between revision of the state of constraints in the light of new evidence and a goal-directed nature of constraint generation until all the constraints are implemented in the design.

One of the distinctive features of TPD is its explicit attention to the role and nature of the evidence in establishing a design's plausibility [70]. The plausibility of a design as claimed by the designer is determined by the kind of evidence invoked in

the support of the claims. The evidence in turn, involves relevant knowledge of the design domains: tools (compilers, simulators, prototypes etc.); techniques of verification (empirical analysis, approximate reasoning, experiments, formal proofs, etc.); and the causal history of the individual constraints as illuminated by the constraint dependency graph and the plausibility statement of the corresponding constraints. The plausibility equation that expresses the state of the design in terms of constraints and its state of evolution provides a basis for explaining the current state of the design and the likely state the design is to evolve to. Hence TPD has explanative capability to explain evolution of the design as well as the process. But it does not feature in its construct its ability to explain the phenomenon of learning in design.

Inferential Design Theory

Arciszewski and Michalski [84] developed the Inferential Design Theory (IDT) based on the Inferential Theory of Learning (ITL) developed by Michalski [63]. The key idea of IDT is to characterise a design process as a goal-directed search through a knowledge space. In this exploration, the designer learns about the design as he/she evolves the design concept. A collection of search operators called *transmutations* is used to explore this space. *Transmutations* or design knowledge transmutations are generic patterns of knowledge change, which transform various components of knowledge: they can generate intrinsically new knowledge (by induction or analogy); produce derived knowledge (by deduction); or change knowledge (by manipulation). A design process is thus viewed as a sequence of design knowledge transmutations that transform the designer's initial knowledge into knowledge satisfying the design goal or a system of goals. During each design cycle the designer analyses the input knowledge in terms of his/her background knowledge and goals, and performs various inferences to generate new design knowledge. When this is produced, it is stored and made available for subsequent iterations. The process terminates when the totality of knowledge satisfies the design goal.

During this exploration process, the best concept(s) is identified through a five-step concept generation procedure. The five-step procedure consists of *transmutation*

selection, attribute selection, operation, learning and conceptualisation. In transmutation selection, the designer chooses a specific transmutation for developing a design concept. Attribute selection enables the designer to focus on the design space by selecting the specific design attribute or a group of attributes. In the operation step, the system generates and displays a possible partial design solution based on the transmutation and attribute selected or additional new attribute(s) required to complete the step. In the learning step, based on the solution generated by the system, the designer learns (i.e. a cognitive association process) through relating the consequences of the operation with some original or past concept. The designer finally formulates a new concept by identifying specific attribute values as constituting a design concept or further extends the design space using new attributes or group of attributes.

All design knowledge transmutations are governed by an underlying inference process which can be expressed as the fundamental equation of inference [63] which is an entailment:

$$P \cup BK = C$$

Where:

P is a set of statements, input statements called the premise.

BK is a set of statements representing the background knowledge.

C is a set of statements called the consequent.

Deductive inference is deriving the consequent C, given P and BK. Deduction is truth-preserving, i.e., C must be true if P and BK are true. Inductive inference is hypothesising the premise P, given C and BK. Induction is falsity-preserving, i.e., if C is false, then P must be false also, if BK is true. If a premise P is the hypothesised that explains the input, then it is called abduction.

IDT exhibits explanative capability because of the underlying inference process of ITL of Michalski. Using a set of 24 design knowledge transmutations and a set of 12 design specific transmutations⁷, Karni and Arciszewski [85] were able to perform

⁷ These include retention/extension, framing/reframing, implosion/explosion, symmetry/asymmetry,

and explain concept generations of an after-sales service facility. Thus IDT is able to explain the knowledge change that occurs during the design process. But a close examination of the design specific transmutation for the design of the after-sales service facility showed that the specific transmutations (e.g. retention/extension) are not the cognitive activities of design. Hence IDT explains the learning process during the conceptual design but not in the context of a specific design activity. Basically IDT is a variant of ITL that has been applied to a specific domain of design.

2.2.2 Predictability

Theories that describe a phenomenon in terms of a system of axioms and theorems or statements that have an inference structure demonstrate the ability to predict to some degree the outcome of each design action. Thus it is observed that GDT, ADT, IDT, and TDP display the ability to predict states of a design through their corresponding inference structure.

The extended GDT, without an underlying inference structure, does not have a predictive capability in describing the design process. However, augmented with a cognitive design process model [74] and logically formalised as a computable design process model [75, 76, 86], the extended GDT is imbued with an inference structure that results in its predictive capability. At the design object level, model formalises design processes as combinations of three types of reasoning, i.e. deduction, abduction and circumscription.

The object level design processes are logically formalised as:

$$K_o \cup D_s \vdash P$$

Where:

D_s is a set of logical formulae describing the design solutions that the designer seeks.

K_o is the knowledge on object properties.

P is a set of logical formulae describing properties of the design solutions

affirmation/confirmation and conjunction/disjunction.

respectively. Required design specifications are described in P.

A design suggestion is an abduction process to obtain D_s from P and K_o . Concept development and candidate evaluation are deduction processes to obtain P from D_s and K_o for getting more information about the design solutions. Circumspection⁸ can be used for resolving contradictions by adding exceptions to the original problem.

Action level design processes are logically formalized as [86]:

$$K_a \cup C \vdash O$$

Where:

K_a is the action level design process knowledge (i.e. the relations between the conditions of the design object and the possible operations).

C is a set of the conditions of the design object.

O is a set of possible operations on the design object.

The deductive reasoning in the action level inference determines the object level operations from the conditions of the object level inference and the action level knowledge [86]. This implies that the extended GDT has predictive capability (due to its deductive reasoning) in determining the nature of the design object and what actions to take to evolve the design.

Through analysing the design equation and the nature of its matrix (i.e. the inference structure) in ADT, the designer is able to infer if the design is uncoupled, decoupled or coupled. Knowing the nature of the matrix, and guided by the corollaries and theorems the designer can proceed to take appropriate actions in manipulating the design matrix (e.g. application of Theorems 1 and 2, i.e. inclusion of additional DPs so that the DPs equal FRs in order to have decoupled the design). So, through its axioms, theorems and corollaries, ADT can predict actions that a designer can take to

⁸ Circumscription is a minimal patching technique for contradictory knowledge bases. For example $\text{bird}(X) \wedge \neg \text{ab}(X) \rightarrow \text{fly}(X)$, $\text{ostrich}(X) \rightarrow \neg \text{fly}(X)$, $\text{ostrich}(X) \rightarrow \text{bird}(X)$. By applying circumscription, the abnormal predicate $\text{ab}(X) \equiv \text{ostrich}(X)$, the modified rule is $\text{bird}(X) \wedge \neg \text{ostrich}(X) \rightarrow \text{fly}(X)$.

achieve a desired state of the design solution. Hence it has predictive capability in guiding the designer in making the appropriate design decision(s).

All design knowledge transmutations in IDT are governed by the fundamental equation of inference as explained in Section 2.2.1. In view of this equation of inference, that governs the underlying inference process in the design transmutations, IDT has predictive capability.

In TPD, the plausibility of D_k is captured by a collection of plausibility statements which describe the plausibility state of each constraint in D_k in terms of ‘assumed’, ‘validated’, ‘undetermined’ or ‘refuted’. As the aim of the plausibility driven process is to achieve a plausibility state of ‘validated’ for the collection of plausibility statements, and remove any constraints that may have been refuted, TPD can predict the state of the design during its evolution.

The propositions and statements in the Theory of Technical Systems are not causal statements and as such they are not predictive in nature.

2.2.3 Operationalisation

GDT is presented in terms of the language of mathematical set theory, though its concepts and constructs can be seen at the level of abstraction more or less equivalent to the Knowledge Level [87]. The use of set theory gives GDT a certain elegance that supports formal proof of the properties of the theory, and thus of designing in general. But it is sometimes difficult to relate to actual design practice. Hence the theory lacks operationalisation.

The operationalisation of Suh’s axiomatic design theory is facilitated by the design equation in which the functional requirements (FRs) are related to the Design Parameters through the design matrix. Appropriate design decisions can be made by inspecting and manipulating the design matrix so that any coupled design or decoupled design can be made uncoupled.

Because of the abstract nature of the statements and propositions, the operationalisation of TTS is dependent on the specific domain. Its basic operator is

the transformation of material, energy and information from one state to another. The construct of the operator is thus suitable for engineering design domains (e.g. mechanical systems, electrical systems) but it will not be suitable for architectural design.

Through its well-defined plausibility statement of the evolving design, the possible states of plausibility, the representation of constraints on the design using constraint dependency graph (CDG) and a clearly defined design procedure, the TPD is well operationalised.

IDT features a set of 24 knowledge transmutations and a further set of 12 design specific transmutations for operationalisation of the theory. However it requires the user to have a comprehensive understanding of what these transmutation operators are, as the user decides which transmutation to use in the design process.

2.2.4 Comprehensibility

GDT and its evolution have been developed as a cognitive theory of the design process. There is an implicit knowledge increment as the design is evolved. This knowledge increment which is considered as learning in design is an important cognitive process that is missing from the General Theory of Design which has been developed as a Cognitive Theory of design process [87]. In developing the design process model of GDT, Takeda et al. classify the design knowledge into six categories; i.e., knowledge about entities, functions, attributes, topological relationships, connection methods and manufacturing. Between these categories of knowledge, they identify eight types of knowledge transitions or knowledge change. Its central claim is about the nature of the cognitive process that constitute human designing but there is little to suggest that it is a comprehensive theory of human cognition [87] or learning in design.

While ADT can define what constitutes an optimum design, explain or predict the evolution of the design solution through its axioms, corollaries and theorems, it can be seen as a specific theory of design phenomena that does not encompass the phenomenon of learning that co-exists during the design process.

The Theory of Technical Systems describes what is design but does not prescribe how to design. It does not consider the phenomenon of learning that occurs during design.

Like GDT and the Axiomatic Theory, TPD remains a specific theory of design. It does not feature the role that learning can play in the design of the artefact.

Although IDT claims to be a new design theory, it does not feature a generic set of design operators that one can operationalise with. It is based on the Inferential Theory of Learning by Michalski, and hence can be seen as a specific theory of learning. Although its implementation in CREDO has been applied to conceptual design, its set of 24 design knowledge transmutations is basically Michalski's knowledge transmutation and the further set of 12 operation-type design knowledge transmutations do not depict the cognitive activity of design. They are in fact operators related to logical operations or management on aspects of knowledge transformations. Hence, IDT is basically a specific theory of learning that explains the knowledge change during a design process but not in a context of a specific design action.

2.2.5 Applicability

The extended GDT has been the basis for the development of a knowledge intensive intelligent CAD system [86]. The system has been used for the design of a hydrofoil boat and the development of two types of self-maintenance photocopiers [86].

Past publications of ADT have demonstrated its application to a wide variety of design problems, such as the design of artificial skin [88], civil engineering structures [89], flexible manufacturing systems [79], manufacturing systems [90, 91, 92], metal processing [93], and paper handling mechanisms of an Automatic Teller Machine [94]. Albano et al. [89] extended the principles of axiomatic design with an interface index that serves as a quantitative measure of the complexity associated with system integration.

Hubka and Eder [81] intended the Theory of Technical Systems to have a wide applicability to technical practice especially in education where the theory can and should serve as the integrating factor. Unlike GDT or Suh's Axiomatic Theory which are expressed in concise mathematical terms that can be subjected to logical inferences, the Theory of Technical Systems is expressed in numerous, often abstract, statements and propositions that have to be comprehended before application. This seems to have limited its applicability.

Originally, the TPD paradigm was conceived in the context of computer architecture design, the 'architecture' of a computer refers to certain abstract properties. Agüero [95] has illustrated the application of TPD to the design of an architecture based on the RISC (Reduced Instruction Set Computer) style.

IDT has been applied to the conceptual design of wind braces in the skeleton structures of a tall building [84]. It has also been implemented as a system, CREDO, which was used for a planning and operation management assignment by students who submitted a wide and varied range of applications [85].

2.3 Evaluation of Models of design and learning and related systems

The following models of design and/or learning are evaluated:

- Models of the design process.
- Exploration-based model of design.
- Design reuse model.
- Design/learning loop model.

While these models reflect aspects of the design process, they also reflect aspects of learning in their formulations. The purpose of evaluating these models is to establish whether learning is depicted and if it is, in what manner is learning in design depicted.

2.3.1 Models of design

Finger and Dixon categorise models of design into either descriptive models or prescriptive models [96]. Descriptive models can be derived based on observations on how designers design from which models of cognitive process in design can be

build. On the other hand, prescriptive models prescribe how the design process ought to progress and/or what attributes the design ought to have [96]. Because it is more difficult to have a consensus on what is design (i.e. a descriptive model) than how design ought to be, there are many more proponents of prescriptive models. There are indeed many prescriptive models advocated for architectural design practice and engineering design practice [97]. Many of these models exhibit common basic features [98]. Hence, only a representation of these prescriptive models for engineering design based on their chronological dates of publication are considered here. The authors or proponents of these models are Pahl & Beitz [99], Andreasen [100], Pugh [101], Ullman [102], Ulrich & Eppinger [103]. These models of design are meant to structure the involvement of a designer throughout the life cycle of the product. These models depict the breakdown of the design process into conceptually distinct phases or activities, leading to certain intermediate results (specification, functional structure, working principle, preliminary layout, detail documentation, etc) [97]. It is not the intention here to give a detailed description of each model but to summarise the key phases in these models as shown in Table 2.1 for comparison.

Table 2.1: Key phases in various models of the design process

Pahl & Beitz	Pugh	Andreasen	Ullman	Ulrich & Eppinger
clarification of task	specification	Problem solving	specification development/planning	concept development
conceptual design	concept design	Product synthesis	concept design	
embodiment design		Product development	product design	system level design
detail design	detail design	Product planning		detail design
	manufacture		production	production ramp-up

These key phases in the various models are characterised by knowledge input from the design team at each phase and the output from each phase is usually arrived at through a consensus (i.e. a joint decision-making process). The knowledge input can be categorised into domain specific knowledge and design procedural knowledge. Pugh [101] refers to them as specific disciplined-dependent techniques and technological knowledge such as stress analysis, electronics, information on

materials and discipline or technology independent knowledge such as analysis, synthesis, decision-making and modelling etc. Within each phase, the team members apply various structured methods or methodologies. Ulrich and Eppinger posit three reasons why structured methodologies are valuable in each phase:

- They make the decision process explicit, allowing everyone on the team to understand the decision rationale and reducing the possibility of moving forward with unsupported decisions.
- By acting as checklists, they ensure that important issues are not forgotten.
- They are largely self-documenting; in the process of executing the methodology the team creates a record of the decision-making process for future reference and for educating newcomers.

The models of design are also characterised by intra-phase iterations of activities (i.e. within each phase) and also inter-phase (i.e. between phases) iterations. Ulrich and Eppinger [103] observe iteration as particularly common when the team is developing a radically new product. Konda et al. [1] observe that modification to the linear sequential process by adding feedback loops is basically a concession to the real world of design.

Explanative capability

The models of the design process describe what the design ought to be and Finger and Dixon [96] observe that they do not reflect reality. They therefore lack the explanative capability.

Predictability

Birmingham et al. [97] observed that in practice, the behaviour of engineering designers seldom resembles the behaviour prescribed in these models. These models are prescriptive in nature, intended to structure and not to predict design behaviour. These models do not reflect the reality of the design process. They are not formalised with an inference structure that can promote predictability.

Operationalisation

Because of simplified descriptions of the design process, these models use a limited set of operators to describe the design actions. Where these operators are used there is no consensus as to the meaning and clear definition of the functionality of these

operators. Hence the operationalisation is left to the interpretation of the users of the design models.

Comprehensiveness

In these models of the design process the learning process is not depicted. In discussing the relationship between creativity and knowledge, Ullman, one of the proponents of modelling the design process, states that knowledge about a domain is only gained through design experience in that domain. Ulrich and Eppinger believe that concept generation is a skill that can be learned and developed. That is learning is an implicit cognitive activity that occurs in the memory of the designer and it is by the process of learning that designers become experts in their profession. Hence the issue of learning in design being an explicit activity is not featured in these models of design.

Applicability

Most of these models are the main themes of prescribed textbooks used to teach design. They prescribe a systematic view of what design should be and hence promote a systematic mindset in the students towards design. However, Birmingham et al. [97] observed that there has not been the widespread acceptance of these models in industry than might be anticipated. This is because the nature of design activity seldom resembles that prescribed by the engineering design models.

2.3.2 Exploration-based model

The exploration-based model of design [59] characterises design as an exploratory process in which there is a co-discovery of the nature of the design problem and its solution. Hence in this model there are two types of exploration. The first explores the problem formulation in terms of a set of expected behaviours. The second exploration is directed at the production of a set of technical specifications that will deliver the functionality described in the final requirement description (i.e. the design solution). The exploration process evolves a user-defined initial design-requirement description into a final design-requirement description. The process involves a collection of inter-related activities such as search, analytical assessment, problem decomposition, parameterisation, synthesis and optimization that is performed in a sequential or concurrent manner. Underlying the exploration process is the

knowledge change that occurs as the design solution evolves, the chronicles of which provides the knowledge of the design history.

The input knowledge is:

- *Initial design requirement* R_i : incomplete and inconsistent statement of the design requirement.
- *Design Knowledge Base* (DKB) which comprises of:
 - Domain knowledge* K_{dm} : knowledge describing the space of possible designs to be explored.
 - Design knowledge* K_{dn} : procedural knowledge (e.g. design strategies, methods) about how the space can be explored.

The knowledge generated is the *Design Description Document* (DDD) which comprises of:

- *Final design requirement* R_f : a complete and consistent requirement description derived through the process of exploring the space of possible designs (SPD).
- *Final design specification*: a functional and structural description of the design solution.
- *Design exploration history* H_d : A record of knowledge acquired in the activities performed, the decisions made, and the rational behind each decision.

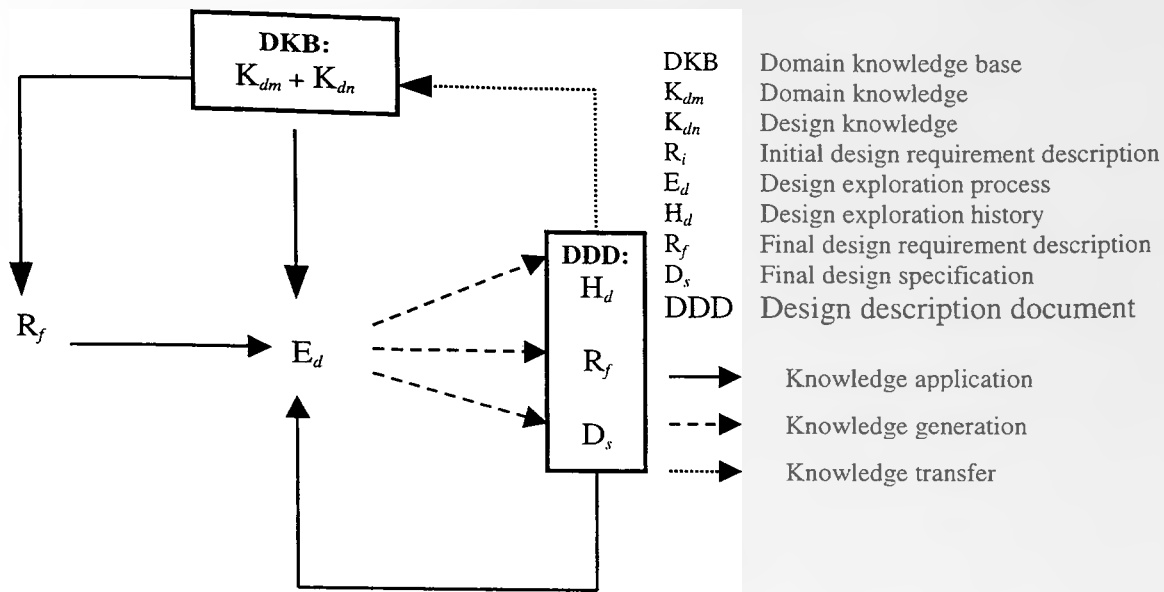


Figure 2.1: An exploration-based model of design [59]

In order to make the knowledge generated in DDD available for reuse during other design tasks the DDD needs to be transferred to the DKB built to support the class of problems concerned. The relationship between the DKB and the DDDs resulting from a series of design tasks is one of abstraction and generalisation. This process of abstraction and generalisation of knowledge from particular design tasks is an example of post-design knowledge change.

The exploration-based design model has been implemented in the system Edinburgh Designer System (EDS) [104, 105]. EDS has four main functional elements that together embody the model of design support:

1. Consistency Maintenance
2. Context Management
3. Knowledge Representation
4. Inference

In EDS, the consistency maintenance and context management subsystems provide strategic support, and the knowledge representation and inference subsystems provide tactical support. Support at the tactical level is provided by a series of general purpose support systems - forward chaining or data driven inference engines which attempt to infer by means of constraint satisfaction the consequences of the

designer's decisions. Control of the interactions between these inference engines is in the style of a blackboard system with the inference engines acting as knowledge sources [106]. The system maintains an agenda of Knowledge Source Activation Records (KSARs) which are intended to perform some particular task.

The production of a large number of alternative design solutions results in a major consistency maintenance problem. In order to consider incompatible design alternatives in isolation and also to retain alternatives for comparison and/or possible future development, an assumption truth maintenance system (ATMS) forms the core of the system and all other system components are implemented using the facilities it provides. The ATMS builds and maintains the DDD and provides an interface between the contents of the DDD and other subsystems, passing out relevant pieces of information to them as required and incorporating new information, which it receives from them, into the dependency structure.

Domain knowledge in a Domain Knowledge Base containing definitions of domain objects, called module classes, related by *kind_of* (specialisation) and *part_of* (aggregation) relations. Each module class declares a set of parameters, variables and constraints that define a particular class or type of object. The creation of an instance of a module class results in datum nodes being created in the ATMS for each of the module's parameters, variables and constraints. *Justifications* link new data introduced into the DDD as a consequence of applying a knowledge source to some particular set of domain data already present in the DDD.

Explanation

The exploration-based model of design describes and explains the design process as exploration of the space of possible designs in which an initial incomplete and inconsistent design requirement is modified into a final design specification. As a result of the exploration process, the knowledge gained is recorded in the design history. It can be examined to understand what parts of the design space were explored, how and why decisions were made. The capturing of the design history, the final requirement description, and the associated design specification provides means by which knowledge can be learned retrospectively. This includes the

description of the nature and structure of the space surveyed – the paths taken and the reasons for doing so, what it looked like on the way and what was learned as a result [59].

Prediction

Smithers et al. [104] observe that the exploration model of design lacks a formal description for the model to be more predictive.

Operationalisation

Due to the nature of exploration, Smithers et al. [104] expect the operationalisation of the model to be domain dependent in terms of the exploration activities. Nevertheless, a set of core activities constitutes the design process within the exploration-based model. They include a collection of inter-related activities such as requirement descriptions, decomposition, strategy planning, search, scheme generation, synthesis, detailing and parameterisation, simulation, analysis, prototyping, optimization and documentation that is performed in a sequential or concurrent manner. But Smithers et al. [104], in assessing the exploration model determined that it lacks the use of knowledge in the control of the design process.

Comprehensiveness

Although the exploration model of design is meant to describe the design process, it has elements that support retrospective learning for design reuse through abstraction and generalization of knowledge from similar DDDs. It therefore has limited mechanism for learning, not to the extent of IDT.

Applicability

The model of design exploration was evaluated within the domains of preliminary design of water turbine [59] and pharmaceutical small-molecule design [105].

2.3.3 Design Reuse Model

Duffy et al. [107] formulated the first *Design Reuse Model* in engineering design as illustrated in Figure 2.2. The knowledge components are:

- *Design requirements*: refers to a statement of a design need/desire.
- *Domain knowledge*: knowledge pertaining to a particular domain.
- *Reuse library*: a storage location holding reusable (compiled) design knowledge, information and data.
- *Domain model*: represents a designer's conceptualisation of a design domain that is applicable to the current design problem.
- *Evolved design model*: a statement of an evolved design.
- *Completed design model*: a statement of a fully designed new design that is believed to satisfy the design requirements.

and the processes are:

- *Design by Reuse*: the process in which knowledge resources are searched and useful knowledge can be identified, retrieved, and applied to the new design.
- *Domain Exploration*: the process of searching, understanding, generalising, and in general, "rationalising" the domain to gain an understanding of the features of that design domain from which reusable fragments of knowledge can be identified, extracted, and stored for subsequent use in design.
- *Design for Reuse*: a learning process carried out to record knowledge that is generated during design for subsequent reuse.

In this model, the knowledge learnt is comprised of the completed design model, the domain model and the reuse library. Through the *Domain Exploration* process, the completed design model together with the domain knowledge is transformed into reusable fragments (chunks) of knowledge for subsequent use in design. Through the *Design for Reuse* process, knowledge generated in the evolved design model is acquired for subsequent reuse. In both these processes there is an increase in knowledge of the design.

The design reuse model has been implemented in two systems, NODES and PERSPECT to demonstrate the learning of knowledge and subsequent application of that knowledge in design. Hence aspects of learning in design in these systems are described here.

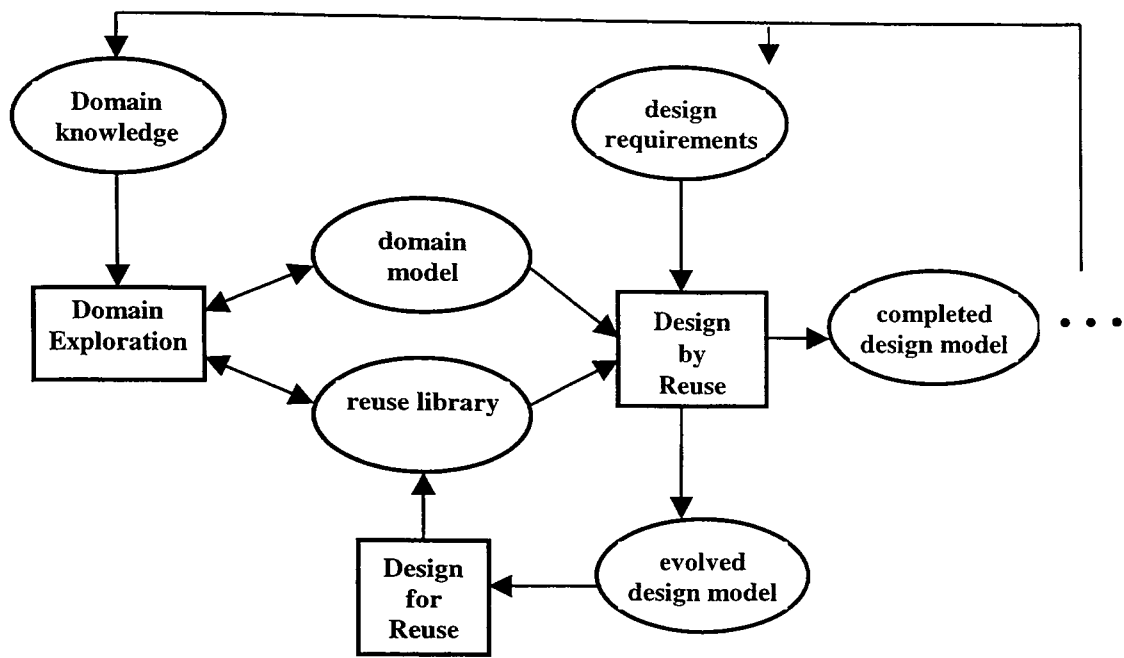


Figure 2.2: A design reuse model (Duffy et. al. [107])

NODES

NODES [108] is a Numerical and Object based Design system developed to support modeling operations during the early stages of design and provides assistance in the building, manipulation, and analysis of a model of the design artefact. To fulfill this role, the system maintains a designer initiated reuse library by allowing newly evolved concepts (models) to be stored in the library and for previously stored concepts to be updated as a result of the new concept. That is the reuse library consists of a set of classification hierarchies (concept libraries) of commonly used domain concepts.

NODES learns from past designs retrospectively in order to support the process *Design by Reuse*. When a design has been completed, the evolved model in NODES

is used to expand its knowledge concerned with that new design. In the acquisition stages, the system automatically decomposes the design into its constituent concepts, along with appropriate concepts from its corresponding library.

NODES also generalises knowledge from the most comprehensive concepts within a concept library to the less specific. Numeric parameter ranges and compositional knowledge are generalized to all associated superclasses to ensure there is no contradiction between a particular concept and its specialisation.

PERSPECT

The system couples learning and design by supporting *Domain Exploration* and *Design by Reuse*. Recognising that within the abundant explicit information of individual past designs, there exists a wealth of implicit knowledge that should be made explicitly available to the designer, according to the different viewpoints that best suits the designers' particular needs, Kerr [109] introduces the concept of *Customised Viewpoints*. PERSPECT represents a dynamic design tool that implements the concept of *Customised Viewpoints*. PERSPECT applies clustering and statistical approaches to rationalise past design examples into abstract groups and support the application of the generated knowledge to develop a new design solution.

Explanative capability

The design reuse model describes and explains how knowledge that is generated during design can be stored for subsequent reuse. It also explains how the stored knowledge (both current and past similar designs) can be searched, abstracted into reusable fragments of knowledge and stored for subsequent use. It therefore explains an aspect of the learning in design phenomenon. That is, learning knowledge retrospectively in order to harness it for subsequent use.

Predictability

The design reuse model does not feature any predictive capability in the sense of predicting when should learning take place or given a design activity, what aspect of knowledge is relevant.

Operationalisation

Two systems, NODES and PERSPECT were implemented to show how the reuse model can be operationalised.

Comprehensiveness

The reuse model is an example of specialised model of learning in design as its architecture is aimed at learning abstract or implicit knowledge from past designs.

Applicability

Although the reuse model has been applied to preliminary design (e.g. NODES and PERSPECT), it is conceivable that it can be used in other design scenarios (e.g. design for manufacture, etc.) in which knowledge needs for different perspectives of the design cannot be harnessed directly from past designs.

2.3.4 The Design/Learning Loop

Duffy and Duffy [110] postulate how the activities of design and learning are coupled in Figure 2.3. The lower loop suggests in-situ learning and application of knowledge that occurs as the design solution is evolved from an initial stage, Stage: 1 to a design solution specification, Stage: N. In-situ learning occurs when knowledge learnt from the solution and its development is fed back to some store of experiential knowledge. Application of knowledge occurs when this knowledge is reused to aid in the evolution to an acceptable design solution. Some of the learned knowledge will transform to long- term experiential knowledge to be reused in later design scenarios whereas some of the transient knowledge learned will be used to assist in the evolution of the design. In addition to the lower loop, there is another loop that represents updates or modifications of the experiential knowledge depicting the designers' ability to explore and learn from their own (specific or general)

Shared Learning

Duffy and Duffy [24] offer a concept of computational learning that can ease the burden of knowledge elicitation and yet maintain (i.e. update and evolve) experiential knowledge depending on the knowledge needs of the designer. Traditional Machine Learning in Design (MLinD) systems reflect viewpoints defined by knowledge engineers and hence present predefined perspectives of designers' required viewpoints of knowledge. Hence, they introduce the concept of Shared Learning in which the designer and a computing system as a learning assistant co-learn when a designer learns new knowledge as a result of a learning assistant automatically learning and presenting previously implicit, and therefore unrepresented knowledge.

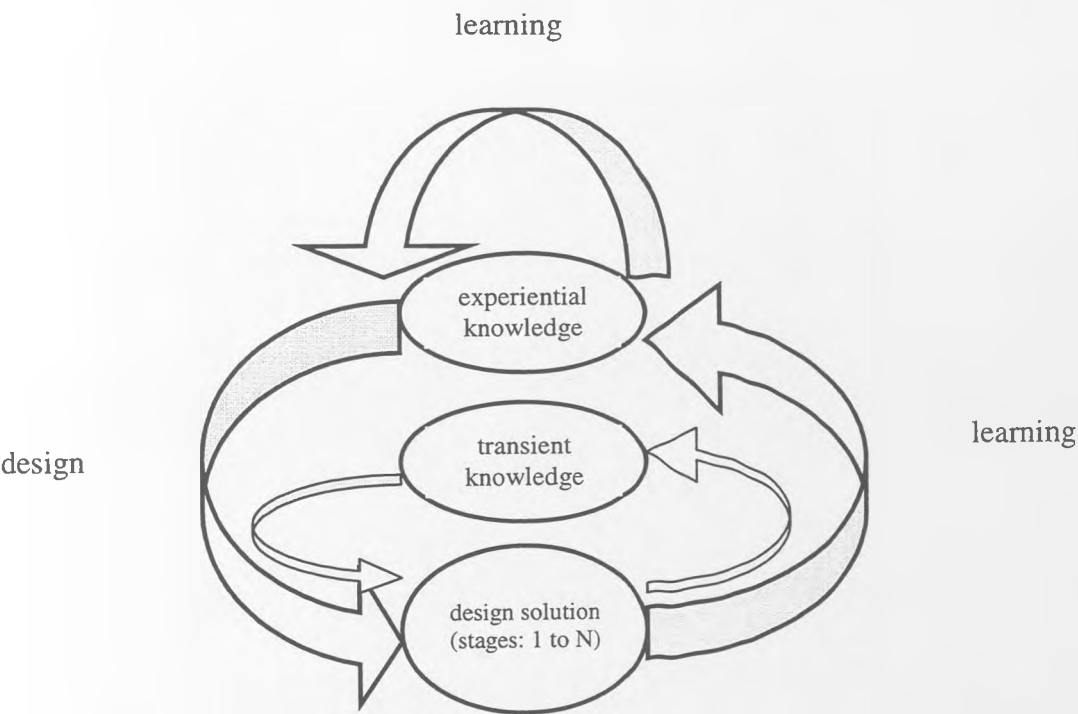


Figure 2.3: Design/Learning loop (Duffy & Duffy [24])

PERSPECT

PERSPECT demonstrates the coupling of design and learning activities within a

single computational environment. The designer using PERSPECT co-learns when PERSPECT is behaving as the learning assistant and is able to abstract design knowledge that meets the knowledge needs of the designer. PERSPECT’s ability to extract and utilize general knowledge originating from past designs is used to assist in two main design tasks. These tasks are domain model preparation and design model instantiation. Figure 2.4 shows the coupling of design and learning activities and details the activity paths taken during the two tasks.

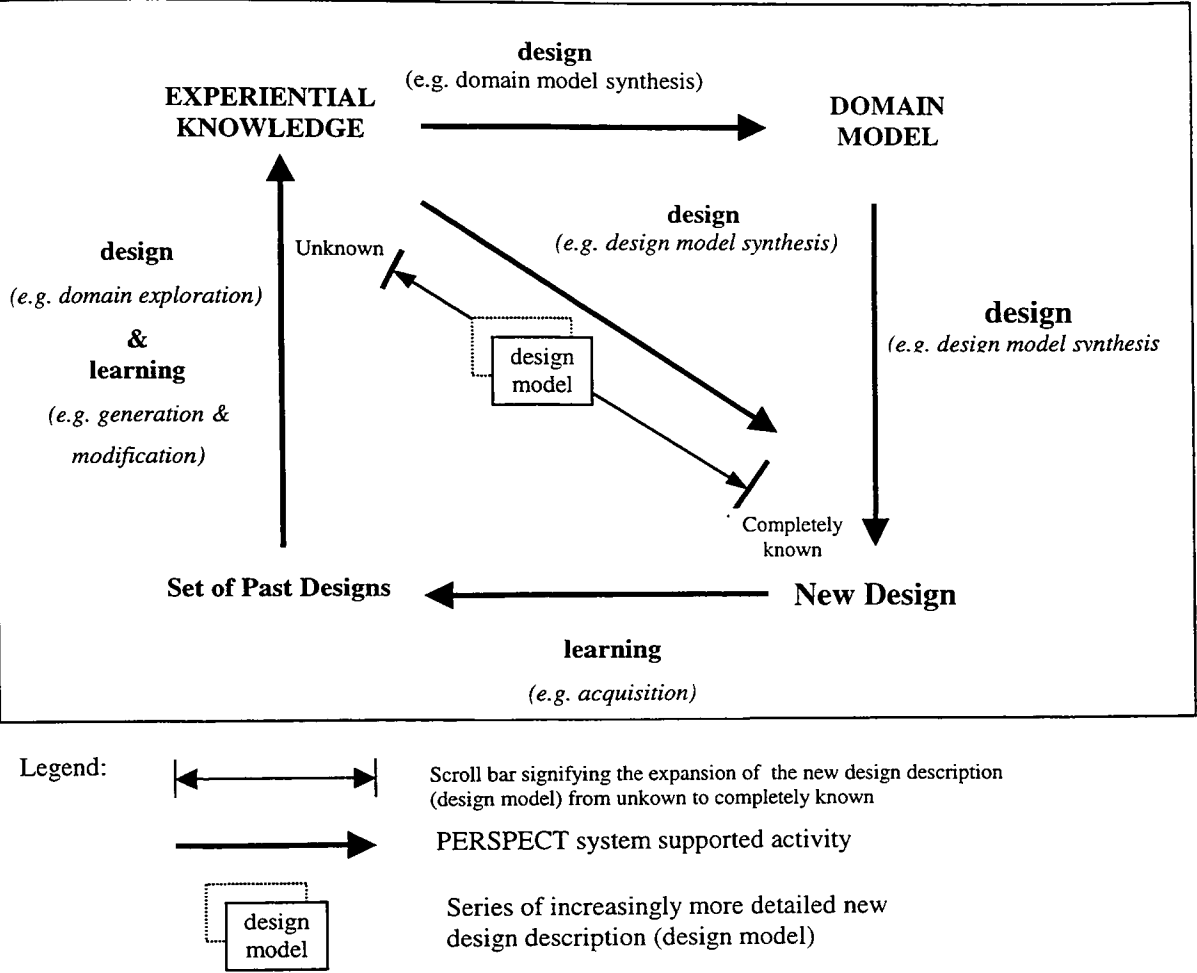


Figure 2.2: PERSPECT’s activity diagram: Coupling design and learning (Kerr [109]).

The first task is the domain model preparation using past design knowledge. A domain model is a representation of a design domain describing the experiential knowledge of that domain. This involves:

- Rendering new experiential knowledge.
- Checking and updating precompiled experiential knowledge of an existing domain model that describes the design domain.

The domain model preparation task required by the design activity of domain exploration illustrates the coupling of design and learning activities in which a set of past design knowledge is learned (generated or modified) to prepare the domain model.

The second task is design model instantiation which involves the allocation and estimation of attribute values for a new design. This involves:

- PERSPECT using a sub-system *Designer* to choose the most reliable empirical equation with minimal information requirement from a respective set available in the domain model.
- PERSPECT using another sub-system *Ecobweb* to generate viewpoints of experiential knowledge from which information of the most similar past design or generalisations of similar past designs can be identified and used to estimate unknown attributes.
- PERSPECT using sub-systems S-Plus and Designer to render new empirical equations or less complex empirical equations in order to simplify and/or supplement an existing domain model. The equations are simplified through the process of abstraction and generalisation.

The design model instantiation task required to synthesise a new design again illustrates in-situ learning using past design knowledge stored in a computer system to aid in the design synthesis process.

Explanative capability

The model of Design/Learning loop is able to explain how design and learning can be coupled (i.e. in-situ learning during the design process) and also learning from past design experience (i.e. retrospective learning). It is further developed into the concept of Shared Learning in which the designer co-learns with a computer.

Predictability

The model however does not provide a formalism by which to predict different types of knowledge learned and how they are learned for different types of design activities. However, in the implementation of PERSPECT, the system is able to predict and therefore provide implicit knowledge available in explicit past design cases.

Operationalisation

It does not define a definite set of operators which the users of the model (i.e. designers or knowledge engineers) can identify with for the implementation of the model. Nevertheless PERSPECT was implemented to illustrate some aspects of learning in design.

Comprehensiveness

As a model to explain the phenomenon of learning in design it is by far the most comprehensive model available. However, it is not a general model of learning in design as it does not explain how knowledge is learnt in the context of a typical repertoire of the design activities encountered in design practice.

Applicability

The design/learning loop model can be illustrated in the concept of Shared Learning in which the designer learns as the computational learning system learns. Two systems, PERSPECT and SPIDA [111] implement the concept of Shared Learning applied to preliminary numerical design and conceptual layout design respectively.

2.4 Machine Learning in Design (MLinD) Systems

The aim of computational learning is to automatically increase, or improve knowledge held within a computer system. Thus, computational learning offers the potential to ease the burden of knowledge elicitation and entering increasing amounts of complex information. Recognising the potential benefits of computational learning to aid in engineering design, researchers and the Intelligent Computer Aided

Design (Int. CAD) community have developed many MLinD systems or tools. These systems have been developed to ease the burden of knowledge elicitation and knowledge acquisition tasks of knowledge engineers [112]. Hence these systems are directed at empowering CAD tools with learning capabilities by using a variety of machine learning techniques to acquire some specific design knowledge. Depending on the type of design knowledge to be acquired for a particular design activity or task, the MLinD systems have been developed to meet that knowledge need. But the knowledge need of designers is dependent on the design activity and hence varied and dynamic in nature. Thus these MLinD systems have been developed on an *ad hoc* basis and therefore do not support the knowledge needs of designer effectively. Hence Sim and Duffy [32] argue for the need to develop MLinD systems on a formal basis in order to support designers more effectively. It has been argued earlier that for systems to be effective in supporting designers they ought to be developed based on some theory or model which describes the phenomena in question. Nevertheless Sim and Duffy [32] have used these MLinD systems as instances to evaluate a proposed model of the learning process (See Chapter 6). Hence it is not the intention here to evaluate these MLinD systems in the manner in which the various theories and models of design and learning have been evaluated.

2.5 **Summary**

Several theories, models and related systems that may describe the phenomenon of learning in design have been evaluated with respect to a set of criteria. These criteria relate to the explanative and predictive capability of these theories or models, the ease of operationisation of the theories or models into practice, applicability and their comprehensiveness in describing the phenomena.

Tables 2.2 and 2.3 present a summary of the evaluation of the various theories and models respectively based on the set of criteria.

Table 2.2 shows that of all the theories evaluated, only one is a theory of learning (i.e. IDT) and the rest are basically design theories. Table 2.3 shows there are

several models that describe some aspects of the phenomenon of learning in design. But none of these explain fully the phenomenon of learning in design. Although the exploration-based model chronicles the knowledge gained during a design process, it fails to explain the manner in which the knowledge gained is achieved (i.e. how learning takes place). The design reuse model explains one aspect of learning in design, that of learning abstract or implicit knowledge from past designs. The design /learning loop model describes the coupling between design and learning activities but fail to explain the nature and the manner of that coupling. Hence it is reasonable to infer that none of these models of design and learning explain the phenomenon of learning in design comprehensively.

Table 2.2: Evaluation of theories

Criteria	GDT	AT	TTS	TPD	IDT
Explanative capability	Yes	Yes	Yes	Yes	Yes
Predictibility	Yes	Limited	None	Yes	Yes
Operationsalisation	Yes	Yes	None	Yes	Yes
Comprehensiveness	Design theory	Design theory	Design theory	Design theory	Learning theory
Applicability	Few	Many examples	Few	Few	Few

Table 2.3: Evaluation of models of design and/or learning

Criteria	Models of design	Exploration-based model	Design reuse model	Design/Learning loop model
Explanative capability	Poor	Yes	Yes	Yes
Predictibility	Poor	Poor	Poor	Limited
Operationsalisation	A limited set of design operators	A limited set of design operators	A limited set of design/learning operators	A limited set of design/learning operators
Comprehensiveness	Model of design process	Retrospective learning	Retrospective learning	Retrospective/ In-situ learning
Applicability	Used in teaching design	Few	Few	Few

The following conclusions can be made with regards to the aim of this Chapter. It seeks to establish if there is a theory or model of learning in design to describe the phenomenon. The critical review and evaluation suggest that:

- There is currently no general theory or model to describe the phenomenon of learning in design.
- The models of learning in design evaluated present some aspects of the

phenomenon of learning in design.

- For operationalisation to be successful leading to the building of systems to evaluate the theories or models, there is a need for a clearly defined set of operators for design and learning.
- Although the design/learning loop model attempts to show the coupling between design activity and learning activity, the nature of the coupling needs further investigation and evaluation.

Hence in order to develop a model or theory of learning in design, it ought to exhibit explanative and/or predictive capabilities. To achieve this, the model or theory needs an underlying inference process as exemplified by GDT, TPD and IDT. A design support system based on such a theory or model can then provide explanations to the user or predict the likely consequences of their design decisions. In order to implement an effective design support system, the model or theory must feature a clearly defined set of operators that depict the events in the phenomenon in question. If the model or theory has explanative and predictive capability, and can be easily operationalised, it will be a comprehensive model or theory of the phenomenon and will be widely applicable.

To explain the phenomenon of learning in design, the model that is envisaged ought to feature the following characteristics:

- A formalism that represents the repertoire of typical design activities encountered in design practice so that knowledge change that occur can be explained in terms of the input knowledge, output knowledge and the goal of the design activity.
- A formalism that depicts the learning activity in terms of elements which explain what design knowledge is learnt, why learning occurs, how learning takes place and when is learning triggered and in what manner these elements interact.
- The nature of interactions that occur between the design and learning activities (i.e. to postulate hypotheses on the nature of these links and substantiate them by experimentation or evaluation).

Chapter 3

Research methodology

The research task involves a series of steps that is initiated by a research problem leading to a proposed solution. The hypothesis that there is no comprehensive model or theory of learning in design is shown to be true as a result of the critical literature review in Chapter 2. As a result of this finding, the formulation of the research problem of the thesis (described in Chapter 4) is the need to develop a model or theory of learning in design. Developing a model or theory of learning in design, however requires a research methodology. Meaningful research must be rigorous, relevant or significant, conducted with integrity and independence. It is desirable that an appropriate research methodology would ensure that the conclusions of the research effort are based on rigour, relevance and significance and that it was conducted with integrity and independence. One approach to adopt is what is generally known as the scientific approach. Smith and Dainty [113] encouraged researchers to adopt a research perspective that may be defined by different possibilities, assumptions, values and paradigms. By considering the strengths and weaknesses of different perspectives, one or several approaches that best serve the research purpose can be adopted. In the domain of Artificial Intelligence (AI) and application of AI techniques to solving problems in engineering design, analysis and manufacturing (AIEDAM in short), Reich [114] has shown that there is a variety of research methodologies that can be considered. Hence before a particular research methodology is adopted here, some of these methodologies have been reviewed. The nature of the phenomenon on learning in design may influence the methodology adopted and hence the rationale for the choice would be discussed before a methodology is adopted.

3.1 A model of a world view

Reich [114] argues that research methodology is intimately connected with, and constrained by, the world view it serves. Guba [115] states that the issues that characterise the world view and hence the research methodology are:

Ontology: What is the nature of the “knowable”? Or what is the nature of “reality”?

Epistemology: What is the nature of the relationship between the knower (the inquiry) and the known (or knowable)?

Methodology: How should the inquirer go about finding out knowledge?

3.1.1 Ontology

Ontology is the branch of philosophy concerned with the meaning of existence in the broadest sense, including, the existence of everyday life-world objects such as screwdrivers, as well as more paradoxical entities such as unicorns or charmed quarks [61]. The answers to what constitutes an ontology must lie somewhere between extending from extreme realism, which emphasises the factual content of science, to the opposite pole of conventionalism, which stresses the theoretical characteristics of scientific knowledge. The basic argument of realism is that science is grounded firmly on empirical observation. Our observations or notions of entities acquire substance and clarity as their existence is corroborated by further investigation until they attain ontological status. While ontology may not be based on materialism alone, conventionalism assigns to scientific knowledge no higher status than that of being a useful hypothesis. For example, the map of science may be the most effective scheme for describing and ordering natural phenomena in an extraordinarily concise form, but however well it is corroborated it remains essentially fictional, whose entities are conventional symbols, not representations of the real things in themselves. Conventionalism also allows for continual change in the theoretical contents of science whether by new discoveries, by the correction of apparent errors or by the consolidation and reformulation of existing concepts.

3.1.2 Epistemology

Epistemology is the branch of philosophy that deals with the theory of knowledge [116]. It is basically a study of the nature of human knowledge. The fundamental concern of epistemology in science is how much of scientific knowledge can be considered true, or how firmly it should be believed [61].

3.1.3 Methodology

Methodology deals with methods for creating knowledge about the world and the interpretation of this knowledge in the light of ontological and epistemological positions [114]. The term *methods* refers more specifically to individual techniques (e.g., surveys, participant observation), whereas *methodology* can be construed broadly to suggest both the presuppositions of methods, as well as their link to theories and implications for society. Methodology, in short, more clearly implies a concern, in an overall strategy of constructing specific types of knowledge, and is justified by a variety of meta-theoretical assumptions. Methodology is thus inevitably prescriptive because it attempts to legitimise the use of particular methods in ways that are consistent with the development of the specific theory in question [117].

3.2 Research perspectives

There are several research perspectives that one can adopt. These are influenced by the world view of science. The common world views of science are: scientism, critical theory (ideolism) and practicism [115]. Reich [114] posits that AIEDAM research is about gaining understanding of some phenomena (e.g. what is design?), while others stress the practical relevance of research (e.g. how can we aid design?). While these differing objectives originate from the two perspectives of scientism and practicism, each of these objectives through its appropriate method of inquiry is most suitable to achieve a specific research objective. Since the objective of the research identified is to develop a model of learning in design in order to explain the phenomenon, and using the model as a basis to build systems to support design, it is important to consider the research perspectives of scientism and practicism as they could influence the research methodology adopted here. Fulcher and Hills [121] argue that because of the nature of design research, the effective research methodology for design research may be a hybrid one derived from the two perspectives of scientism and practicism. Critical theory [115, 117] or ideolism as a label for a research perspective that includes ideologically oriented paradigms such as neo-Marxism, historical materialism, re-constructed historical materialism and feminism is, however, not relevant here. Hence only the research perspective of

scientism and practicicism will be described below.

3.2.1 Scientism

Scientism represents the essence of the world views like rationalism, positivism, post-positivism and logical empiricism [114]. Scientism asserts the existence of an external world which is the primary concern of science. Scientism assumes that the reality it purports to study has some discernible regularity or order in what is observed that is independent of time and location [61]. Scientism demands that researchers stand outside of time and context, and indeed, “outside of themselves as persons”, in order to deliver research results that stand apart from human values. This is to ensure objectivity in the result and that it is free of bias or prejudice.

The position of scientism concerning the three issues is as follows [115]:

Ontology: *Realist* – Reality exists “out there”. Reality is governed by immutable natural laws of cause and effect. By discovering these laws or theories, science achieves its goal to predict and control the phenomena.

Epistemology: *Objectivist* – Researchers can acquire objective knowledge about the real world through employment of appropriate methodology.

Methodology: *Experimental/manipulative* – Scientific method in which hypotheses are stated in advance and are subjected to test under carefully controlled conditions. The researcher adopts a distant position, thus achieving value-free knowledge (i.e. knowledge that is not context-dependent).

3.2.2 Practicism

Practicism is a world view that captures the essence of perspectives such as action research, human-centred engineering or critical constructivism [114].

The position of practicicism concerning the three issues is as follows [115]:

Ontology: *Relativist* – Reality exists as mental models (constructs) within a certain value-laden theoretical framework. This results in various interpretations of researchers dependent on the theoretical framework

adopted. Unless there is a foundational process by which the ultimate truth or falsity of these several constructions can be determined, there is no alternative but to adopt a position of relativism.

Epistemology: *Subjectivist* – As realities exist only as mental models of the researchers, subjective interaction will contribute to the knowledge of the world. To avoid misuses of subjectivism, a critical methodology must be adopted.

Methodology: *Critical hermeneutical/dialectical* – Consensus is sought among the various constructions elicited. This process involves two aspects: hermeneutics and dialectics. The hermeneutics consists of depicting the individual constructions as accurately as possible, while dialectics consists of critical comparison and contrast of these constructions so that informed choice(s) can be made. In the process the “world” is reconstructed in the minds of the researchers.

3.2.3 Evolution of Research

The study of the history of science has laid the foundation of various models of scientific change [61]. Scientific change is often seen as a result of the research effort of a community rather than that of a single scientist. Of the models, two that resemble the world view of scientism and practicisism are Kuhn’s research paradigm [118] and Lakatos’s research program [119] respectively.

Research Paradigm

The key feature of Kuhn’s theory of scientific change is the emphasis placed on the revolutionary character of the scientific progress, where a revolution involves the abandonment of one theoretical structure and its replacement by another compatible one [120]. Another important feature is the important role played by the sociological characteristics of scientific communities.

Kuhn sees research as an evolutionary process in which there is a period of “normal science” culminating in a period of scientific revolution which leads to another period of new normal science. The history of science gives evidence to what Kuhn described as scientific revolutions whereby one theory is superseded by another more

generalised theory; for example the Newton's law of motion being replaced by Einstein's theory of relativity. In the period of normal science, scientists employ various imprecise criteria to guide their research. The outcome of the research can be described as Kuhn's paradigm in which there is a corpus of concepts and theories shared by a group of researchers. This paradigm would consist of theoretical assumptions and laws and the techniques for their application that members of a particular scientific community would adopt. When certain anomalies are discovered, paradigms may clash and a crisis may arise leading to revolutionary changes that cannot be explained on the grounds of empirical evidence or rationalisation. A crisis is resolved when a new paradigm emerges and attracts the allegiance of the scientific community

Kuhn's theory of research paradigm and scientific revolutions seems to invoke a relativist position [120]. It is not the aim here to discuss the alignment of Kuhn's theory in terms of its epistemological stance (i.e. objectivism or relativism) but rather to state the important role that the scientific community contributes towards the development of a theory or paradigm.

Research Program

Lakatos's alternative to Kuhn's paradigm is the phrase "research program". Lakatos views science as having a set of fundamental principles that form the hard core of the research program augmented by a set of auxiliary hypotheses. The fundamental principles is seen as the defining feature of a science and the basis of development of the research program and the auxiliary hypotheses as a protective belt to emphasise its role in protecting the hard core from falsification. The main indication of the merit of a research program is the extent to which it leads to novel predictions that are confirmed and a progressive program of research it has to offer [120]. A progressive research program is one that retains its coherence and leads to novel predictions that are confirmed, while a degenerative program is one that loses its coherence and/or fails to lead to a confirmed novel prediction. Progress is seen as the replacement of a degenerating program with a progressive one, with the latter being an improvement on the former based on its more efficient predictor of novel phenomena. Hence Lakatos' version of Kuhnian's revolution involves the ousting of

one research program by a superior program. Lakatos's methodology however is in no position to diagnose any contemporary theory as non-scientific since judgment can only be made in retrospect through the benefit of historical hindsight.

The discussion on the two models of scientific change highlights the role that the scientific or research community plays in contributing towards the evolution of theories or models. It is difficult to predict whether the model of learning, which evolved as part of the research work will be in accordance to the research perspective of Kuhn's paradigm or Lakatos's research program. However, it was considered as a merit in involving the research community by presenting the proposed model of learning in design for their critique as a means of evaluating and evolving a better model. This is described further in Section 3.3.3, Step 8.

3.3 Choice of Research methodology

Fulcher and Hills [121] argue that the choice of a research methodology for design research is rendered difficult because of the nature of design. Design is not a science of the natural, but a science of the 'artificial' [122]. It is considered as an investigation into what could or should be, not an analysis of what is [121]. Developing a model or theory of learning in design requires the knowledge of two human cognitive activities, that of design and learning and their interactions. Learning is grounded in the logical process of induction, deduction and abduction [63]. Learning in design as a phenomenon of design practice has to do with human learning behavior and cognition in design. Unless it is possible to analyse in a formal way, what is learning in design in order to derive a model, it is subjected to interpretation of what it could be or should be, the basis upon which many design support systems have been developed. That is, the appropriate world view to derive a model of learning in design that is objective, universal and context independent is that of scientism. But the research methodology of scientism requires that results from two or more experiments under similarly controlled conditions to be the same. In the context of validating or evaluating the model of learning in design, the dependence on human designers to be the subjects of experimentation in evaluating the model would pose a stumbling block in achieving unequivocally the same results

in two nominally identical experiments. This is because no two designers are alike in their design practice. So, although the principal research methodology adopted here is that of scientism, the method of evaluation is not through experimentation as is the norm for traditional scientific method, but through a case study of a designer at work. The results of the case study will be used to modify the model, a procedure of evaluation that is more akin to that of practicism. Thus the research methodology of formulating the model in terms of formalisms and hypotheses is based on scientism and the methodology of evaluation is that of practicism. Hence the research methodology adopted here is a hybrid one, that of pragmatic scientism.

3.3.1 Ontology of learning in design

Learning in design as a phenomenon is inferred to exist from the experience of designers. Designers do learn while designing which explains the different design strategies adopted by an experienced designer and a novice towards a common design problem [123]. Understanding the cognitive behaviour of designers is possible through the design activities that they perform. Understanding the process of learning in design requires a simplification of the complex cognitive behaviour into that of design activities and learning activities. This simplification is necessary in order to derive an ontology for design activities and learning activities. The ontology of design activities and learning activities can be derived through the corpus of concepts and theories shared by research communities of design, cognitive science, machine learning and machine learning in design. An important property of deriving an ontology is that it gives a precise and unambiguous definition of concepts necessary for representing knowledge [124].

3.3.2 Epistemology of learning in design

Ontology is closely linked to epistemology in that it is necessary to have a conception of the nature of reality (in this case learning in design) before one can justify a scientific analysis of it. Having identified what design and learning activities exist, the purpose is to study the knowledge transformation that occurs as a result of these activities. In order to analyse these activities in a systematic manner, a formalisation of a model of the design activity and a model of a learning activity

will be required.

3.3.3 Methodology

The purpose of this research is to develop a model or theory of learning in design. Since the research methodology adopted is that of pragmatic scientism, a model of the methodology is adapted from Reich [114] and Duffy and O'Donnell [125]. Figure 3.1 shows the major steps in undertaking this research and the relationship of each step with the organisation of this thesis. These major steps are grouped into three parts as follows:

- Part 1 – Problem identification (Steps 1 to 4).
- Part 2 – Solution definition (Steps 5 and 6).
- Part 3 – Evaluation and conclusion (Steps 7 to 10).

Step 1

Design practice is shown to be characterised by the cognitive activities of design and learning, resulting in the developments of MLinD systems on an ad hoc basis. A design problem is identified in that there is a lack of understanding regarding the phenomenon of learning in design and therefore inadequate provision of functionality in design support systems or design tools in acting as assistants to designers throughout the design process.

Step 2

This led to the hypothesis that there is no comprehensive model or theory of learning in design. A critical review of the literature on theories and models of design and some models of learning in design using a pre-defined set of criteria (that of explanative capability, predictive capability, operationalisation, comprehensiveness and applicability) shows the hypothesis to be true. This leads to the statement of the research problem and its scope in Step 4.

Step 3

Before initiating the research program, various research perspectives that influence the nature of the research methodology needed to be identified and evaluated against the research problem formalised in Step 4.

Step 4

Having identified the need to develop a model of learning in design, Step 4 is to describe what is the nature and features required of the model that will explain the phenomenon of learning in design. This requires a clear definition of what is learning in design to differentiate from other forms of learning.

Step 5

Through the process of induction, by analysing the publications of research communities and an understanding of design practice, a model of design activity and a model of learning activity are formulated and formalised.

Step 6

A model of learning in design that considers the coupling of the design and learning activities was developed. Various hypotheses regarding the inextricable link between the design activity and learning activity are postulated.

Step 7

In order to evaluate the model of learning in design, a protocol recording of a designer performing a design task was recorded. Protocol analysis of the transcript of the recording was analysed in terms of the knowledge change that occurred in the design activities and learning activities and against the hypotheses defined in Step 6.

Step 8

In Step 7, the proposed model of learning in design was evaluated using the results of the protocol analysis. The model was presented at the Machine Learning in Design Workshop [126] for critic and feedback. The evaluation through the protocol analysis and peer critique was to provide a basis for modifying the model. The evolved model was evaluated in terms of the pre-defined set of criteria (used in Step 2) in providing an understanding of the phenomenon of learning in design and hence highlights the main contribution of this research work.

Step 9

Step 9 is the discussion on the strengths and weaknesses of the research methodology adopted and the outcome of the research task in developing a model of learning in design. Based on the results achieved thus far, it is necessary to discuss the future work to be pursued in order that the evolved model can become a theory of learning

in design.

Step 10

Step 10 summarises the main contribution and other supporting contributions of this research work.

3.4 Summary

The various world views and their influence on the research perspectives of scientism and practicisim in terms of ontology, epistemology and research methodology have been reviewed and discussed. A research methodology that is a hybrid between scientism and practicisim was chosen to develop the model of learning in design.

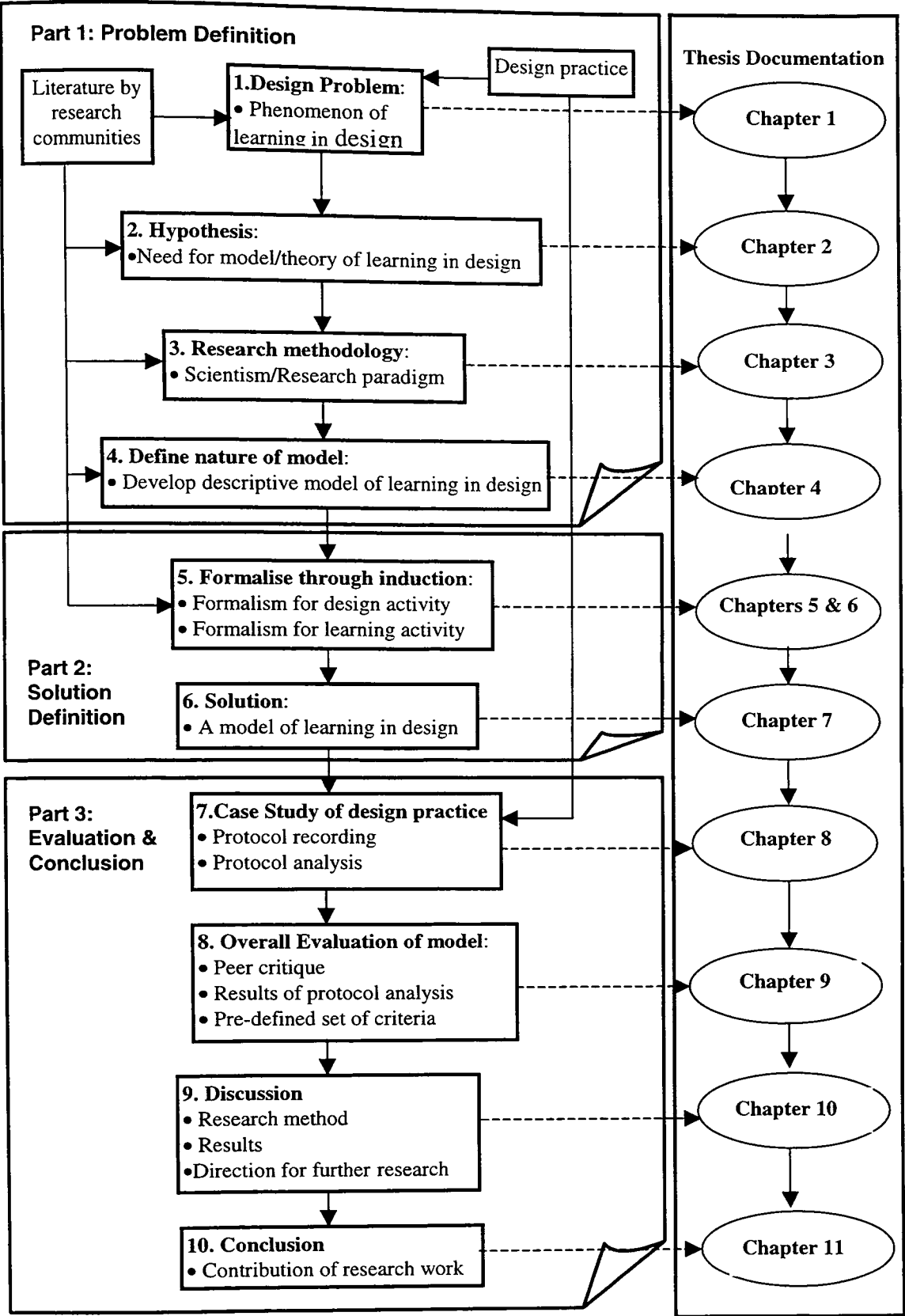


Figure 3.1: The research methodology for developing a model or theory of learning in design.

Chapter 4

Learning in Design

Having established the need to develop a model or theory of learning in design in Chapter 2, the purpose of this Chapter is to establish the nature of learning in the context of design, to define what does learning in design entail. This understanding then provides the basis of the two approaches taken towards developing a model or theory of learning in design (see Section 4.3.2).

4.1 The context of learning

Whilst there may be different meanings given to learning by different researchers, learning is considered here as a cognitive process by which knowledge is gained and stored in some permanent depository known as memory. Learning, however cannot exist in *vacuo*. It has to be considered in the context of a domain and the nature of knowledge gained is related to that domain.

Learning requires a context or a situation. It can be said that when an individual encounters a learning task (e.g. a problem to be solved) their understanding of the problem or of the learning situation itself and their understanding of the phenomena that are relevant to the problem or situation are related [42]. A dictionary definition of a situation is the manner in which something is placed in relation to its surroundings at a given time. Hence, a situation is always experienced with a socio-spatio-temporal location – a context, time and a place [42]. A phenomenon is experienced as abstracted from a situation. That nobody can experience a phenomenon in the absence of a situation is strongly intuitive [42]. A situation can be experienced in terms of our ability to make sense of the “here and now” only through our past experience. Thus not only is our experience of a situation moulded by the phenomena as we experience them, but our experience of the phenomena is modified, transformed and developed through the situations we experience them in. This led Reffat and Gero [127] to state that situated learning is based on the notion that knowledge is more useful when it is learned in relation to its immediate context, i.e., its situation, and less useful when it is learned out of context. Design is an

example of a situation or context in which designers learn in the context of design.

4.1.1 The situatedness of learning in design

In the ‘world’ of the designer, the designer converts indeterminate situations into determinate ones. Beginning with situations that are in part uncertain, ill-defined, complex and incoherent, designers construct and impose a coherence of their own. They discover consequences and implications of their constructions – some unintended – which they appreciate and evaluate [128]. Through a series of actions, consequences, implication, appreciation and further actions, the designer re-frames the design problem/solution until a coherent one is achieved. In all these actions, the designer is performing a series of thought experiments, some of which result in concrete actions to make the design more coherent. Schon [128] calls the re-framing of the design problem a reflective conversation with the situation.

In the designers’ conversation with the material of their design, they can never make a move that has no effect intended for it. Their continuous interaction with the design, causes them to apprehend unexpected problems and potentials. As they appreciate such new and unexpected phenomena, they also evaluate the decisions or actions that have created them. They use their experience and knowledge to decide their course of action. They thus build up a repertoire of examples, images, understandings and actions. Each new experience of reflection-in-action enriches their repertoire. This may be generalised to other cases contributing to their repertoire of exemplary themes from which in subsequent cases of their practice, they may compose new variations. Thus Schon, by using the model of reflection in action, describes design as consisting of a series of dynamic situations which the designers steer to their ultimate goal of achieving coherent design(s). In the process they bring to bear their past experience on the evolving design and learn new experiences at the end of it.

This situatedness of design in which designers learn design experience points to a phenomenon – that of learning in design. Design is one example of the context in which learning can be experienced. Hence it can be said that learning is inextricably

linked with a situation or context in which the designer or the participant gains experiential knowledge of the situation. This observation is supported by protocol studies in design in which it has been shown many times that experienced designers in all problem areas can reach satisfactory design solutions more effectively than novice/naive designers [123].

Although the above description of phenomenon of learning in design implies that learning occurs during the design process, learning of experiential design knowledge can also occur outwith the design process, from past designs [129]. This is because designers do learn from reviewing the failures of past designs or the salient features of successful designs in retrospect and gain design knowledge in the process. Hence learning in design can occur in-situ or outwith the design process.

4.1.2 Learning experiential knowledge

Kolb [130] views learning as a transaction between the person and the environment or situation; the environment is the ‘real world’ and not the artificial settings found in the laboratories of some cognitive experiments. In this transaction between the domain/social knowledge and personal knowledge, new knowledge is created. In the ‘real world’, “Knowledge is continuously derived from and tested out in the experiences of the learner”, Kolb [130]. Hence Kolb [130] remarked that people do learn from their experiences.

The nature of a unit of experience

Through their 25 years of systematic research into the experience of learning, Marton and Booth [42] are able to give a succinct definition of what constitutes a unit of the learner’s experience of learning. The main thrust of Marton and Booth’s research is: “*How do learners gain knowledge about the world, and why do some do it better than others?*”

To answer these questions they perform a number of studies of students’ learning through reading texts, writing essays, writing programs and solving problems. Through their experimental studies of learners, Marton and Booth are able to show that learning as an experience has a structural aspect and a referential (or meaning) aspect. The two aspects, meaning and structure are dialectically intertwined and

occur simultaneously when we experience something. The structural aspect refers to the context in which one discerns the parts in relation to the whole in that experience. Moreover intimately intertwined with the structural aspect of the experience is the referential aspect, i.e., the meaning. In perceiving the parts and the whole, and the relationship between them, the meaning is discerned.

Using these two aspects Marton and Booth characterise a basic unit of experience as illustrated in Figure 4.1.

The external horizon refers to the way in which the phenomenon we experience in a certain way is discerned from its context, but also to relate it to its context, and even to other contexts at the same time. The internal horizon refers to the parts and its relationships to the whole in relation to the context.

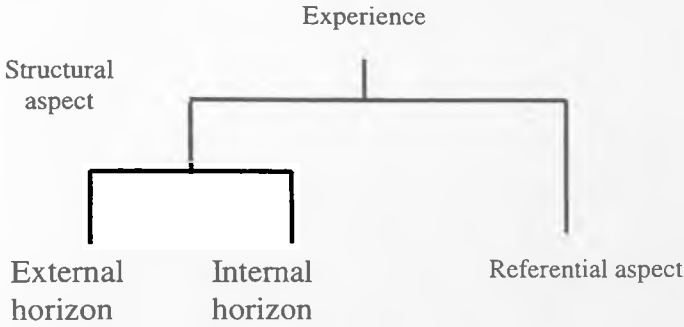


Figure 4.1: The unit of experience (adapted from [42])

In the context of design, MacCallum [25] observes that the whole process of undertaking and managing design depends for its success on good design experience. Duffy and Kerr [129] refer to design experience as consisting of knowledge generated from personal exposure to events and artefacts. Hence, the referential aspect in Figure 4.1 refers to the design situation while the external horizon refers to the past experience in related context and the internal horizon refers to the current design focus in relation to the design.

The fact that learning is a continuous process grounded in experience has important implications in design and in the development of design support systems. What is

available through design experience is a wealth of design knowledge that can be brought to bear on the problem being faced by the design agent through the use of effective design support systems.

Explicit and tacit knowledge in design

In his book “The Nature of Knowledge”, Kemp [3] in answering the question “What is knowledge?” concludes with some certainty that it is difficult and probably impossible to find a definition of knowledge which would be universally accepted by all philosophers. Nevertheless, he defines knowledge as:

- Firstly, what a person believes and states as his belief (or what people collectively believe and state).
- Secondly, the extent to which statements are to be regarded as true depends on the extent to which these statements are true; the judgement of the truth of a statement being based on the extent to which it is compatible with other statements that are believed to be true.

Polanyi [131] makes a distinction between tacit knowledge and explicit knowledge. Tacit knowledge is personal, context-specific and therefore hard to formalise and communicate. Explicit or “codified” knowledge, on other hand, refers to knowledge that is transmittable in formal, systematic language. Polanyi contends that human beings acquire knowledge by actively creating and organising their own experiences. Products are physical manifestations of knowledge, and their worth largely, if not entirely, depends on the value of the knowledge they embody [132]. The design of products can be viewed as the application of scientific and technological knowledge to meet a perceived need. Most complex designs today require the integrated application of different disciplines of engineering and technological knowledge in order to meet the increased demand on functionality, quality, durability and environmental safety and product life cycle issues. The various aspects of the design must be analysed, evaluated by a team of designers (or agents) using their respective domain and experiential knowledge. For a given need, different teams of designers with a varied profile of design experience may design a diverse range of products using the same or different scientific principles and technologies. Hence, whilst scientific and technological knowledge are explicit knowledge, design knowledge

tends to belong to Polanyi's tacit knowledge, which is subjective, context-specific and hard to formalise. Nevertheless, designs can be characterised by some invariant properties, namely function, structure and behaviour. Therefore design knowledge can often be further categorised to functional knowledge, structural knowledge and behavioural knowledge. Since design knowledge is derived from teleological⁹ knowledge, scientific and technological knowledge, there is a mapping between the explicit scientific knowledge and the tacit design knowledge. Figure 4.2 shows the mapping between the two types of knowledge.

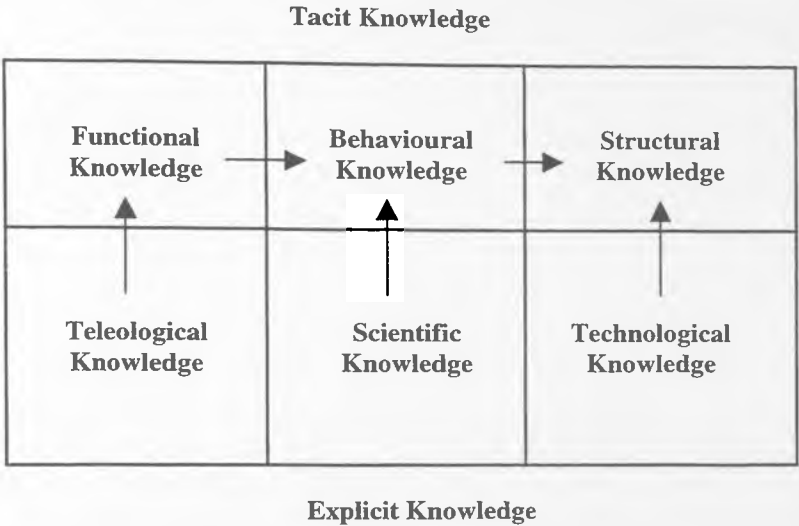


Figure 4.2 Mapping between tacit knowledge and explicit knowledge in design

The design process is the vehicle that designers used to transform explicit knowledge to tacit design knowledge. In this thesis, the knowledge transformation that occurs during the design process is the basis for understanding learning in design.

Nonaka and Takeuchi [132] suggest that tacit knowledge should include cognitive and technical elements. The cognitive element is based on what Johnson-Laird [133] calls “mental models”, in which human beings create working models of the world by making and manipulating analogies in their minds. Mental models, such as schemata, paradigms, perspectives, beliefs and viewpoints, help individuals to

⁹ Teleological knowledge is defined as knowledge of purpose of things in the world.

perceive and define their world. On the other hand, the technical element of tacit knowledge includes concrete know-how, crafts and skills.

In general, there are three types of tacit knowledge which broadly can be classified as declarative knowledge (know what), procedural knowledge (know how) and causal knowledge (know why). In the domain of engineering design, the three types of knowledge have been commonly referred to as design object knowledge, design process knowledge and design rationale knowledge. Collectively, it is also known as experiential knowledge. Researchers in psychology (Anderson [134], Cohen & Squire [135], Schacter [136]) have made a distinction between declarative and procedural knowledge. Declarative knowledge is explicit knowledge that we can report and of which we are consciously aware. Procedural knowledge is knowledge of how to do things, and it is often implicit. Such implicit procedural knowledge is often referred to as skill. However skill-type learning that results in implicit procedural knowledge will not be considered here as it is difficult to articulate the knowledge although it is an important facet of human learning.

The design object knowledge relates to the invariant elements of a design namely, the functional knowledge, the behavioural knowledge and the structural knowledge of a design. The design transformation knowledge refers to the mappings between the invariant elements of the design. Yan [137] identifies four types of design transformation knowledge or knowledge change:

- From function to structure ($F \rightarrow S$) which results in design catalogue type knowledge.
- From function to behaviour ($F \rightarrow B$) which characterises the designer's intent.
- From behaviour to structure ($B \rightarrow S$) which characterises the generative knowledge used in design synthesis.
- From structure to behaviour ($S \rightarrow B$) which characterises knowledge of design performance from which one can infer whether the purpose of the design is met.

4.1.3 Summary

Learning in design describes a phenomenon that occurs in the context of design (i.e. situated learning). Here learning is seen as being inextricably linked to and outwith the design process. As the designer constructs the various design situations through design actions, there arise opportunities for learning knowledge of the evolving design and about the process which is referred to as tacit or experiential knowledge. Learning in design encompasses situations of learning where knowledge is gained in retrospect from past design cases. Learning in design can result in a gain in experience and knowledge that can be applied to new design problems.

4.2 The requirements of a model of learning in design

Having described what is the nature of the phenomenon of learning in design and the nature of the experiential knowledge gained, it is necessary to discuss the nature and manner by which the model was to be derived in order to fulfill the objective of this research work.

4.2.1 The nature of the envisaged model of learning in design

From Section 1.3.1, it has been established that learning is a cognitive process in which there is a gain of knowledge that is recorded in some permanent memory. Persidis and Duffy [35] suggest that the learning process can occur in three distinct ways:

- Acquisition: the process of receiving new knowledge.
- Modification: the process of altering existing knowledge
- Generation: the process of creating new knowledge from existing knowledge.

For learning to occur, there is an underlying basic process of acquiring new knowledge that has to be stored in some permanent memory. The acquiring of the new knowledge could be through acquisition, modification or generation. Figure 4.3 shows a model of learning that incorporates the knowledge acquisition process and its integration into memory.

Figure 4.4 presents a model of learning in the context of design. This model takes into account the context of learning in design (Section 1.3.1). For each design

situation, the designer uses his design experiential knowledge to change the current state of design into a new state. As a result, the new design knowledge consisting of the output knowledge and the knowledge change can now be a subject of learning. The new knowledge is acquired or integrated into the memory through the process of acquisition, modification or generation.

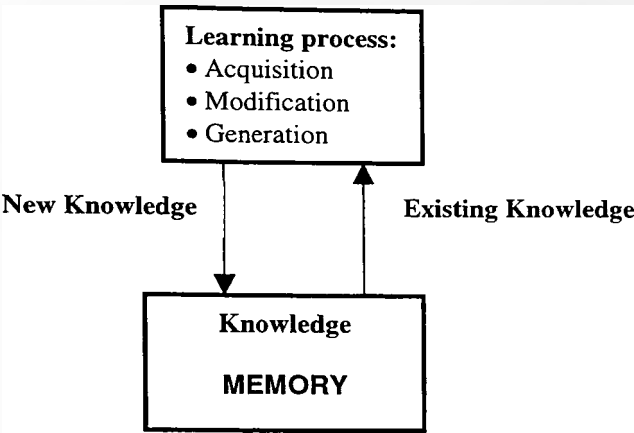


Figure 4.3: A model of learning

The model presented here, however, cannot answer the epistemological questions, (raised in Section 1.3.1), in a conclusive or comprehensive manner. It does not explain what is the nature of the design knowledge learnt nor does it explain the nature of the knowledge change that occurs during the design situation. It does not explain what is the goal of learning and when learning occurs. Furthermore, it does not describe in what manner learning is inextricably linked to design. The model however, suggests that learning in design can be dichotomised into two distinct cognitive activities, that of a design activity and a learning activity.

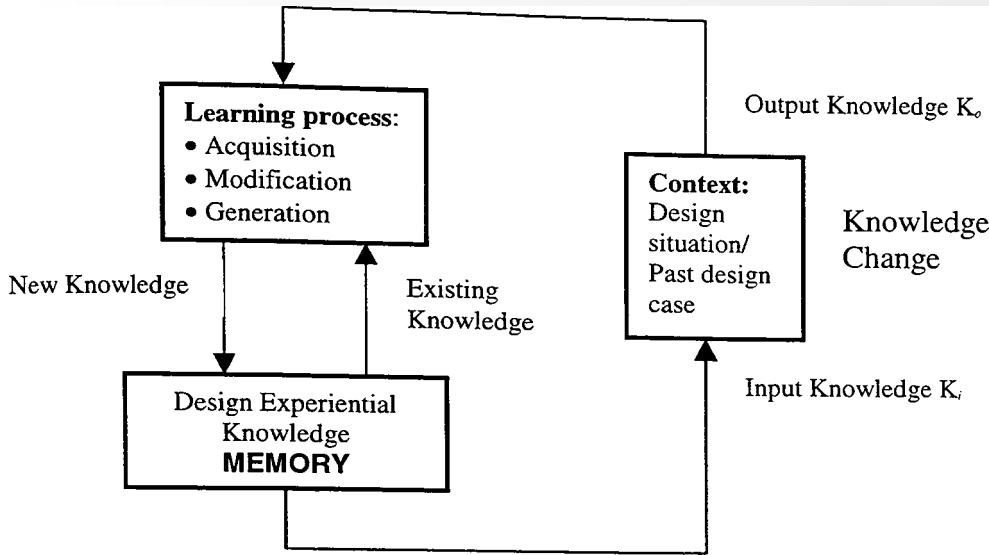


Figure 4.4: A model of learning in the context of design situation

4.2.2 Modelling of learning in design

In order to develop a model or theory of learning in design, there is a need to study learning in the context of the design situations that the designer creates in order to achieve coherence in the final design. The study is based on the assumption that there are regularities in the observed phenomenon of learning in design. These design situations can be observed in order to study and model the nature of the phenomenon of learning in design. Such a study is presented in the following:

- The use of past experiential knowledge in the various design situations as evidence that such knowledge has been learnt in the past and re-used in the design situations.
- To make inferences with regard to knowledge changes that occur and infer whether learning has occurred.

It is proposed that the phenomenon of learning in design be studied in two complementary approaches.

The first approach involves an assumption that it is possible to dichotomise the phenomenon into two distinct cognitive activities that can be studied. Firstly, there is a need to understand what the typical or generic design activities that designers

'use' to create design situations are and what is the likely knowledge change envisaged. This is the focus of Chapter 5. Secondly, a formalism for learning in design has been formulated through a review of existing work in learning and machine learning in design. In this formalism, the process that leads to learning can be categorised into various knowledge transformers and how these can then transform the input knowledge into new design knowledge. This is the focus of Chapter 6. Thirdly, a model of learning in design and the hypotheses regarding how the two cognitive activities are coupled or linked together is formulated in Chapter 7.

The second approach is to study the phenomenon in the practice of a design task by recording the protocol(s) of a designer at work and then to analyse and infer from the various design situations observed in the protocol the manner in which learning in design occurs. This approach is like performing a data directed search without any idea of what we are looking for except the notion that we are trying to find 'learning in design'. Hence an understanding afforded by the first approach gives a goal directed search for the phenomenon since the elements and relationships and the hypotheses that may constitute the phenomenon are used to analyse the protocol. This goal directed approach in analysing the protocol and evaluating the formulated model of learning in design is the focus of Chapter 8.

The two complementary approaches are based on the pragmatic scientism that has been advocated in Chapter 3 as the research methodology.

4.2.3 Summary

In this Chapter, learning is shown to be a cognitive process in which there is a gain of knowledge. Learning in design as a phenomenon has been described as the learning of experiential design knowledge in the context of a design situation. To derive a model of learning from the study of the phenomenon, two complementary approaches have been proposed. Because learning and the design situation are perceived to be inextricably linked, there is a need to make the assumption that it is possible to dichotomise the phenomenon into two distinct cognitive activities. Based on this assumption, the aim of the first approach was to derive formalisms for the design and learning activities and to formulate a model of learning in design based

on hypotheses regarding the manner in which these activities are inextricably linked. The second approach was to study the phenomenon of learning in design through the analysis of a protocol recording of a designer at work.

Chapter 5

A Formalism for Generic Design Activities

Design is a knowledge intensive cognitive activity. It has been shown in Chapter 4 that both designing and learning as complex cognitive activities can be described at the knowledge level. The design problem has been described as ill-defined and ill-structured i.e. uncertain, ambiguous and highly complex [138, 139]. Designers use various design activities to solve the design problem until design solution(s) is achieved. Here, a design activity is considered as a cognitive activity that applies domain and design knowledge to the evolving design thereby reducing/resolving the uncertainty, ambiguity and complexity associated with it. This Chapter therefore has three primary aims. The first is to identify and classify generic design activities based on the nature of the knowledge change that occurs as a result of these activities. The classification is based on the notion that the knowledge applied in these activities reduce/resolve the uncertainty, ambiguity and complexity associated with the design and/or the process. The second is to present a formalism to describe generic design activity at the knowledge level. Using that formalism, the third aim is to analyse the knowledge change that occurs in these generic design activities. It is important to identify, classify, represent and analyse these generic design activities in order to understand the knowledge change that occurs in these activities and by coupling them to the learning activity (see Chapter 7) seek to present a model/theory of learning in design.

However it is not the aim here to be exhaustive in trying to identify all the design activities. Neither is it necessary to be absolutely precise about the definition of each of these design activities as the subsequent analysis shows that there are different understanding by different researchers what each of these activities are. Furthermore the understanding of the phenomena of learning in design can be studied by itself through protocol analyses of design practice or interviews of designers without the need of the analysis and the formalism presented here. However, the value of this Chapter, as the identification and description of design activities using a formalism, will provide the background knowledge by which the protocol recording of a

designer at work can be systematically and coherently analysed (See Chapter 8). Further, only enough activities need to be identified in order to provide a sufficient basis upon which to investigate learning in design.

5.1 Generic design activities

In the course of designing, designers will perform a range of cognitive acts namely perceptive acts, communicative acts, designing acts and learning acts. These acts can be described at many and different levels of abstraction. For example, perceptive acts would include seeing, observing, reading, listening etc. Communicative acts would include speaking, explaining, writing, sketching, drawing, gesturing etc. For designing acts, these could range from concept generation to analysing, calculating, dimensioning, measuring etc. In the context of this Chapter, the focus is on designing acts that result in learning of design knowledge. Therefore not all acts that the designer performs will be considered although it can be argued that learning could take place as a result of these acts. For example, in performing the act of calculating which is a rather common cognitive activity that humans perform, one can learn about the operations required in calculating. As far as design knowledge is concerned, such operations offer no increase in the design knowledge. In fact, the acts of calculating, comparing etc. are basic operations required in algorithmic design methods (finite element method [140], Pugh's method of concept evaluation [101], etc) which are based on scientific substantive knowledge. Therefore in identifying what these generic design activities are, the focus is on activities that would result in the increase of design knowledge.

5.1.1 Granularity of design activities

It has been shown that as with any concept, it is possible to describe design activities at a number of different levels of abstraction [108]. Hubka and Eder [141] describe designing as a rational cognitive activity that can be decomposed into smaller (design) steps, stages and/or phases. The level of abstraction is to match the nature of the design problem. Hubka [142] has derived a hierarchy of design activities. In this hierarchy, Hubka has used different names to describe the design activities in the

hierarchy (e.g. design operations, basic operations, elementary activities, elementary operations). Each activity in a given level comprises sub-activities in the next lower level. Included in the list of elementary operations are a variety of activities such as calculating, measuring, drawing, analysing and synthesising etc. Although the repeated performance of some of these activities may follow the “power law” leading to skill-based learning [37], the design activities analysed here are considered on the basis that they lead to an increase in design knowledge. Hence, activities such as measuring, calculating and comparing that appear in Hubka will not be considered in the context of learning in design. These are considered as relatively basic operations that support higher-level design activities through which learning in design may take place.

Both perceptive and communicative acts performed by designers will not be considered as they are primarily acts for information processing in the design process.

5.1.2 Identification of generic design activities

Because of the notion that a design activity should result in an increment in design knowledge there is now a need to identify which of the cognitive activities of the designers can be considered as generic design activities. Here in this Chapter, the aim to identify generic design activities stems from the requirement to show the knowledge increment that results from design activities.

The notion of the existence of generic design activities is derived from the study of three sources that describe the cognitive activities of designing. The three sources are:

- Different systematic models of various design processes: engineering design [142, 99], product design [77, 101], mechanical design [102, 103].
- Protocol analysis of design experiments in different design domains: architectural design [143], mechanical design [144, 145, 146, 147, 148, 149].
- Case studies of large complex electromechanical artifacts [150].

The list of generic design activities identified from the analysis of different systematic models of various design processes is shown in Table 5-1. This list has been further corroborated by analysing various protocol analyses and case studies. These activities are considered as generic because they occur in different design domains. Although the nature of design problems in each of the domains may be different, each of the activities identified has characteristics typical of that activity.

5.1.3 Criteria for classification of design activities

The design problem has been described as ill-defined and ill-structured and highly complex [1, 2]. The problem is ill-defined in that there is no definite formulation of the goals of the design problem, fraught by a lack of knowledge of the constraints and criteria that may influence the problem. The design requirements are often initially ill-defined because of changing demands from customers or clients during the design specification phase. Both parties, the client/customer and the designer are trying to explore and establish the viability of the proposals within each own perception of criteria and objectives. With the agreement of the design requirements, the design problem is left to be discovered and defined. Problem formulations are considered “wicked” by Rittel and Webber [151] in that they are often solution dependent; it is difficult to formulate a problem statement without being implicitly or explicitly influenced by the solution.

Many elements of the problem cannot be expected to emerge until some attempt has been made at generating solutions. Because the boundaries of possible solutions are not known *a priori*, it is possible that some aspects of the problem will remain either unrecognised or undefined for much of the design process. Simon [138] called such problems 'ill-structured' or they exhibit many degrees of freedom. The ill-structured nature prevails for most parts of the design process. The ill-structured property of the conceptual design phase gives rise to the search space being ill-defined since it is being formulated while exploration is in progress. Logan and Smithers [139] conclude that the fundamental objective in design is therefore to understand the structure of the problem, with a major part of the effort in design directed at structuring problems and only a fraction of this effort directed at solving them once they are structured.

During the initial phase, there are no definite criteria to evaluate a solution. The design problem space cannot be defined due to radical lack of information although a large base of potentially relevant knowledge may be available for its solution. Even the set of alternative solutions is by no means suitable to generate a product description.

The design solution space is highly complex in that there is often combinatorial ways of solving the problem, with each partial solution leading to more alternatives before one can arrive at the final solution. Each partial solution has to be evaluated before proceeding to the next solution. There is therefore a need to manage complexity of the design solution space by prioritising and ordering design tasks. Duffy [152] posits that the factors that contribute to design complexity are the artefact *being designed*, the design activities, the actors involved, the design decision making process, the aspects impinging upon design and the knowledge and sources used and generated. These factors are elements which make execution of the design process non-trivial.

The ill-defined and ill-structured nature of design has resulted in the uncertainty/ambiguity and complexity of design and its corresponding process in the following aspects:

- i) The *design requirements* are often ill-defined because of changing demands of customers or the need to challenge market competition.
- ii) The numerous and varied possibilities and potential *design solutions* to be generated and explored in order to satisfy the design requirements in all the phases of the design.

The *design goals, criteria and constraints* imposed on the potential design solutions change as the design solutions evolve. Initial goals guide the first steps of the design process and these are soon augmented with others that are evoked as the process progresses. Goals and constraints that are postulated are means by which to limit the potential solutions.

The interdependencies of related *design activities* in terms of preconditions (inputs)

and effects (outputs) must be identified so that the resulting design process is streamlined to meet design time constraints.

Designing is a process by which the uncertainty and complexity of the evolving design is being reduced and resolved. From the three sources (i.e. various protocol analyses and systematic models of design and case study), three dominant characteristics of the design process can be identified. Firstly, designers constantly use abstractions/concepts/models as a means to generate and manage the complexity of the design task and to reason from the intended functionality of the product to derive some form or structure. Secondly, the expected behaviour(s)/effect(s) of numerous generated designs have to be evaluated against some design criteria or requirements to derive some design solutions. Thirdly, the protocol analyses of both mechanical and architectural design suggest that designers do formulate specific but varied plans as they pursue the design process to completion. These specific plans or episodes involve activities of control and information assimilation similar in nature to management.

In general, designers have to contend with these characteristics of design:

- The ambiguity and uncertainty associated with the evolving design.
- The complexity of the design solution space due to the need to generate many solutions before arriving at some accepted solution.
- The time complexity involved in the management of numerous evolving design solutions that are initially ambiguous and uncertain in description and their related design activities.

By performing appropriate design activities in the design cycle, designers seek to overcome the ill-defined, ill-structured and complex nature of design problem. Each of these activities involves the application of design knowledge and domain knowledge to the problem to reduce/resolve the ambiguity/uncertainty and complexity associated with the design problem. Here the generic design activities identified are classified into three categories as follows (see Table 5.1):

- Design definition activities: these activities seek to reduce/resolve the ambiguity and uncertainty associated with the evolving design states.

- Design evaluation activities: these activities seek to reduce the complexity of the design solution space.
- Design management activities: these activities manage both the space complexity and time complexity of the design solution.

Table 5.1: Identification and classification of generic design activities from sources on design processes

Design activity	Hubka [142]	Pahl & Beitz [99]	Suh [77]	Pugh [101]	Ulrich & Eppinger [103]	Ullman [102]
Design Definition Activities (Function to Form/Structure)						
Abstracting	✓	✓	✓			✓
Associating	✓	✓		✓		
Decomposing	✓	✓	✓		✓	✓
Defining	✓	✓		✓	✓	
Detailing	✓				✓	
Generating	✓	✓	✓	✓	✓	✓
Standardising	✓	✓		✓	✓	
Structuring/Integrating	✓				✓	
Synthesising	✓	✓	✓	✓	✓	✓
Design Evaluation Activities (Form to Behaviour/Effects)						
Modelling	✓	✓		✓		
Analysing	✓	✓	✓	✓	✓	✓
Testing/Experimenting	✓	✓		✓	✓	
Evaluating	✓	✓	✓	✓	✓	✓
Decision making	✓	✓	✓	✓	✓	✓
Verifying	✓	✓				
Design Management Activities						
Constraining		✓				
Exploring					✓	
Identifying	✓	✓	✓	✓		
Information gathering	✓	✓		✓	✓	✓
Planning						✓
Prioritising						
Resolving						
Scheduling		✓		✓		
Selecting	✓	✓		✓	✓	✓
Searching	✓	✓			✓	✓

Design definition activities

Design begins with an abstraction - a mental model to satisfy some predefined societal need expressed in the form of functional requirements obtained through the voice of the customer or some perceived need by the designer. It ends with a description of the product or artefact in the form that it can be interpreted for manufacture. Design as a cognitive process is concerned with the reception,

processing and transmission of information/knowledge from the beginning of the design process to the end [99]. The nature of the design information/knowledge (both qualitative and quantitative) is transformed from abstract, vague and sparse in representation to accurate, consistent and voluminous as the design progresses. Table 5.2 gives a description of the nature of the design information/knowledge based on the various phases of design. Each of these phases are characterised by design definition activities. For example, in the conceptual design phase, the design activities of generating and associating of concepts are prevalent. In the embodiment phase, structuring and integrating of systems and sub-systems are the main design activities. In the detail design phase, the design is synthesised supported by activities of integrating, detailing and standardising. Hence, design definition activities have been classified based on the effect that the activities have in reducing/resolving the uncertainty and ambiguity associated with design; that is, there is a gain in knowledge of the design through an increase in information content of the design. Each resolution of the uncertainty and ambiguity changes the state of the design that result in an increment of knowledge of the design. Therefore, the primary goal of design definition activities is to reduce the design’s uncertainty and ambiguity during different design states.

Table 5.2: Description of design information/knowledge

Phase of design	Information type	Information characteristic
Design specification	Symbolic.	Sparse and vague.
Conceptual design	Symbolic, schematics, simple sketches.	Sparse, inaccurate incomplete, may be inconsistent and temporal. Many possible solutions.
Embodiment design	Symbolic, numerical 2-D graphic (layout).	Fairly precise, consistent. Possibly only one solution.
Detail design	Largely numeric. 3-D graphic.	Accurate, complete, consistent, voluminous.

Design evaluation activities

The aim of design evaluation activities in general is to reduce the complexity of the design solution space. Through activities such as evaluating and decision-making, infeasible or less optimal solutions are ruled out, hence leading to a reduction in the search space of the design solution. For example, before generating the design solution, the designer may analyse and evaluate constraints that influence the design.

By imposing constraints on the design, the feasible design solution space is defined. When design solutions are generated, analysing and evaluating these against a set of design criteria similarly limits the number of feasible solutions. Thus the design evaluation activities may result in some design solutions being considered as failed design solutions, some may be sub-optimal and some may be optimal. The knowledge increment arising from failed design or successful designs may be useful for future design considerations.

Design management activities

The design problem has been characterised as an ill-defined or wicked problem in that its solution is linked in such a way that in order to think about the problem, designers have to commit themselves to some sort of solution. In the cycles of committing to some sort of solution for some design problem under consideration, the designer has to constantly evaluate and decide what step to pursue next. So a set of design activities which is used constantly by the designer has to do with deciding what to do next in the design process.

Although the design problem is ill-defined, designers do formulate specific plans as to how the design problems or sub-problems should be solved. Protocol analyses reveals the existence of specific or localised plans to tackle closely related design goals (see Chapter 8). The need to control and manage the complexity of the design problem gives rise to various design activities. These activities include identification of needs and goals, planning of design activities to achieve goals and sub-goals, prioritising various tasks of the design project, searching of solutions for the design problem and selection of the best solution(s). These set of design activities that manage the complexity of the design problem, pervade throughout the design process. Therefore, the purpose of design management activities is to streamline the design process so as to meet design time constraints and/or to reduce the complexity of the design solution space. For example, the activity of prioritising is to ensure that design tasks that significantly impact on the outcome of the design solution, which in turn may influence other down-stream design task(s), are considered first. In the detailing design phase, the 'selecting' activity may be used to select a particular component to implement a design concept or a particular manufacturing process to shape the ultimate form of the product. The selecting activity is also used in the

decision-making process to select one or more design concepts to evolve the design further. Thus, the activity of selecting which affects both the *design definition activities* and the *design evaluation activities* is included under *design management activities*.

In general, these activities may result in an increase of knowledge in the management of the time complexity and space complexity of the design solution(s). These activities do not increase the knowledge of design *per se* but rather contribute to the knowledge of the design process. So the criteria in classifying the design management activities as one category of design activities is based on the contribution of knowledge increment to management of the complexity of the design process.

5.2 A Formalism for Generic Design Activities

Hubka and Eder define a design activity as a change of the state of information by means of a design method or methodology [141]. In Gero's model of design [153], design is considered to consist of a number of distinct activities that are carried out between different design states. In this model, it is the set of design activities that transforms an initial design state through a series of successive refinements into a final design state (See Figure 5.1). The initial design state is described by a set of design functions (F) and a related set of expected behaviours (B_e). The design activities transform the design in which the design structure becomes more clearly defined until finally a satisfactory structure is obtained (S) in which the predicted behaviours (B_s) approach the expected behaviours (B_e). The role of design activities in Gero's model is that of transforming the state of the design expressed in terms of its function, structure and behaviour.

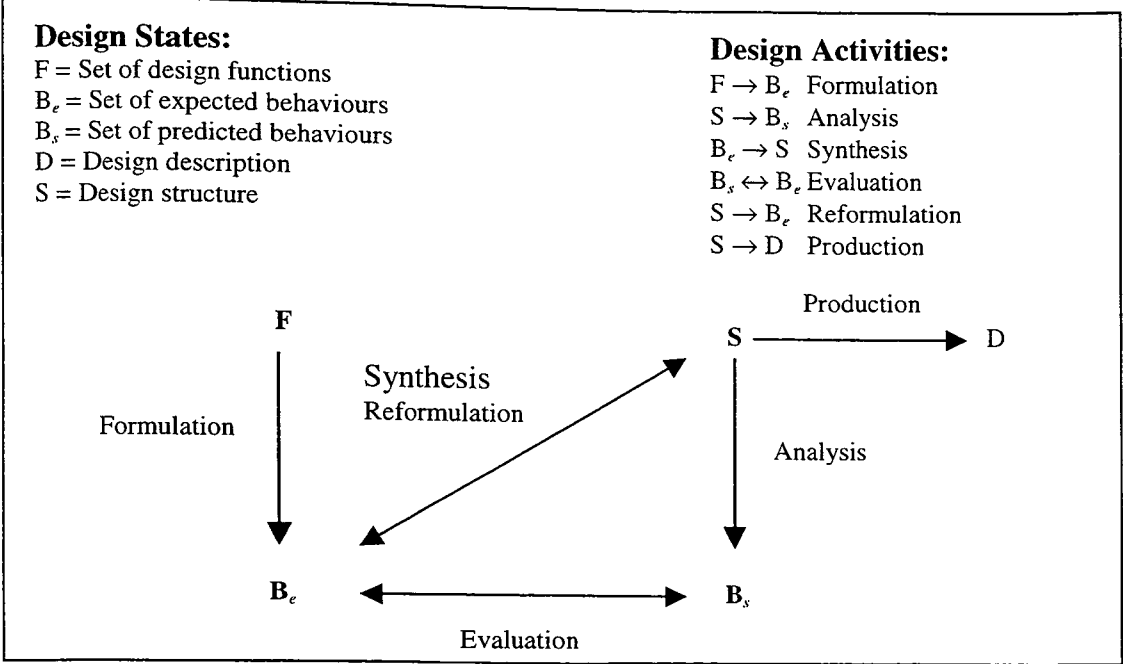


Figure 5.1: Gero’s model of design activities

It has been shown in Chapter 4 that design and learning activities as cognitive activities can be described at the knowledge level. A cognitive system at the knowledge level can be referred to as an agent [154]. At the knowledge level, cognition is described or explained in terms of goals, actions, knowledge and intended rational behaviour. Hence, the main entities with which an agent is concerned are goals, actions and knowledge (which include facts, beliefs, rules, laws, theories, and values).

The principle of rationality, the law of behaviour at the Knowledge Level by Newell [155], says that actions are selected to attain the agent’s goals. It is shown in Section 5.1 that the identification and classification of design activities has been based on their contribution towards an increase in the knowledge of the design or the associated process. Hence, here in this thesis, a *design activity* is defined as follows:

A design activity is an action or cognitive process taken by a design agent to achieve a knowledge increment in the state of the design and/or its associated design process in order to achieve some design goal.

While Hubka describes design activity in terms of a change of the state of information, Gero describes design activity as transforming the state of design in terms of function, behaviour and structure. The definition given in terms of knowledge increment is considered to be more generic than those of Hubka and Gero. Firstly, it includes the change of state of information which in the context of design is factual design knowledge. Secondly, it includes the transformation between design function, behaviour and structure as the classification of activities given in Section 5.1 groups design activities in terms of the transformation from function to structure and from structure to behaviour. Thirdly, the definition is generic because it includes the knowledge increment arising from the design process. Thus, it is reasonable to state that the definition of design activity in terms of knowledge increment associated with the design goal is a generic one.

A design activity is characterised by knowledge input. The inputs are the current state of the design, the resources in terms of people and knowledge (i.e. domain knowledge, heuristics, methods or techniques), tools, constraints and external requirements (e.g. regulations). The goal(s) of the design activity D_g determine the type of design activity to be performed. The output knowledge O_k from the design activity stems from the application of the input knowledge to enable the design to progress nearer to the design solution.

Given the above, the basic elements of a design activity may consist of:

- Existing design knowledge as Input knowledge, I_k .
- Design activity, D_a
- Design goal, D_g .
- Output knowledge, O_k

It is suggested that the basic elements of designing may be related as shown in Figure 5.2.

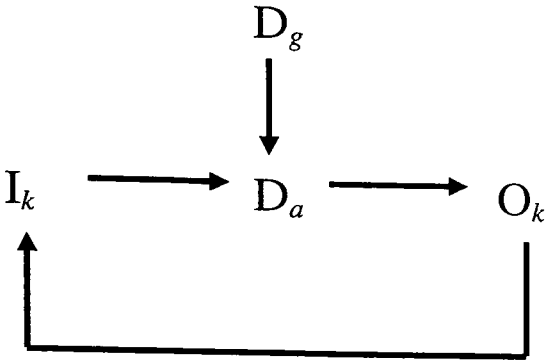


Figure 5.2: Formalism for a design activity

In this figure, design knowledge, domain knowledge and knowledge of the current state of the design serve as input knowledge to a design activity through which a new state of the design results and output design knowledge is modified or generated. The general goal of the design activity is to reduce the complexity of the design problem until a design solution(s) is finally achieved.

5.2.1 Input Knowledge

Design knowledge can be categorised into declarative knowledge (i.e. knowledge about the product) or procedural knowledge (i.e. knowledge of the design process) [156]. Past designs represent a wealth of product design knowledge. Design plans and/or design histories represent another source of valuable design process knowledge. The procedural knowledge becomes important when it is considered in management and control of design activities for a particular design task. However, for a design activity the main input knowledge sources would consist of the range of domain knowledge for a particular design problem, design methods, design mechanisms [157], design guidelines [158, 159], and design methodology [160]. These knowledge sources are basically experiential design knowledge learnt through a logical and rational understanding of procedures that have proved to be effective in expediting the design activity. They, therefore, represent experiential knowledge learnt from the design of similar products. But before they can serve as input knowledge, they are acquired as experiential knowledge learnt as a result of the design process.

5.2.2 Goal of Design Activity

Fricke [161] describes design as a goal directed process of solving problems. The goal of a design activity is to incrementally reduce/resolve the uncertainty, ambiguity and complexity of the design problem until the final design solution(s) is reached. In Sections 5.2, generic design activities are identified and classified in accordance to the knowledge increment associated with reducing/resolving uncertainty, ambiguity and complexity of the product and/or process. Here the goal of a design activity will be influenced by the nature of activity being considered. The nature and description of the goal for each of the generic design activities are given in the description of each activity (see Sections 5.3, 5.4 and 5.5).

5.2.3 Output knowledge

The output knowledge O_k from the design activity stems from the application of the appropriate activity upon the input knowledge to enable the design to progress towards the design goal and hence towards the ultimate goal, the design solution. The nature of the output knowledge is therefore dependent on the design activity and the evolving design solution. The nature and description of the output knowledge is given in the description of each activity (see Sections 5.3, 5.4 and 5.5).

5.2.4 Interactions of design activities

The design activities may progress in sequence (in series) or concurrently (in parallel) or coupled or as a compound activity (the grouping of related activities) or in iteration [162]. The output from one activity becomes the input into another activity. There could be more than one way to sequence these design activities which may be different for different design domains and for different types of design. Design strategy is used to determine the general direction of the design procedure. For some design domains the procedure become established with repeated use, resulting in what is known as design methodology. Design methodology therefore provides designers with knowledge on the design process. Included in this knowledge are: (a) models of the design and development processes, representing the structure of thinking and action in designing, (b) the methods and techniques used within these processes, and (c) a system of concepts and

corresponding terminology, [160].

5.3 Knowledge increment in design definition activities

In this section, generic design definition activities identified in Section 5.2 are described in greater detail to show the knowledge increment as a result of the design activity. Each activity is described in terms of the knowledge/information input into the activity, the design goal and the knowledge output as a result of that activity. Resources in terms of design methods/algorithms or procedures used in the activity are also given.

5.3.1 Abstracting

Abstracting as a generic design activity is to ignore what is particular or incidental and emphasise what is general and essential [99]. Pahl and Beitz consider abstracting as an activity to find higher level interrelationships (i.e. which are more generic and comprehensive) [99]. To Hoover et al. [163], 'abstractions are more than simplifications of behaviour and form; they are the result of cognitive decisions to ignore classes of behaviour and portions of the design object. Useful abstractions do not ignore classes of behaviour or parts of the design object but must capture the important relations between behaviour and form to help make good design refinements'. As a result, designers use abstractions and models to focus on various characteristics of a design and to simplify the complex relationships between function and form, and form and behaviour. The most useful abstractions are those which enable a designer to understand and predict the relationship between a required behaviour and the physical attributes of the design object. Both Goel and Porolli [164] and Ullman et al. [146] have hypothesised that abstractions are required because of cognitive limitations and problem complexity. Therefore, the purpose of this activity is to reduce complexity [99].

Designers use various kinds of abstraction depending on the degree of evolution of the design from conceptual to detail design. In the conceptual stage, the abstraction could well be just sketches, diagrams or data structures, etc. In the analysis stage, the abstraction will be in terms of some models such as geometric model, finite-

element model or spatial models or flow models. It is therefore not surprising that the use of abstractions has been documented in the protocol analyses of various design processes [143, 146, 147]. Here a distinction is made between abstraction and models and their related design activities (i.e. abstracting and modelling). Abstraction is seen as a means by which designers generate design solutions. Models serve as a more complete, well-defined and compact representations of the abstraction [165] by which designers communicate, control and predict the performance of the design solution.

In summary, Table 5.3 shows the input knowledge, the output knowledge, the goal of the design activity of decomposition and the knowledge change.¹⁰

Table 5.3 Abstracting – details of the elements of the activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_g	Knowledge change
• Types of abstraction • Domain knowledge	• Appropriate abstractions of design object (e.g. sketches, schematics).	• Reduce complexity of design object.	• Knowledge abstraction(s) of that depict useful relationships of the evolving design concept(s)

5.3.2 Association

Designers relate ideas and concepts through the cognitive activity of association. *Association* groups related ideas together giving rise to a perception which usually remains fixed unless being revised by subsequent activity [166]. Association theories consider association as a process within which links (associations) between contents of thought are established [160]. For example, a picture comprising a round shape and a red colour is first registered and then associated with the concept of a tomato. This hypothesis was supported by Ash [167] from his observations that the formation of pictures is responsible for the construction of the mind by perception. Links that are new or novel are considered as indication of the designer’s creativity.

The best known method to generate flood of ideas and concepts through association is brainstorming [168]. Brainstorming stimulates the production of strongly divergent associations by encouraging a number of people to work together in a

¹⁰ Similar statements will not be made for remaining design activities.

Table 5.4: Associating – details of the elements of the activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_g	Knowledge Change
• Domain knowledge • Methods for generating ideas (e.g. brainstorming, Gallery method).	• Ideas, concepts and their links.	• Generate as novel or new ideas/concepts as possible through association of ideas/concepts.	• Knowledge of novel ways in which ideas/concepts are linked.

5.3.3 Decomposition

Decomposition is a common problem-solving activity whereby a complex problem/object is solved by first breaking (i.e. decomposing) it into a set of smaller problems that can be easily handled [169]. Thus, decomposition can be defined as a process that breaks down a task/problem/object into a set of independent entities. For design of complex products involving hundreds/thousand of parts/components, the complexity of the design problem is simplified by decomposition for the purpose of focussing.

The key question is ‘on what criteria should decomposition be based?’ Challa and Fadel [170] identify three major ways of decomposition. They are object decomposition, aspect decomposition and sequential decomposition. In object decomposition, the problem is decomposed into sub-problems based on the physical/structural aspects of the objects involved. Aspect decomposition deals with the decomposition of a problem into the various aspects of knowledge or disciplines involved. The sequential decomposition decomposes the activities of a problem into groups and orders them in sequence with feed-forward links between them so that the design problem iterations are reduced to a minimum. For near routine design and parametric design, Liu and Brown [218,219] identified decomposition factors (bases for decomposition) using the knowledge compilation mechanism, [219]. The decomposition knowledge identified is design object knowledge, functional knowledge, design cases, design heuristics, general problem-solving knowledge and domain knowledge.

Here, decomposition is categorised into decomposition using existing

product/structure configuration, decomposition by function and decomposition of tasks. The first two categories relate to the design itself while the third category is concerned with the design process.

Decomposition using product configuration

The decomposition of a design problem is dependent on the types of product being designed. If the product is a repeat or a variant design, the basis for decomposition is the product configuration(s) of existing similar products. This approach will expedite the design process especially for complex products involving hundreds/thousands of parts/components, subsystems and systems. Therefore, Hashim [173] et al. define design decomposition as the resolution of a product into design entities¹¹ and the analysis of these and their functional relationships.

Decomposition of functional requirements

For innovative products, the decomposition knowledge of the product structure or configuration is usually non-existent and the design agent will have to rely on knowledge related to the overall function of the product. The basis of decomposition of the overall function may be governed by the key disciplines contributing to the solution of the design problem. For example, in mechatronics, a discipline-oriented decomposition is applied to facilitate the parallel development of the mechanics, electronics and software of the product.

Most systematic design processes [77, 99, 103] use decomposition by function as a basis for developing the design without reference to existing or known designs.

Function-oriented decomposition

Function-oriented decomposition relies on the notion that a design has a main function that can be broken down into its constituent sub-functions or partial functions¹² [99] in order to reduce the functional complexity of the design problem. The purpose of this decomposition is to identify the necessary supplementary functions to achieve functionality of the composite system. Several authors of

¹¹ A design entity is either a product, an assembly, a part, a component or a feature [173].

¹² In Hubka's hierarchy of functions, partial functions would include auxiliary functions, driving, propelling or energy delivery functions, regulating and controlling functions, connecting and supporting functions.

systematic design process (e.g. Pahl and Beitz [99], Ullman [102], Ulrich and Eppinger [103] advocate functional decomposition based on understanding the design problem in terms of flow of types of energy, material and information. Suh's axiomatic approach to engineering design [77] relies on the mapping of functional requirements (FR) to design parameters (DP). To satisfy these requirements, a physical embodiment characterised in terms of designs parameters must be created. Therefore decomposition is seen as a mapping process from the functional space to the physical space to satisfy the designer's specified FRs such that both functional requirements (FRs) and design parameters (DPs) can be decomposed into a functional hierarchy and physical hierarchy respectively.

In another approach, the function-oriented decomposition of a mechanical system involves the identification of the functional elements in the system. This is based on the law of Hubka that states that there exist causal relationships between functions and means of a mechanical system. The function is achieved by a means or a working principle(s). Based on Hubka's law, a mechanical system can be modeled by the function-means tree. A means may require one or more sub-functions. This recursion of mapping function to means and means to function will generate the function-means tree.

Umeda et al. [174] view functional decomposition as decomposing a function into networks of sub-functions that is composed of either abstract-concrete relations or whole-part relations. They distinguish functional decomposition into causal decomposition and task decomposition. Causal decomposition results in sub-functions whose behaviours are causally related to each other by the relationship *caused-by*. Task decomposition results in sub-functions whose behaviours can occur independently. For example a required function "to charge drum" in the design of photocopiers can be task-decomposed into two sub-functions "to discharge electricity to drum" and "to rotate drum".

Table 5.5: Decomposing – details of the elements of the activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_e	Knowledge change
• Knowledge of functional requirements of new design. • Various domain knowledge.	• Knowledge of key disciplines involved.	• Reduce complexity of design knowledge involved or lack of it.	• Knowledge of key disciplines for a design or class of design.
• Knowledge of functional requirements of new design. • Existing product structure in terms of part/sub-part relationship, system/sub-system relationship.	• Specific taxonomy of new design in terms of part/sub-part relationship, system/sub-system.	• Reduce multi-component complexity.	• Knowledge of function sub-function, part/sub-part or system/sub-system hierarchy of a product.
• Knowledge of function requirements. • Existing knowledge of function means mapping, function component mapping.	• Specific taxonomy of complex function in terms of sub-functions mapping, function means mapping, function component mapping.	• Reduce multi-functional complexity through decomposition of overall function.	• Knowledge of overall function in terms of sub-function mappings, function component mapping.

Decomposition of tasks/activities

Associated with each design project is a document reflecting a project plan which defines the tasks that need to be completed in order to solve the design problem [102]. That is, the design problem is decomposed into tasks and sub-tasks. Each task/sub-task with definite goal(s) and time constraints are assigned to appropriate design agent(s). Hence a design task represents a design effort that must be performed in order to achieve key milestones in a design process. A task or sub-task may comprise a sequence of interrelated activities. Task decomposition simplifies the design process and allows one to determine a potential group of tasks that might be performed simultaneously.

These tasks are organised to decrease the product development life cycle by analysing the interdependency between design activities. Kusiak and Wang [175] represent the relationships among the design activities using an activity-activity incidence matrix and the corresponding directed graph. By analysing the nature of the incidence matrix, design activities can be categorised into uncoupled matrix, de-

coupled matrix and coupled matrix. Kusiak and Wang develop a triangularisation algorithm that minimizes the number of non-empty elements in the upper triangular incidence matrix. A lower triangular matrix (i.e. matrix with zero elements in the upper triangle) implies de-coupled activities leading to reduction in the number of iterations in the design process. The details of the triangularization algorithm are given in Kusiak and Wang [175].

Using the incidence matrix to represent relationships among tasks, Kusiak and Wang [176] presented a methodology for decomposition of a design task resulting in minimum interaction density among groups of design tasks. The interaction density is defined as a measure of dependency among groups of tasks. Cluster identification algorithm is used to recognise groupings among tasks. Two types of dependency, serial dependency and interdependency are investigated. The knowledge of the interdependency of related tasks and their sequencing will enable the management of the design project in assigning of appropriate resources for timely completion.

Table 5.6: Decomposing of tasks/activities – details of the elements of the activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_k	Knowledge Change
• Knowledge of interrelated activities and their precedence orders. • Knowledge of algorithms for clustering activities (e.g triangularisation algorithm by Kusiak and Wang).	• Knowledge of sequencing of decoupled and/or coupled activities.	• Maximise decoupling of design activities into tasks/sub-tasks so as to reduce design iteration(s). • Minimise information flow between activities.	• Knowledge of how to sequence a interrelated set of activities into tasks/sub-tasks to reduce design iterations for a given design project.

5.3.4 Generating

Generating concepts or concept generation is a generic design activity to satisfy function requirements that are usually derived from customer requirements or a perceived need. The design activity closely associated with concept generation is concept association and/or concept combination. Individual concepts are generated for each function in the function sub-function hierarchy. To derive the design solution, these individual concepts must be combined into a single design solution. This results in many possible design solutions. Concept generation is therefore a

compound design activity in that it is composed of the activities of generating and combining design concepts. The goal of generating as many potential individual concepts is to satisfy specific sub-functional requirements while the aim of combining these individual concepts is to fulfill the overall functional requirement. Hence the purpose of generating concept(s) is to derive as many ideas or concepts as possible so as to accomplish the functional requirements.

As the information available in concept generation is limited, inexact and incomplete, the concepts generated are usually described qualitatively rather than quantitatively. Concept generation involves the process of associating (mapping) function (F) to structure (S) (i.e. $F \rightarrow S$ mapping) for routine designs or associating function (F) to behavior (B) to structure (S) (i.e. $F \rightarrow B \rightarrow S$ mapping) for non-routine designs. Prior to concept generation, the functional requirement(s) expressed in the form of a functional structure or hierarchy is obtained through the function decomposition activity.

Concept generation by use of morphology charts

The use of a morphological matrix or chart [99], [160] represents an example of $F \rightarrow S$ mapping. A morphological chart sets out the complete range of elements, components, or sub-solutions that can be combined together to generate solution(s). The chart consists of a grid of empty squares and is constructed from past design cases. On the extreme left column are listed the essential features or functions of the product. Across each row of the chart are entered the appropriate lists of sub-solutions or means of achieving the functions. When it is finished, the morphological chart contains the complete range of all the theoretically-possible different solution forms for the product. The complete range of solutions consists of the combinations made up by selecting one sub-solution at each time from each row. Of these combinations, some may be existing solutions; some will be feasible solutions and possibly a great number will be impossible solutions.

Concept generation ($F \rightarrow WP \rightarrow S$)

The concept of mapping function to structures via different “working principles” (physical principles and phenomena) is shared by several researchers (Pahl and Beitz [99], Freeman and Newell [177], Kota and Lee [178]). Each function may be

decomposed into several sub-functions; this decomposition continues until each sub-function is mapped to a solution principle or structural building block. Pahl and Beitz [99] define a solution principle as a combination of physical principle and main form design features (surfaces, motions and materials). Kota and Lee postulate that the level of detail or the extent of originality that is required for a given design dictates how far down the function decomposition tree should be traversed before one can map a sub-function to a structure.

Concept generation ($F \rightarrow DP \rightarrow SF$)

According to Suh [77], both functional requirements (FRs) and design parameters (DPs) can be decomposed into a functional hierarchy and physical hierarchy respectively. Functional requirements are the designer's characterisation of the perceived needs for a product. To satisfy these requirements, a physical embodiment characterised in terms of design parameters (DPs) must be created.

The FRs at the i th level of the hierarchy however cannot be decomposed into the next level of the FR hierarchy without first developing the solution in the corresponding physical domain hierarchy in terms of the design parameters that satisfies the i th level FRs. At each level of the two hierarchies, concepts to satisfy the FRs must be generated. By switching between the two corresponding hierarchies to generate concepts and decomposition into lower levels of the two hierarchies, the mapping between functional requirements and the corresponding design parameters can be derived in a hierarchical manner. But this mapping process is non-unique as more than one set of DPs could satisfy a given FR. Suh introduces two axioms by which one unique mapping can be selected in the end.

Concept generation ($F \rightarrow B \rightarrow S$)

Several researchers (Welch and Dixon [179], Bozzo and Fenves [180], Qian and Gero [181]) have favoured the $F \rightarrow B \rightarrow S$ mapping approach to concept generation. As quantitative or detailed information may not be available during the concept generation, these researchers use qualitative physics by DeKleer and Brown [182] to reason about the transformation between $F \rightarrow B$ mapping and between $B \rightarrow S$ mapping.

Welch and Dixon consider concept generation as a two-step process. In the first step known as phenomenological design the functional requirements are transformed into a behavioural description. This is followed by the embodiment step in which physical artefacts are matched to this behaviour. The two-step process overcomes the problem that designers become fixated to familiar design solutions that morphology charts and function to structure mapping encourage [179].

To facilitate behavioural reasoning, Welch and Dixon developed a new representation derived from bond graphs called behavioural graphs. A behavioural graph is a directed graph where the nodes include both the functional parameters and the mapping between the parameters. Behaviour is represented as a mapping between three parameters (i.e. input parameter, the output parameter and optional parameter). Each attribute of the input parameter is mapped to the corresponding attribute of the output parameter. Welch and Dixon developed a set of fourteen primitive behaviour mappings that represent physical principles and phenomena based on the generalised principles of bond graphs.¹³ Welch and Dixon specify two rules¹⁴ for combining these 14 primitives into behavioral graphs.

Phenomenological design generates one or more behaviour graphs to meet the functional requirements. This is done by replacing each edge of the function graph (each dependency between two functional parameters) with a sequence of primitive behaviour mappings that successfully map the input parameter attributes to the corresponding output parameter attributes. The results for each edge of the function graph can then be combined (in all possible combinations) to form a set of initial behaviour graphs using the two rules. Computationally, the generation of the behaviour graphs is limited by the knowledge of embodiments stored in the library of embodiments.

¹³ There are 10 main primitive behaviour mappings which are derived from 5 basic bond graph elements (resistance, capacitor, inductance, gyrator and transformer), each having 2 possible causal assignments (effort input or flow input). There are 4 primitive behavior mappings for signal tap mappings.

¹⁴ Firstly, a mapping that outputs an effort parameter must be connected to a mapping that inputs an effort parameter, and correspondingly for flow parameters. Secondly, the input and input parameters of a mapping must be of the same class: both signals or both powers. Signal and power class parameters are related either by using a signal to modulate a mapping between power parameters or

Embodiments are represented in terms of their behaviour, constraints on that behaviour and evaluation information. In general, embodiment behaviour can be represented by a complex behaviour graph, indicating not only the primary but also secondary or side-effect behaviours. Embodiment constraints describe limitations or restrictions on the attribute values of the input and output signal parameters associated with the primitive behaviour mapping representing the embodiment behaviour. For each primitive behaviour mapping, embodiment design searches its predefined library to find all the behaviour that is described by the same mapping. This set is then culled by checking the constraints of each embodiment against the attributes of the connected parameters to the mapping in the behaviour graph. If the constraints are satisfied the embodiment is considered acceptable and used as part of a system design. A set of different system configurations is formed using all possible combinations of the acceptable embodiments that collectively meet the behaviour specified by the behaviour graph.

In the Qian and Gero's approach of design by analogy, the design variables such as function, behaviour and structure variables are obtained from an existing design. From this known design state space an analogous design in terms of function, behaviour and structure variables can be retrieved. By mapping the corresponding function, behaviour and structure framework between the existing design and the analogous or source design, useful information is transferred by analogy to the current design to modify the existing state space. The goal of the analogy-based design is to find structures and external effect mappings (in terms of design variables) through function or behaviour mapping. If two functions match, their associated mapping for behaviour and structure plus external effects can be found. If two behaviours match, only structure plus external effect mapping can be obtained. The mapping process aims to find structure correspondence between the source and the target design in terms of four design variables: element, attribute relation and process. Qian and Gero suggest a set of mapping rules through which design knowledge is transferred from source to the target design.

by using a signal tap mapping to extract a signal from a power flow.

Table 5.7: Generating – details of the elements of the activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_g	Knowledge Change
• Knowledge of function/sub-function hierarchy. • Design catalogues which map function to solution principle/component.	• Examples of knowledge of function to solution principle/component mapping.	• Generate as many solutions principle(s)/component(s) that satisfy function hierarchy.	• A mapping of function/sub-function hierarchy to solution principle/component for a given design problem is derived.
• Knowledge of function to solution principle mapping/structural building block. • Knowledge of function/sub-function decomposition	• Knowledge of $F \rightarrow WP \rightarrow S$	• Generate as many solutions principle(s)/component(s) that satisfy function hierarchy and the solution principles.	• A mapping from function to working principles to structure is derived.
• Knowledge of function to design parameters and design parameters to structural forms. . • Knowledge of mapping between function hierarchy and physical hierarchy.	• Knowledge of $F \rightarrow DP \rightarrow SF$	• Generate as many solutions principle(s)/component(s) that satisfy function hierarchy.	• A mapping from function to design parameters to structural forms is derived.
• Knowledge causal relationship between function and behaviour (i.e. $F \rightarrow B$ mapping). • Knowledge of rules of combination. • Knowledge of embodiments.	• Specific qualitative causal knowledge and relational knowledge of concepts in terms of $F \rightarrow B \rightarrow S$ mapping(s).	• Generate as many design solutions using causal reasoning. • Investigate new concepts based on new scientific discovery or new technology.	• A mapping from function to behaviour to structural forms is derived to explain the causal relationships among them.

5.3.5 Detailing

Detailing prepares the design for production in that the arrangement, form dimensions and material and surface properties of all the individual parts are finally specified, and all the drawing and production documents produced. The output from detailing is the complete manufacturing information, typically detail drawings, part lists and instructions for assembly, testing, adjustment, maintenance etc. [141].

Table 5.8: Detailing – details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_g	Knowledge Change
• Domain knowledge relating to the manufacturing, assembly and testing of product. • Design requirements.	• Detail drawings. • Documentation (e.g. design specification, assembly procedure etc).	• Remove all ambiguity and uncertainty related to the manufacture and assembly of the design.	• Knowledge of the design expressed in drawings or documentation for manufacture, testing etc.

5.3.6 Defining

Defining is a generic activity used throughout the design process [183]; from defining the purpose of the artefact, defining the design problem, defining design specification¹⁵, defining design tasks, to the details of defining relationships of concepts/components, and defining dimensions of design objects. The outcome of this activity provides information for downstream design activities and ultimately in creating the desired artefact [183]. But defining is often preceded by analysis and evaluation. For example, in order to define the design specification, customer surveys have to be analysed and evaluated. In defining which size and type of structural member to use, strength analysis is performed to derive the required moduli and different structural sections evaluated.

Table 5.9: Defining – details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_e	Knowledge Change
• Knowledge of design requirements. • Summary of design decisions made at the end of design iterations or milestones in the design process.	• Various statements of specifications (e.g. specification of requirements, product specification, interface specification, bill of materials).	• Definitive decisions representing milestones in the design process that has influence on down stream activities.	• Specifications that satisfy design requirements.

5.3.7 Standardising

Standardising is a special kind of selecting activity that is peculiar to the detail design phase. The choice is guided by the principle of uniformity provided functional and performance requirements are satisfied. Designers should attempt, as much as possible to utilise as many of the existing parts and components as possible in several places instead of using components of a special design in each portion of a product and in each product. Ulrich and Eppinger [103] refer to this as internal standardisation. External standardisation occurs when designer(s) use externally supplied component(s) and parts(s) that are commonly used by manufacturers of similar product lines. This will lead to standardisation of manufacturing and assembly processes. Standardisation has many advantages. Bralla [159] lists

¹⁵ Other terms used are performance specification, product design specification, design requirements, engineering specification.

benefits such as elimination of development costs of new components, reduced start-up costs of equipment and machinery, reduced lead time, reduced tooling costs since tools are already available from previous manufacture of similar parts, higher production quantities leading to economies of scale. Just-in-time arrangements are made easier due to larger usage quantity and reduction in number of parts/components.

Table 5.10: Standardising - details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_g	Knowledge Change
• Knowledge of detail design of the product. • Knowledge of specifications components/parts of the product • Knowledge of classes of standard components	• A set of standard components selected. • Criteria or explanation for the standardisation.	• Minimise manufacturing time and cost of products	• Knowledge of nature of the standardisation and the basis for it (i.e. group rationalization based on similarity)

5.3.8 Structuring/Integrating

A product can be described in terms of functional and physical elements [85]. Functional elements are usually described in schematic form before implementing them in specific technologies, components or physical working principles. The physical elements are the parts, components and subassemblies that ultimately implement the product’s functions. The physical elements of a product are typically organised into several major physical building blocks called chunks. Because the chunks could interact with one another in many ways, to facilitate integration Ulrich and Eppinger [103] suggest that the identification of two categories of interaction between chunks, namely fundamental interactions and incidental interactions. Fundamental interactions are explicitly represented by the schematic showing the clustering of elements into chunks. Incidental interactions arise from the physical implementation of functional elements or because of the geometric arrangement of the chunks [103]. Identification of these chunks with important interactions becomes crucial for co-ordination and communication of group design activities. By using the interaction information early in the design process, the design team can choose an architecture that minimises the complexity of the co-ordination required to develop the system. Pimmler and Eppinger [169] present a methodology to cluster functional

elements into chunks based on quantification of interactions between the elements. Knowledge of the incidental interactions can be documented using schematic and interaction graphs or matrices, [103].

Table 5.11: Structuring - details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_g	Knowledge Change
• Knowledge of interfaces/interactions between parts, systems • Knowledge of specifications components/parts, systems of the product	• Knowledge product architecture in terms of chunks and their interactions. • Reasons for fundamental and incidental interactions.	• Optimal product architecture that minimise the complexity of co-ordination required to develop the total product/system	• Knowledge grouping system to system interactions (i.e. group rationalisation based on system similarity or functional dependency).

5.3.9 Synthesis

Several researchers (Ivezic & Garrett [184], Flemming et. al. [185], Alberts [186]) have described synthesis as a mapping of dependencies between function, behaviour and form. But synthesis is more than just a mapping of dependencies. Pahl and Beitz [99] describe synthesis as putting together of parts or elements to produce new effects and to demonstrate that these effects create an overall order. Synthesis as a design activity entails configuring entities of a domain to construct a realisable system structure that satisfies design requirements, [187]. Synthesis when applied to the designing of artificial products/systems entails the integration of parts/systems such that the physical laws of the domains involved when acting together in a given environment give rise to a desired behaviour and performance. Hall [188], in his book entitled “The Age of Synthesis” states:

“All of these systems¹⁶, whether small or large, involve complexity, interconnectivity, interfacing and integration that are the handmaidens of synthesis.”

Therefore, the primary aim of the synthesis activity is to integrate parts into a whole (components to assembly/sub-system, sub-systems into systems) so that the desired function of the design is achieved. In short, as Roozenburg and Eekels [160] put it, synthesis is aimed at the totality of the entity to be designed, not only at a specific aspect, but at the entirety.

¹⁶ Referring to inanimate objects made by human beings.

Synthesis through different levels of abstraction

Synthesis is also seen as an activity that progressively evolves the design from an abstract concept to a concrete entity in terms of its description. To allow for the description of a technical system at multiple levels of abstraction, Alberts [186] introduces the concept of technology-based layers or levels. At each of these technology-based layers, there is a collection of primitive generic system models; these elements are labeled primitive because they cannot be further decomposed into smaller units at that particular level.

The synthesis process involves the combination of primitive generic system models into more complex generic system models so that the behaviour of which should contain the function specified in the functional specification. Obtaining the required function from the behaviour requires determining the relevant part of that behaviour and the instantiation of corresponding variables and parameters with specific values specified in the function. The instantiation should further satisfy the constraints specified in the performance requirements. This process is repeated in the technology-based layers until a description of the technical system that can be physically realised has been obtained. This requires a translation between generic system models at different technology-based layers. The results obtained at a particular level, after translation, serve as a new and more specific design description at the next lower level.

In Alberts' model of synthesis [186], generic elements are defined as having the following characteristics:

- Explicitly represent the dependencies between function, behaviour and form of a product.
- At multiple levels of abstraction ranging from the original specifications to the actual realisation,
- Using an exact scientific and product-oriented approach.

Mechanisms for synthesis

Mechanisms to aid designers in the synthesis activity at different levels of abstraction have been developed by various researchers. These mechanisms are

amenable to be developed into computer software to enhance the capability of design support systems. These mechanisms include schematic synthesis by Ulrich and Seering [189], functional synthesis by Chakrabarti and Bligh [190]).

Schematic Synthesis

In the early stages of conceptual design where detailed descriptions of the design are not available, describing designs in terms of schematics has been a common practice in various domains of design (e.g. pneumatic circuit design, hydraulic circuit design, electrical circuits design, piping circuit design). A design described schematically consists of a specification of its constituent functional elements and their interconnection. The key distinction between schematic descriptions and other types of design descriptions is that schematic descriptions generally contain no information about the design's geometrical and material properties. Generating a schematic description in response to a specification of desired device behaviour is schematic synthesis. By focusing the initial problem solving effort on schematic descriptions the functional aspect of the problem is de-coupled from the structural aspects. Generating a schematic description without concern for a possible physical implementation, forces the design system to get the ideal device behaviour right first. This problem strategy is sound since only devices with correct schematic descriptions will meet the design specifications. Once the schematic description is right, the synthesis of an efficient physical implementation can proceed. For example, in an electrical network, synthesis is defined as the problem of constructing a network of basic electrical elements in such a way that the resulting behaviour of the network includes the particular function that the network must perform.

Ulrich and Seering [189] introduce what they called schematic synthesis problem and prescribe one of its solution techniques. The problem domain consists of devices that can be described as a network of lumped-parameters, idealised elements in the translational-mechanical, rotational mechanical, fluid-mechanical and electrical media. Design problems in this domain are characterised by an input quantity, an output quantity and the desired relationship between the input and output. The approach adopted by Ulrich and Seering is to first create a very modular physical description by direct substitution of physical elements for functional

elements and then to subsequently look for opportunities to simplify this design by using particular physical features of the design to perform more than one function.

In order to evaluate a schematic description with respect to a design specification, a system for performing design must be able to derive behaviour from schematic descriptions.

Functional Synthesis

Chakrabarti and Bligh [190] introduce the concept of functional synthesis based on functional reasoning. In functional reasoning, the design agent reasons at the functional level in order to generate solutions to specified design problems and to evaluate these solutions for their suitability to specified design criteria.

The approach involves:

- (1) Identifying a set of primary functional elements and rules of combining them and,
- (2) Developing appropriate representations and reasoning procedures for synthesising solution concepts using these elements and their combination rules; these synthesis procedures can produce an exhaustive set of solution concepts, in terms of their topological as well as spatial configurations to a given design problem.

The general rules of combination for a given system to be valid can be summarised as:

1. A set of structures can be connected at a common connection only by combining the outputs of one of its subsets with the inputs of the rest.
2. A connection is possible only when the input-output involved in the connection have the same characteristics.

With the system of connected structures in which the constituting transformations of each structure involved are known, it is possible to analyse that system to:

1. Check whether or not it is a valid system.
2. Identify, for a valid system, its transformations, i.e. the transformation between the characteristics of its input-output variables.

Chakrabarti and Bligh [191] advocate a method for synthesising these structures. The approach is to solve the design problem at each instance of time, following the order in which the I/O characteristics are enlisted above (kind¹⁷, orientation, direction magnitude and position requirements). By considering the problem of 'kind synthesis', the design solutions are synthesised based on I/O kinds required by the design problem. Each of the potential solutions is evaluated for orientation, thereby configuring the valid orientation of these solutions. These orientations are then checked for direction requirements, and the valid direction configurations are computed and retained. These solutions are now ready to be evaluated for the magnitude and position requirements, again allowing for the elimination of the infeasible solutions. In this manner the functional synthesis by Chakrabarti and Bligh seek to synthesise a sense of totality in the design.

Knowledge output as a result of synthesis

Knowledge of the design space

In order to configure and integrate the design, the knowledge gained as a result of the synthesis activity is the knowledge of all relevant aspects and concerns of the artefact that must be accounted for in the design. These aspects and concerns are expressed in terms of design properties and their relationships.

Knowledge of product configuration

A product configuration describes the set of physical chunks (parts/components, sub-systems/sub-assemblies) that comprise the whole system and the connections between these chunks. The knowledge of product configuration that results from the synthesis activity is the knowledge of the physical chunks and their connections, the local composition constraints that are applied to derive the configuration. The knowledge of product configuration becomes useful in the design of similar products.

Knowledge of relationships of design properties

Valid relationships between design properties of the product can be established and these become a reliable and useful source of product knowledge for design reuse.

¹⁷ Kind refers to the type of force transmission i.e. direct force or torque

Table 5.12: Synthesising- details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_k	Knowledge Change
• Knowledge of function to behaviour to structure mapping. • Knowledge of different level of abstractions. • Knowledge of the appropriate representation of abstractions. • Knowledge of relevant domains for different aspects of product.	• Knowledge of the design space. • Knowledge of product configuration. • Knowledge of relationships of design properties. • Knowledge of integrating physical building blocks/chunks.	• Totality in the design of product	• A final design that satisfies all the design requirements expressed in some design documents detailing all aspects of the design for manufacture.

5.3.10 Summary

The apex of the design definition activities is synthesis. Synthesis is considered here as a compound activity as it involves search and discovery of design solutions, composition and integration of these solutions. In mechanical design, synthesis is seen as a systematic construction of designs based on generic elements. According to Hansen [192], synthesis of a design from generic elements given a required function consists of:

- Decomposition of this function into sub-functions.
- Finding (assemblies of) elements that can perform these sub-functions (concept generation).
- Combining these (assemblies of) elements into a more complex assembly that implement the total function (concept combination).

The synthesis process iterates through these steps, which are interleaved with evaluations and the corresponding adjustments to the resulting design descriptions. Hansen’s notion of the synthesis activity however fails to capture the importance that inter-connectivity, interfacing, integration plays in the synthesis activity. To Hall [188], the building blocks of synthesis are parts, aggregation, connection, integration, system. Hence, Figure 5.3 represents the hierarchy of design definition activities in which synthesising is the result of generating and abstracting design concept(s) and structuring concepts to form a unity. Generating is supported by decomposing

known solutions and/or associating ideas or concepts and combining them into overall design concept(s). Feasible design concepts are refined through detailing where abstractions are made concrete through detailing. Detailing can be achieved by defining and/or standardising.

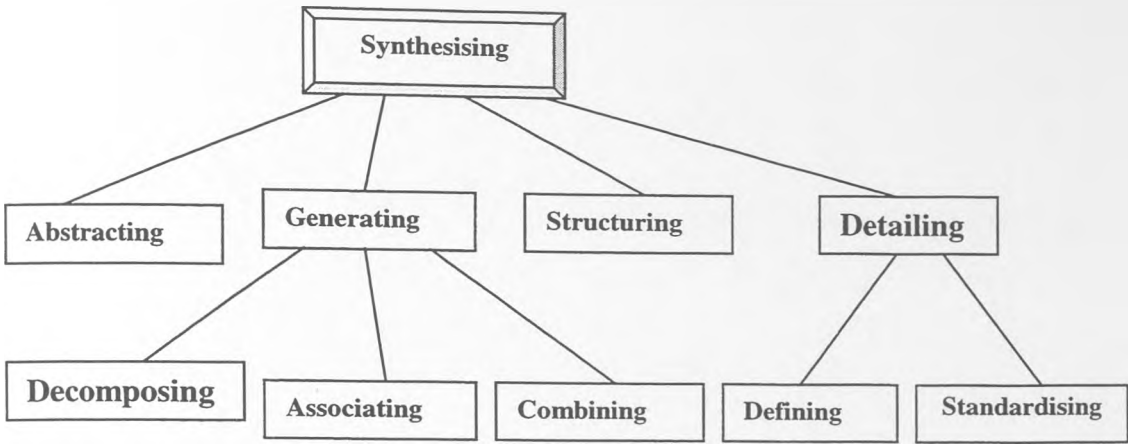


Figure 5.3: Hierarchy of Design Definition Activities

5.4 Knowledge increment in design evaluation activities

5.4.1 Analysis

Analysis as a design activity involves the use of models of physical phenomena to answer questions about an engineering product - typically about its behaviour. Gero [153] defines analysis as the means by which the behaviour of a design structure can be predicted.

Finn [193] considers analysis as consisting of three interrelated activities, namely: *modelling, simulation and evaluation*. Modelling involves reasoning about a design structure (or physical system) with the aim of abstracting an analysis model. This model provides the basis for the simulation activity, the mechanism by which qualitative or quantitative results that describe the behaviour of the physical system are obtained. Evaluation is the activity by which these results are verified with respect to the analysis model and validated with respect to the physical system [193]. For example, in Finite Element Analysis, the model of the structure to be analysed is first constructed of arrays of finite elements. The behaviour of the model subjected

to various loading under different environmental conditions is then simulated using the Finite Element Method (FEM) to calculate the stress-strain behaviour of the structure. The stress-strain behaviour can then be validated against a set of design criteria to determine its acceptability as a feasible design solution.

Brooke [194] et al. view the selection of a solution technique as being part of the analysis process. Equally, the construction of a mathematical model is seen as being part of the analysis activity.

Although one would normally associate analysis with quantitative aspects of the design, analysis encompasses both qualitative and quantitative aspects of the design. In fact, Finn [193] classifies analysis techniques into three categories, namely: qualitative techniques, approximate techniques and detailed techniques.

Qualitative Analysis

Qualitative analysis can be considered to be a cognitive or mental activity that is often augmented by qualitative notes and sketches. Qualitative analysis is generally motivated by the need for a behavioural or functional understanding of a physical system. The type of reasoning used during qualitative analysis activity can range from shallow surface reasoning using engineering heuristic knowledge and experiential knowledge to deeper reasoning based on engineering laws and first principles using qualitative reasoning.

Approximate analysis

In design, the amount and granularity of the information provided by qualitative analysis techniques can become inadequate in order to advance the design. In this scenario, approximate analysis becomes the core and more powerful analysis approach, providing additional quantitative information but avoiding the complexities associated with detailed analysis such as simulation.

Essentially, approximate analysis involves the use of engineering formulae and correlation (and all their inherent assumptions) to give quantitative estimations of the expected magnitudes of the phenomena being analysed. Approximate analysis often provides sufficient information to detect incompatibility between the expected and predicted behaviour. In approximate analysis, *a priori* qualitative reasoning to

derive a behavioural understanding usually forms the basis by which assumptions are made and the appropriate engineering approximations chosen and applied.

Detailed analysis

Detailed analysis techniques can generally be classified as numerical techniques that are carried out to give precise solutions at a fine level of granularity over a domain [193]. Detailed analysis can itself be carried out at a number of levels of granularity depending on the requirement of the analyst. As in approximate analysis, detailed analysis can provide sufficient information to permit a design to be reformulated should the need arise. The output is a set of data values that describe the required product attributes.

In order to perform an analysis, it is necessary to draw upon existing stores of information/knowledge about:

- Physical phenomena.
- Theories about these phenomena.
- Possible assumptions that might be made.
- Available solution techniques.

Table 5.13: Analysis - details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_e	Knowledge Change
• Knowledge of physical phenomena and theories. • Methods of analysis related to the physical phenomena • Knowledge of constraints, assumptions made and degree of accuracy required. • Knowledge of the structure/form of the design. • Knowledge of the working environment of the design (e.g. external loads).	• Knowledge of the behaviour of the design.	• The design behaviour meets the functional requirements.	• Knowledge that a particular design in terms of form displays behaviour(s) that meets design criteria.

5.4.2 Modelling

The activity of modelling pervades throughout the design process. Duffy et al. [90] view models as central to current design methodologies because models allow a decoupling of the experimentation process from the artefact itself, provide a means of communication, control and prediction of the performance of a design and most

importantly, serve as an abstract representation of the design. A model is therefore a *means* by which a designer represents some aspects of the intended product to focus their attention on. Designers can use many different models to reflect the state of the design. The models used are largely dependent on the perspectives or aspects of the design that they are focusing their attention on. A model starts out as some cognitive notion (a mental model) that is eventually developed into a concrete entity. Through different types of design activities, the mental model is transformed into various other models (functional model, behaviour model, geometric model, structural model). Finn [195] identifies different models that are required for the finite element analysis and interpretation of the results. A functional and behavioural model is needed to reason about the behaviour of a system or product. Physical modelling involves the application of modelling idealisations so that a representative physical model can be created. Mathematical modelling involves describing the physical model in terms of a set of partial differential equations over a geometric domain bounded by a set of boundary conditions. The modelling activities described by Finn [195] addressed the evaluation of the behaviour of the product or system when the structure or form of the product is known.

Barzel [165] states canonically that a model can be decomposed into an abstraction, a representation of that abstraction and an implementation of that representation. Abstraction describes the most fundamental aspect of a model in terms of a set of features and characteristics of the thing (product) being modelled. The concepts in the abstraction are usually informal, vague and imprecise. A representation is an approach that describes the model in some complete, well-defined and compact manner. For example, a scaled drawing of that model can be used to describe a geometric model of a product. Implementation represents the actual construction of the model in some medium(s) such as plastic, wood, etc. According to Barzel, the concepts of abstraction (A), representation (R) and implementation (I) constitutes the ARI structure of a model. The ARI structure of a model therefore gives a succinct overview and encapsulation of the model.

Qualitative Modelling

In the conceptual design phase, the modelling activities usually have to do with

mapping the design purpose of the product to that of their means in satisfying that purpose (i.e. functional modelling). To distinguish from the more concrete and precise nature of modelling used for analysis, the term abstraction or qualitative modelling is used to denote the kind of modelling used in conceptual design where precise and complete information is not available.

Qualitative modeling has been applied to the concept design of structures where the description of a structure is incomplete and imprecise. This lack of information has not prevented experienced engineers to reason qualitatively about the performance of a schematic structure based on physical principles and experiential knowledge. For example, during the conceptual structural design, an engineer generates structural schemes to transfer loads from structural members to the foundations. Fruchter et al. [196] developed a system called QLRS to assist structural engineers to perform qualitative modeling, analysis and evaluation of conceptual design.

Bozzo and Fenves [180] also develop a $(F \rightarrow B \rightarrow S)$ concept evaluation mechanism for conceptual structural engineering design, which corresponds to the selection and arrangement of the overall structural system. At the conceptual design stage the structural engineer is concerned with understanding the overall behaviour of a structure and identifying undesirable characteristics. However, there is not sufficient information at this stage to develop a mathematical model of the design and perform a quantitative analysis. Therefore, Bozzo and Fenves formulate a qualitative reasoning framework called the space-centred framework for reasoning about the behaviour of structural systems subjected to loads. This framework overcomes a serious limitation of existing frameworks when applied to structural engineering problems; existing frameworks may infer qualitative solutions that do not correspond to the physical solution.

Mechanisms for modelling

Mechanisms for maintaining faithful models include expansion, modification, deletion, rationalisation and focusing [108]. Expansion, modification and deletion are self explanatory. Rationalisation involves the removal of redundant information, which can lead to inconsistencies; focusing is the decomposition of a model into

submodels in order to facilitate the handling of complexity.

Table 5.14: Modelling- details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_e	Knowledge Change
• Knowledge of the appropriate level of analysis (i.e. qualitative, approximate or quantitative) • Knowledge of appropriate assumptions made. • Knowledge of the appropriate analysis method and/or experimental/simulation techniques.	• Knowledge of the reason for deviation between theoretical results of the analysis method and experimental results if any. • Knowledge of the behaviour performance of artefact compared to the design specification. • Knowledge of the appropriate models and modelling techniques to support analysis.	• Faithful modelling of the design dependent on the perspective required for the current design activity.	• A more detailed description of the design depending of the type of analysis.

5.4.3 Evaluation

Evaluation as a design activity is concerned with assessing the suitability of design(s) to satisfy the aims and objectives of the design specification. This is a compound design activity that involves the activity of identifying, measuring and comparing. Reasons for evaluating include checking to make sure that the proposed system is unlikely to fail, comparing proposed systems to find the best for the situation, and/or comparing a candidate system to an ideal to establish its potential quality [141]. In some cases, evaluation is only possible through full-scale testing (e.g. sea trial to evaluate contract speed or sea-keeping capabilities). The result of the evaluation process may initiate other activities such as modifying, standardising and optimising.

In evaluating a design, a design agent can use analysis and its tools, including computer programs such as finite elements (FEM), boundary elements, computational fluid dynamics, circuit analysis, system simulation and so on.

Evaluation procedure

Variations in procedures for evaluation can be observed in the various models of design process, [99, 101, 142]. The key steps in these procedures comprise of the identification and selection of evaluation criteria, the selection of choice of scales for these criteria, the determination of values for the criteria and the processing of the

individual values into a total value.

The procedure requires the use of various evaluation methods (e.g. objective trees with influence values [197], weighted assessment [99, 197]). Because of the need to assign influence values and weightage to the evaluation criteria based on arbitrary scales, the evaluation is to some extent subjective [141].

Evaluation criteria

In any evaluation, a set of objectives and the corresponding evaluation criteria must be identified. The set of objectives is mainly based on the specification (function and performance) and other desirable product attributes (e.g. safety, quality, ease of assembly, maintenance etc.) which differ greatly in importance.

Evaluation needs criteria, statements about acceptable performance related to the properties - tolerable limits, maxima or minima, [142]. Some objective evaluation criteria will contain values or mathematical relationships expressing limits of physical phenomena (e.g. critical buckling strength of plate, the resonance frequency of a structure), but others will be (more or less) subjective, depending on human judgment. These sets of design criteria differ in scope and content for the different phases of design. For the conceptual design phase, the design criteria are usually qualitative in nature. In order that important criteria are not omitted, Pahl and Beitz [99] summarise two separate checklists for evaluation during the conceptual design and embodiment design phases. However, Hubka [141] suggests that evaluation criteria should appear in the design specification, but only a limited selection of these criteria is applicable at any design stage.

The primary aim of the evaluation activity is to determine (measure) either qualitatively or quantitatively the quality or value of a design solution with respect to a given objective. This will result in selecting the optimal solution(s) based on the set of objectives.

Evaluation methods

In the conceptual phase, evaluation of concepts is qualitative in nature using methods such as concept screening (called Pugh concept selection [101]). During concept screening, rough initial concepts are evaluated relative to a common reference

concept using a screening matrix [103]. Concept scoring is used when increased resolution will better differentiate among competing concepts. By weighing the relative importance of the selection criteria, the concept scores are determined by the weighted sum of the ratings.

In embodiment or detail design phase, evaluation of the design goes beyond meeting the functional and performance requirements. Many additional aspects of the design have to be considered for the design and manufacture of the product to be cost effective in order to be market competitive. The evaluation of the design includes other issues such as manufacturing cost. This has given rise to methods of evaluation that today is commonly referred to as Design for X (DFX). DFX can be defined as a knowledge-based approach that attempts to design products that maximise all desirable characteristics - such as high quality, reliability, serviceability, safety, user friendliness, environmental friendliness, short time-to-market - in a product design while at the same time minimizing lifetime costs, including manufacturing costs [159]. For example, Design for Manufacture (DFM) (where X = Manufacturing) is an analytical and knowledge-based technique involving the application to product design of a series of guidelines or rules of thumb, proven from prior experience. Almost all DFM and DFX guidelines are still qualitative in nature and often conflicting. Each of the DFX methodologies involves the application of best practices or guidelines which designers learned as a result of evaluating various aspects of product attributes. Through application of these design guidelines and refining them with each design, the experience of best practices embodied in the sets of rules and procedures increase. This is how the design methodology for the various DFX perspectives have been formulated and documented.

Table 5.15: Evaluating - details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_k	Knowledge Change
• Knowledge of the appropriate level of analysis (i.e. qualitative, approximate or quantitative) • Knowledge of appropriate assumptions made. • Knowledge of the appropriate analysis method and/or experimental/simulation techniques.	• Knowledge of the reason for deviation between theoretical results of the analysis method and experimental results if any. • Knowledge of the behaviour performance of artefact compared to the design specification.	• A measure of the quality or value of the design solution(s) with respect to a given criteria.	• Knowledge of the assessment and rationale of the quality of design solution(s).

5.4.4 Decision making

Eekels [198] defines decision making as the activity of choosing between a number of alternative possibilities for further action. It is a compound activity comprising of evaluation and selection.

There are two basic types of design decisions. The first type is process oriented; that is, the decisions are made to advance the design. The second type is product oriented; that is, decisions are made to effect changes to the state of the design. In analysing an individual designer’s protocol¹⁸, Akin and Lin [199] identified these two types of design decisions. Akin and Lin’s study also indicates that most of the designer’s time is occupied making routine decisions and when novel decisions are made, though few, they are made in the conceptual and development stage of the protocol.

Decision making process

Throughout a decision making process, the designer identifies issues, establishes criteria and provides evaluation statements, until an explicit decision is made to proceed with a particular alternative. Dwarakanath and Wallace [200] observe that many decisions taken were determined by a single dominating factor¹⁹. This dominating factor whether real or imagery has to be identified to understand the

¹⁸ One of the protocols used for analysis by participants at the Delft Protocol Workshop.
¹⁹ In a design experiment (Delft Protocol Workshop), spherical ball was identified as a potential design solution with many advantages but the designer repeatedly returned to the apparent difficulty of manufacturing a spherical ball and this totally dominated the decision to reject this alternative.

influences that affect the decision-making chain.

Design decisions

Good progress towards the design goal occurs when design agent(s) make quality design decisions and act upon them. Quality decisions are those that minimise uncertainty in the design process. Quality decisions cannot be made unless the design agent has appropriate type and sets of criteria for making decisions. To Starkey [197] these quality decisions are absolutely crucial to the success of the design project. He argues that these decisions have effects that are far reaching consequences in that they predetermine many of the constraints that will be placed upon future decisions, as yet unmade. Starkey [197] called these decisions as fundamental decisions from which large numbers of dependent, relatively less important decisions (called intermediates) may spring forth. Following these intermediate decisions, will come a vast quantity of almost unimportant decisions, each springing from decisions already made.

Fundamental decisions can be identified as those that are irretrievable without catastrophic redesign. These crucial decisions are usually made in the front end of the design process when the designer's knowledge of the downstream activities is minimal and minimum data is available. Each change in the objects or their relationships puts new constraints on all downstream decisions.

Intermediate decisions are extensions and supplementary to fundamental decisions that have significant impact on the design should they be retrieved. Their retrieval usually leads to significant redesign in associated areas.

Minor decisions are usually concerned with design details. They cover decisions with regards to component geometry, materials, finishes, processes, tolerances, etc. Their retrieval therefore has less impact on the design.

Types of design decision criteria

Criteria are the yardstick against which alternatives are judged. Design requirements and constraints listed at the early stages of the design process are transformed explicitly or implicitly into criteria. From the protocol analysis of design

experiments, it was observed that criteria evolve and new ones emerge during the design process, [200, 201]. McGinnis and Ullman [201] identify three sources of criteria. Given criteria are imposed on the problem from outside the design space. They may be part of the original problem statement/design specification or imposed on the design problem by the solution of other components of the problem not under the control of the designer responsible for the problem under consideration. Introduced criteria are those criteria brought to the problem by the designer based on specific domain knowledge. Derived criteria arise as the consequence of earlier design decisions. Apart from the *given* criteria, Dwarakanath & Wallace [200] consider the *introduced* criteria and the *derived* criteria as evolved criteria i.e. developed during the design process.

The full set of criteria used for evaluation is clearly not established before generating alternatives, and the very process of generating alternatives triggers new criteria. Of all the criteria generated, designers often only use a subset of the criteria to evaluate alternatives.

Depending on the phase of the design process, criteria can also be classified as objective design decision criteria (mostly quantifiable weighting factors), subjective design criteria (involving qualitative measures) and insufficiently described criteria (based on the designer's intuition, experience, culture, mood, judgement, belief etc.). Inevitably the number, quality and characteristics of the design decisions will increase at each succeeding step. In these situations, making a rational choice may be beyond the capacity of a designer. Also, the designer may arrive at a point where preference precedes a rational choice between alternatives (*a priori* decisions resulting from the designer developing patterns of preferences or values).

Types of decision-making process

Dwanakanath and Wallace [200] identify two main types of decision-making process. In the first type, several alternatives were generated and a set of criteria established. Then, alternatives were compared with criteria and against each other, to arrive at a decision. This type of parallel comparison of alternatives was observed only in the early phase of the design process. Most of the evaluation methods available are directed towards supporting this type of decision-making.

In the second type of decision-making, an alternative was evaluated as and when it was generated. After an alternative was evaluated, it was either modified or a new alternative generated. This was a strongly coupled process of generate and evaluate. Many researchers therefore describe designing as a series of inter-related and sequential decisions.

Design rationale

During the design process, decisions made regarding the design object especially fundamental ones represent the design rationale of the designer. The information related to each design decision would record the alternatives, the evaluation criteria, the pros and cons and the decision made. This set of information represents explicitly the design rationale of the designer. The knowledge of the design decisions becomes crucial for design reuse, design modification/refinement.

Design History

The capture of the history of the design decisions made and acted upon during the design process (from the initial concepts to the final product) can be replayed for the design of similar designs. Due to its impact on the design process, knowledge of both the fundamental and intermediate decisions for the design of a product have led to the development of design support systems that capture the rationale and history of the design process.

Table 5.16: Decision-making - details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_g	Knowledge Change
• Knowledge of design requirements. • Knowledge of design alternatives. • Knowledge of the appropriate analysis method and/or experimental/simulation techniques.	• Knowledge of given criteria, introduced or derived criteria. • Knowledge of fundamental decisions made (i.e. design rational). • Knowledge of selecting the appropriate methods and analysis methods/techniques to support analysis.	• Choose the best alternative(s) from a set based on some criteria.	• From a potential set of design solutions, the knowledge of and the rationale for the choice of best design(s).

5.4.5 Testing/Experimenting

Most designs require some form of testing either during the design process (e.g. model testing of hull resistance) or after manufacture. Products for the consumer or

engineering markets usually require a factory test to verify the quality of the product and its compliance to design specification [101]. Unlike the activity of analysis in which the behaviour of the design is derived through simulation, in testing/experimenting the behaviour of the design is derived through measuring the various parameters describing the behaviour.

Table 5.17: Testing/Experimenting - details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_g	Knowledge Change
• Design specification. • Testing criteria.	• Test results (e.g. resistance against speed curve).	• To verify actual behaviour against expected behaviour.	• Knowledge of design's compliance to specification.

5.4.6 Summary

The categories of design activities classified as design evaluation activities are related as shown in Figure 5.4 below. The arrows indicate the typical sequence of the activities

The purpose of analysis or testing/experimenting is to evaluate the performance of the design. While analysis is achieved using symbolic model(s)²⁰[160] of the physical phenomena the design is subjected to, testing or verifying is performed using physical models of the design or actual design (i.e. full scale). Analysis is a compound activity that consists of mathematical modelling and simulation succeeded by the activity of evaluation. Testing or verifying is a compound activity that consists of scale or full scale modelling and measuring of data followed by the activity of evaluation. Decision-making involves evaluation of alternatives and the act of choosing one or more alternatives (i.e. selecting) based on some design criteria.

²⁰ Symbolic models are systems of concepts and symbols (e.g. diagrams, networks, flow diagrams and mathematical models).

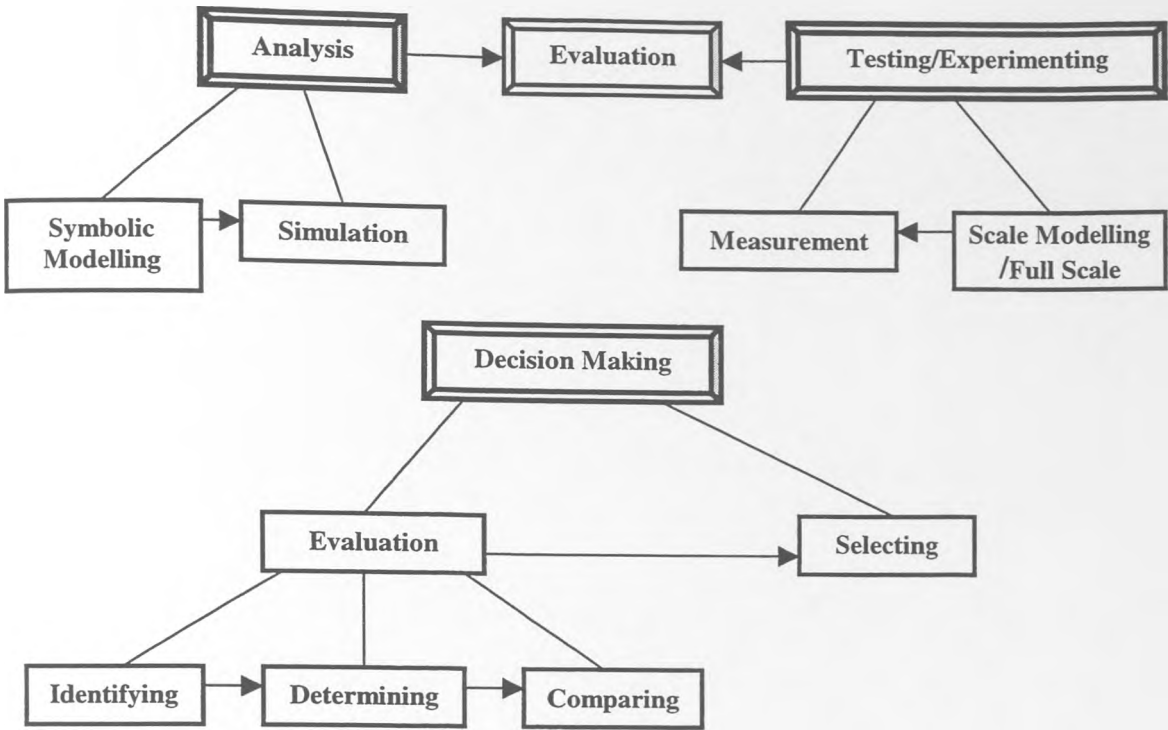


Figure 5.4: Hierarchy of Design Evaluation Activities.

5.5 Knowledge increment in Design management activities

5.5.1 Constraining

The aim of the constraining activity is to limit the exploration of the design solution space. Constraints imposed by the design agent could arise from design requirements [99, 102], engineering codes and standards, design office procedure [202], knowledge and experience of the design agent or design decisions made during the design process [102]. Pahl and Beitz, however, warned against applying constraints during the problem formulation phase of the design process [99] as it may lead to fixation of design solutions. However, as the design is being evolved, it is inevitable that constraints are being applied in order to reduce the design space complexity. Constraints introduced based on well-informed design decisions are dependent on the designer’s knowledge and experience.

Methods of constraining include enforcing hard²¹ constraints at all times and relaxing

²¹ Hard constraints are constraints that must be met at all times; they represent constraints imposed by

soft constraints should there be any violations.

Table 5.18: Constraining - details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_k	Knowledge Change
• Specific design requirements. • Knowledge of types of constraints applicable.	• Knowledge of specific constraints (hard or soft constraints) applied. • Rationale for applying the constraints.	• Reducing the complexity of the design solution space.	• Knowledge of the feasible design solution space.

5.5.2 Information gathering

In any design task, information relevant to the task has to be gathered from a variety of sources, Cross and Cross [203]. Baya and Leifer [204] identify three categories of activities that designers perform with information during design:

1. Generate - an action that adds new information to the information space from an unidentified source. This includes actions such as writing, drawing, talking.
2. Access - an action that references information within or outside the information space. This includes actions such as reading, listening, observing.
3. Analyse - an action that changes the form or representation of information. A fragment of information is considered to have change if its classification changes. This includes actions such interpret, negotiate, organise, calculate, and reason.

Table 5.19: Information gathering - details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_k	Knowledge Change
• A variety of knowledge/information sources relevant to the design task.	• Specific knowledge/information related to the design.	• To progress the design to the next stage or a concrete definition.	• Additional knowledge of the design or the process.

5.5.3 Identifying

Designers reduce the complexity of the design problem by identifying the relevant domain knowledge required, any past design cases relevant to the current design problem and the type of product (i.e repeat design, variant design, innovative or

strict regulations or engineering laws or spatial requirements.

strategic design). They also identify which aspects of the design to focus their attention on during the current state of the design problem (e.g. identifying customer needs, essential design problems, interactions of subsystems, constraints). They also identify what design methods, methodology to use for the activity of analysis, evaluation or what computer software or tools to use.

Table 5.20: Identifying - details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_g	Knowledge Change
• Domain knowledge • Past design cases • Design methods/methodology	• Relevant domain knowledge • Specific design case • Specific design method(s)/methodology	• To progress the design.	• Knowledge of important issues that influence the design leading to a better design strategy or design.

5.5.4 Exploring

Reitman [205] noted that in many open-ended real-world problems like design, the start states, the goal states and the transformation functions are radically under-specified. Because of the ill-defined and ill-structured nature of the design solution space, exploring as a design activity helps designers to define the structure of the design space. This is achieved through decomposing the design solutions into sparsely connected modules [206] and developing each of them incrementally but without any need to make irrevocable commitment to a particular solution [206]. The development of a design solution has several distinct phases. Goel [206] makes a distinction between the problem structuring/problem framing and problem solving. The aim of exploring is to structure the design solution space [207]. Problem structuring is the process of drawing upon knowledge to compensate for missing information and using this knowledge to construct the problem space [206]. From protocol analyses involving architectural design, mechanical design or instructional design, problem structure is associated with how the artefact may be used and what resources are available to form it, whereas problem solving is associated with the specification of the function and form of the artifact [206]. On the one hand, designers need to keep their options open as the design emerges, so as not to crystallise too soon. On the other hand, they need to make, record and propagate commitments so as to bring the design to closure. The client and design briefs are important sources of knowledge during problem structuring. This is consistent with

the purpose that problem structuring is to bring new information into the problem space. This is done through the ‘add and propose’ mechanisms.

Table 5.21: Exploring - details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_g	Knowledge Change
• Knowledge of past designs and solutions • Client and design brief	• Knowledge of problem structure and any missing information and knowledge	• To explore the design space without committing to some solutions.	• Knowledge of potential designs as candidates for solutions. • knowledge of design strategies to choose from.

5.5.5 Searching

Searching is similar to the exploring activity but with an expected end result [183]. Successful search is usually accomplished by search strategies applied to identify search fields. Ulrich and Eppinger [103] classify the search activity into two categories, external search and internal search. External search for solutions is essentially an information-gathering process and the sources include lead user interview, expert consultation, patent searches, literature searches and competitive benchmarking. In searching for design solutions, Hubka [142] suggests two methods, namely, discursive methods (e.g. analogies, aggregation, similarity laws, structuring, inversion method) and intuitive methods (e.g. brainstorming, synectics).

Table 5.22: Searching - details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_g	Knowledge Change
• Knowledge of search strategy • Knowledge/ information sources (e.g. patents, lead user interview etc.)	• End result of the search (e.g. information, patent, component etc.)	• To satisfy some requirement of the design problem or solution.	• Knowledge that contribute to the definition of the design.

5.5.6 Resolving

Conflicting interest, requirements and viewpoints are inherent in design. Conflicts exist in individual agent’s design or in a collaborative design effort [208, 209]. Conflict resolution or resolving conflicts is an important design activity that pervades throughout the design process. Brazier et al. [208] posit that detecting, resolving and managing conflicts requires extensive knowledge of conflict management strategies. Brazier et al. classify four conflict types based on manipulation of requirements, design object description, single agent’s design process and the design process

coordination between the agents and individual agents’ design process. Oh and Sharpe [209] identify technological or task-level sources of conflicts for interdisciplinary design so as to manage and exploit them effectively.

Table 5.23: Resolving - details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_r	Knowledge Change
• Knowledge of possible type of conflicts • Knowledge of conflict resolution strategies	• Knowledge of hard constraints enforced and/or relaxation of soft constraints	• To resolve conflicting interest, requirements and viewpoints.	• Specific knowledge of conflict resolution relating to the design or process.

5.5.7 Selecting

Selecting is a design activity that involves choosing a design object/task that satisfies design requirements from a specified set of alternatives. The term object is used in a general sense, covering, for example, the selection of a working principle for a device, a material type for a component, component from catalog [146], a functional module, or a completed design, design goal [143]. However, selection is usually preceded by evaluation. Selection occurs in all phases of design. Design selection requires the knowledge of attributes of the alternatives, attribute-defining requirements, and the choice criteria to define optimality in selection [187].

Table 5.24: Selecting - details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_r	Knowledge Change
• Knowledge attributes of alternatives • Knowledge of attribute-defining requirements • Knowledge of choice criteria	• Knowledge of object selected • Knowledge of criteria used	• To achieve design definition so as to contribute towards partial solution or complete design solution.	• Knowledge of the optimal choice for a specific design or the process.

5.5.8 Prioritising

Duffy et al. [108] consider prioritising as the ordering of goals on the basis of their importance to and general strategies for the solution process. The inputs to this activity are:

- The knowledge of the relative importance of goals, the implications of not achieving them and the effect they have on limiting the possible solutions.

- The knowledge about the information requirements of each sub-task and resource.

Table 5.25: Prioritising - details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_e	Knowledge Change
• Knowledge of relative importance of goals. • Knowledge of the information requirements of each sub-task and resource.	• Knowledge of agenda of goals in order of priority.	• Focus on important goals that have influence on down-stream design activities.	• Knowledge of specific ordering of goals and subgoals for a design task/project.

5.5.9 Planning

The planning activity organises resources for the design process in terms of the order of tasks, the assignment of personnel and tools for each task [183]. For repeat and variant designs, such plans may be available for reuse with some modification. For innovative and strategic designs, decomposition of the design problem into design tasks and ordering these tasks can be achieved using the triangularisation method by Kusiak and Wang [175].

Table 5.26: Planning - details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_e	Knowledge Change
• Design tasks, resources (e.g. design staff), tools (e.g. CAD systems, design software) • Planning method/algorithms (e.g. PERT)	• Sequence of design tasks and allocation of resources and tools	• Minimise time to market by streamlining design tasks.	• A specific sequence of tasks for a given design project that minimise design iteration.

5.5.10 Scheduling

Scheduling differs from planning in that actual time stamps, time duration (starting and ending time) are specified for each task. Time sensitive tasks that lead to a critical path can be identified using critical path analysis.

Table 5.27: Scheduling - details of the elements of the design activity and knowledge change

Input Knowledge I_{dk}	Knowledge Output O_k	Goal of Design Activity D_e	Knowledge Change
• Design tasks, resources (e.g. design staff), tools (e.g. CAD systems, design software) • Scheduling method/algorithms.	• Sequence of design tasks and allocation of resources and tools in terms of time stamps and due dates.	• Minimise time to market by streamlining design tasks.	• A specific sequence of tasks for a given design project that minimise time to market.

5.5.11 Summary

The design management activities can be further classified into two groups: namely, those that are concerned with the design problem/solution (See Figure 5.5) and those that relate to the design process (See Figure 5.6).

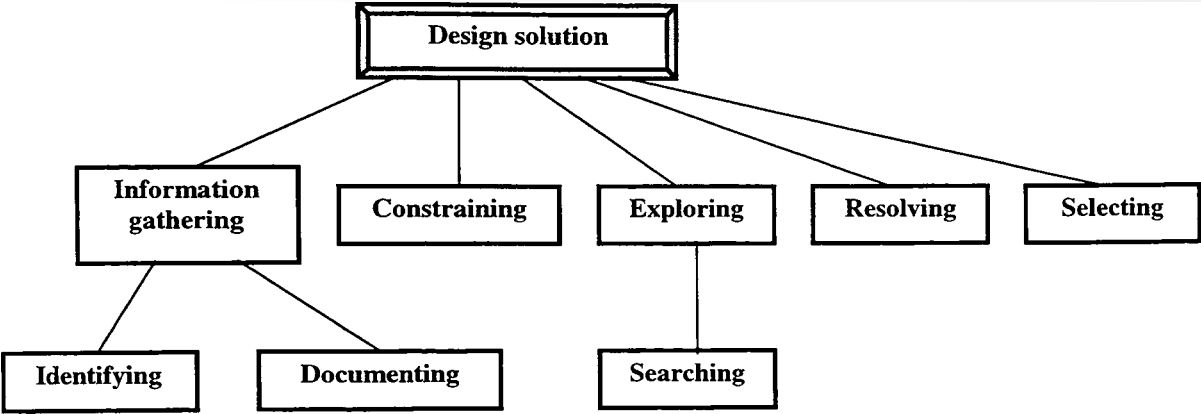


Figure 5.5: Hierarchy of Design Management Activities for Design solution

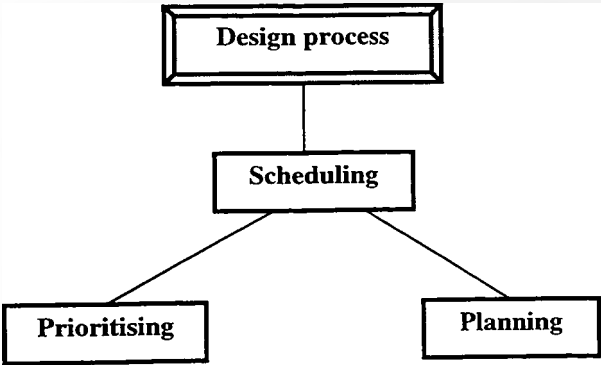


Figure 5.6: Hierarchy of Design Management Activities for Design Process

5.6 Summary

Typical design activities are identified and classified based on the published literature on design and corroborated by design practice as recorded in case studies and protocol recordings. The main difficulty in identifying generic design activities lies in the level of granularity of these activities. By defining the level of abstraction at the knowledge level, generic design activities identified are defined as those that result in knowledge change. These activities were classified into three categories, based on how these categories reduce/resolve the complexity of the design solution and the design process. One of the aims of this Chapter was to present a formalism to represent a generic design activity. The formalism presented has three basic elements, namely the input knowledge, design goal and output knowledge. Using this formalism the knowledge change for the generic activities identified was analysed. The formalism derived here contributes towards the formulation of a model of learning in design (See Chapter 7). An understanding of the knowledge change in generic design activity provides a foundation to study the knowledge change of design activities identified in the protocol recording of a designer at work (See Chapter 8). Hence, the work presented here contributes to the formulation of the model of learning in design and the protocol analysis of knowledge change of design activities, the evaluation of which was used to evolve the model of learning in design (See Chapter 9).

Chapter 6

A formalism for the learning activity

Design experience provides a wealth of knowledge that designers can (re)-use in order to design better products within a shorter time-to-market period and at the same time be economically competitive. The research work in the area of machine learning has contributed many methods that have been applied to the acquisition of knowledge in design. This has been evident from the body of work reported in the field of Machine Learning in Design (MLinD) [32]. What is evident from this work is the application of particular machine learning methods to the acquisition of some specific design knowledge.

Given that there is sufficient evidence from MLinD research in which generalised past design knowledge can be acquired using certain machine learning methods there is a need for a systematic approach to formalising learning in design. Leith [210] argues strongly for a formalism in order to arrive at the state of what “ought to be” in artificial intelligence (AI) and computer science rather than what “is” (i.e. current state of these disciplines) so as to overcome the *ad hoc* basis of software writing and the inefficient development of software systems. Using the same argument of Leith, since machine learning in design is a specialised application of AI and computer science, there is therefore a need to put the study of MLinD research and the development of the MLinD systems to support designers on a formal basis. Hence a formalism has been derived for machine learning in design based on a set of basic elements of learning [32]. The five elements are the input knowledge, the output knowledge, the knowledge transformer, the learning triggers and the learning goal. Since the research work in MLinD has resulted in the development of many published systems in support of knowledge acquisition in design, the formalism has been used to analyse and evaluate these MLinD systems. Hence this Chapter presents the type of design knowledge learnt (both product knowledge and design process knowledge) given the input knowledge and the goal of learning as described in the MLinD systems. It provides evidence of knowledge transformers used to learn design knowledge but implemented in various machine learning methods and

techniques. Evidence of the types of learning triggers and when learning occurs are also described here. A comparison of the formalism of machine learning in design with “Dimensions of Machine Learning” raised by Grecu and Brown [211] concludes that the formalism presents a structured basis upon which to research and develop the field of Machine Learning in Design. Since MLinD systems have been developed based on each of the developer’s perception of what learning in design is, it is reasonable to argue that the formulated formalism [32] can be generalised as a formalism for learning in design.

6.1 Defining a Foundation of Design Learning

In Chapter 5, it has been shown that a systematic approach to the study of learning design knowledge can be based upon analysing the knowledge change of design activities. This perspective is based upon the hypothesis posited by Persidis and Duffy that learning is inextricably linked to design. Thus, learning in design is viewed as a knowledge gaining activity associated with the activity of design.

A formalism for learning in design must be able to address some of the key questions in learning raised in Chapter 4. Since learning can be viewed as an activity, any formalism must invariably consider what knowledge is input into that activity, what is the output knowledge and the knowledge change that transforms the input knowledge into output knowledge. Knowledge change involves the transformation of the existing knowledge into some new knowledge. Since there are many possible ways in which knowledge can be transformed, it is necessary to define what kind of knowledge can be learned. For example given a past design one can learn about the composition of the product in terms of part-of hierarchy, or about the relationships between attributes of components. Further, the learning activity will often have a specific goal, e.g., gain new knowledge of a product, explore and generalise a particular domain. It is also necessary to define what triggers learning and when that trigger can occur.

Given the above, Sim and Duffy [32] have postulated that the basic elements of a learning activity consist of:

- Existing knowledge as Input knowledge, I_k .
- Knowledge transformers, K_t
- Output knowledge, O_k
- Learning goal or reason, L_g
- Learning triggers, L_t (rationale learning trigger, L_{tw} and temporal learning trigger, L_{tt}).

These basic elements of learning may be related as shown in Figure 6.1.

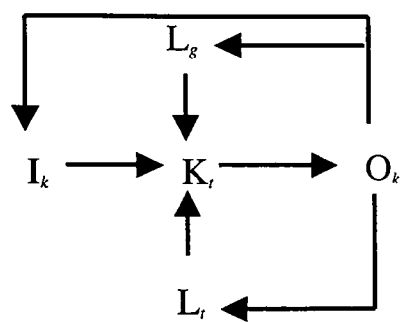


Figure 6.1: Elements of learning

In this figure, it is suggested that the input knowledge is transformed into new output knowledge which can then feed back into the learning activity as input knowledge for yet new knowledge. This output knowledge may in itself also trigger or act as a reason or goal for a learning activity.

6.1.1 Elements of a foundation for machine learning in design

Using the above elements as a basis, we can now map Persidis & Duffy’s, and Grecu & Brown’s, issues and “dimensions” as shown in Table 6.1. It can be noted from Table 6.1 that Grecu & Brown go into far greater detail than Persidis & Duffy, possibly reflecting the evolution of our own understanding of the field. Having said this, the approaches seem complementary and would seem to reflect the basic elements identified above.

Table 6.1: Elements of Machine Learning in Design

Basic Elements of learning	Persidis and Duffy	Grecu and Brown
I_k (input knowledge)	• Not explicitly addressed.	• What are the elements supporting learning? • Availability of knowledge for learning.
K_t (knowledge transformer)	• How is learning carried out?	• Methods of learning. • Local vs. Global Learning.
O_k (output knowledge)	• What knowledge is learnt?	• What might be learned?
T_l (trigger)	• What can trigger learning? • When is learning triggered?	• What can trigger learning?
G_l (goal/reason)	• Not explicitly addressed.	• Consequences of learning.

6.1.2 Types of design knowledge learnt

The types of design knowledge to be learnt are dependent on the activity of the design process, the types of input knowledge, the goal of the learning process and when learning takes place. Persidis and Duffy [35] state five main types of design related knowledge; design requirements, design description, domain knowledge, case histories and design management. Failures in design provide an opportunity to learn about the causes of the failure in order to avoid committing the same failure again. Knowledge learnt from design failure could be in the form of types of failures and conflicts [211, 212], heuristics for failure recovery and conflict resolution [211, 212]. On the other hand, successful designs and design processes [211, 212] also provide opportunities for learning about the characteristics of the successful past designs and goals/plans of design processes. The review of MLinD systems in Section 6.2 shows that indeed there are numerous types of design knowledge that can be learned from past designs. It is therefore reasonable to conclude that there are many types of design knowledge that can be learnt.

6.1.3 The learning goal

Because of the variety of knowledge that can be learnt, it is sometimes necessary to specify a learning goal. The learning goal influences what parts of the existing knowledge are relevant, what knowledge is to be acquired, in what form, and how the learned knowledge is to be evaluated. Such goals, as in design, can be hierarchical in nature in that a specific goal could be a sub-set of a more general goal. For example, learning about a particular technology may help improve the quality of the product or speed up the design process. Another example of specific

learning goal such as learning about constraint violation in design may result in better conflict anticipation and resolution which in turn may lead to better design plan and hence shorter design cycle [3]. Thus, specific learning goals may be the stimulus for many of the unforeseen consequences of learning.

6.1.4 Types of knowledge change

The design process is the vehicle that designers use to change the state of the design through the application of scientific and design knowledge. As the state of the design changes, there is the corresponding change of knowledge state of the design. Persidis and Duffy presented three main classifications that knowledge change can be considered under, namely acquisition, modification and generation:

- *Acquisition*: the process in which new knowledge, either through direct input from the user or derived from knowledge sources, is received.
- *Modification*: the process of altering existing structure of knowledge through either addition or deletion of the knowledge component from the structure. The modified knowledge structure remains the same type.
- *Generation*: the process of creating completely new knowledge from existing knowledge.

Whilst knowledge change through modification is explicitly described as addition/deletion of knowledge component from the knowledge structure, the several ways in which knowledge change through generation is achieved are described below. The different processes by which knowledge change takes place are called knowledge transformers. In the study of machine learning in design systems, seven knowledge transformers for knowledge generation and one knowledge transformer for knowledge modification are identified. Some of these transformers are similar to those described by Michalski's set of eleven knowledge transmutation [63]. Of these eleven, the knowledge transmutation of replication/destruction, sorting/unsorting²² and selection/generation are not classified as knowledge transformer as they are basically processes for re-organising knowledge bases. There is no learning in the

²² From the protocol analysis of Chapter 8, this was subsequently added as an additional pair of knowledge transformers to cater for the knowledge change arising from the design activity of prioritising.

form of knowledge change *per se*.

6.1.5 Knowledge Transformers

Because of the nature of design knowledge, which may exist in symbolic form or numerical form, knowledge change operators or knowledge transformers consists of two types: the symbolic type and/or non/sub symbolic type. A knowledge transformer is an operator that derives a piece of new knowledge from a given input or an existing piece of knowledge.

Group Rationalisation (or Clustering)/Decomposition (Ungroup)

Group rationalisation involves the grouping of past designs according to their similarities when considering particular perspective(s) or criteria [109]. The grouping may be based on single attributes and/or nested attributes. Decomposition removes the groupings.

Similarity/Dissimilarity comparison

Similarity comparison derives new knowledge about a design on the basis of similarity between the design and similar past design(s). The similarity comparison is based on analogical inference. Case-based design is a specific application of learning by similarity comparison in which the source of past design knowledge is a set of similar design cases rather than generalised design rules. The opposite is dissimilarity comparison that derives new knowledge on the basis of the lack of similarity between two or more past designs.

Association/Disassociation

Association determines a dependency between given entities or descriptions based on some logical, causal or statistical relationships. For example, two entities may be related together by a taxonomic relationship (i.e. kind-of) or compositional relationship (i.e. part-of). The opposite is disassociation which asserts a lack of dependency (e.g. not part-of).

Derivations (Reformulation)/Randomisation

Derivations are transformations that derive one piece of knowledge from another piece of knowledge (based on some dependency between them). In contrast, randomisation transforms one knowledge segment into another by making random changes. Because the dependency between knowledge components can range from

logical equivalence to a random relationship, derivations can be classified on the strength of dependency.

Generalisation/Specialisation

Generalisation generates a description that characterises all the designs based on a conjunction of all the specialisations of that concept. Typically the underlying inference is inductive. But generalisation can also be deductive, when the more general description is deductively derived from the more specific one using domain knowledge or existing knowledge (e.g. as in explanation-based generalisation). Specialisation increases the specificity of the description.

Abstraction/Detailing

Abstraction generates a new version of the knowledge with less detail than the original through the use of representation of abstract concepts or operators. As distinct from generalisation, the underlying inference is deductive in nature. Detailing is the opposite in which the new knowledge is generated with more details.

Explanation/Discovery

Explanation derives additional knowledge based on domain knowledge. Discovery derives new knowledge without an underlying domain knowledge.

The knowledge transformers described above are not synonymous with machine learning methods that are used machine learning systems. The knowledge transformers describe the cognitive learning process whereas the machine learning techniques describe how that process can be implemented as a algorithm. For example, explanation-based generalisation uses generalisation and explanation of domain knowledge to derive new knowledge.

6.1.6 The learning triggers

Persidis and Duffy [35] posit the need to identify events that act as “triggers” for the self activation of the learning process. Sim and Duffy [32] presented that these triggers can be classified into three main categories: in-situ, provisional and retrospective.

- *In-situ triggers* are activated when there is a need to acquire new knowledge while the design is under focus of attention.
- *Provisional triggers* are activated when there a foreseen event that is

envisaged to require additional knowledge. An example of a provisional trigger arises when there is a need to learn heuristics that may lead to a reduced search for a design solution. This learning is to be done in anticipation of or provisionally for a foreseen event.

- *Retrospective triggers* are activated after an event. That is, learning is triggered by the need to learn from successful design(s)/failed design(s) and/or processes in hindsight.

Grecu and Brown [211] identify several situations that could trigger learning. The reasons for learning are numerous but may occur due to the following:

- *Novelty driven*: when there is a new design problem, a new technology, a new process or a new design requirement.
- *Excellence driven*: can the design be improved to achieve better quality and reliability? Can the design process be more streamlined or time-efficient or resource efficient? Can the design be market competitive?
- *Failure avoidance driven*: can failure(s) in the design be avoided? Can design constraints be overcome?

The above examples serve to illustrate reasons that give rise to learning and are by no means exhaustive in coverage. For specific examples (gleaned from the study of MLinD systems) of what triggers learning and when learning is triggered, more details are given in Section 6.5 and in Table 6.4.

6.1.7 Current work of Machine Learning in Design

There is now considerable effort in supporting computer-based learning in design. Such work can act as a basis upon which to discuss the above elements. The review of such systems has been broadly classified into two groups:

- Systems that acquire knowledge of a *product* (i.e. design concepts).
- Systems that acquire knowledge of the *design process* (i.e. procedural knowledge and control knowledge).

It is difficult to ascertain the true nature of the goal or reason for an associated MLinD system's learning activity. Consequently, the reviews had concentrated upon attempting to identify:

- The types of design knowledge acquired (Element O_k) and the associated learning goal (Element L_g).
- The knowledge change as a result of the learning activity (Elements O_k/I_k).
- The knowledge transformers (Element K_t) and the corresponding machine learning method(s) used in implementation.
- What and when trigger that learning activity (Elements L_w for rational learning trigger and L_t for temporal learning trigger)

6.2 Types of Design Knowledge Learnt

Many MLinD systems have been developed to automate the learning of design knowledge. The purpose of this section is to show the variety of design knowledge that have been acquired through these MLinD systems. For the convenience of description, the types of design knowledge learnt can be categorised into the following:

- Product design knowledge,
- Design process knowledge.

This categorisation is important as it was revealed through the review of the MLinD systems that in most cases learning of meaningful product design knowledge occurs retrospectively. Similarly certain types of design knowledge is best learnt while the design is in progress.

6.2.1 Product Design Knowledge

Systems such BRIDGER [213], NODES [108, 214], CONCEPTOR [215], PERSPECT [109] and NETSYN [184] have addressed the issue of learning product knowledge from past design concepts. The knowledge learned can be discussed under the heading of:

- Composition of the components/subsystems that constitute the product.
- Constraints imposed on the attributes due to the physics of the problem, geometrical and spatial requirements, life-cycle issues, design requirements, etc.
- Decomposition of the components that constitute the product.

- Performance evaluation knowledge.
- Dynamic learning of implicit knowledge.

Learning Knowledge of design composition

Knowledge of the composition of past designs serves as a useful starting point to initiate the synthesis process of design problems that have a similar design specification. For example, BRIDGER was primarily developed for the synthesis of different concepts of cable-stayed bridges. NODES was also developed primarily to support concept modelling operations during early stages of design [108].

Whilst BRIDGER expresses the design composition knowledge as a taxonomic hierarchy (i.e., kind-of relationship), NODES expresses both the taxonomic relationships as well as the compositional (part-of) relationships of a design object. BRIDGER uses the concept formation capability of ECOBWEB to learn design composition knowledge, NODES uses maximum conjunctive generalisation (MCG) to build a generalisation of a new concept into existing concept(s).

The knowledge input into NODES or BRIDGES consists of past designs instances or abstraction in terms of taxonomic hierarchy (i.e. kind-of relationship and/or part-of relationship). Through the process of knowledge change (i.e., clustering or generalisation) the output knowledge will be a new taxonomic structure updated to reflect the changes caused by the inclusion of the new design instance.

Learning Empirical Knowledge of Quantitative and Qualitative Relationships

The numerical relationships of attributes of design concepts are important in defining the preliminary definition of the form or structure of the concept. The estimation of these values has been achieved by scaling or interpolation (BRIDGER [213]), empirical network of association (CONCEPTOR [215]), statistical approximation (NODES using Designer [108]), and neural network (NETSYN [184]). Each of these types of estimation of attribute values based on records of past design cases is described below.

Although there is in BRIDGER a taxonomic structure generated from past design

concepts in terms of attribute value pairs, the numerical relationships of these attributes are not explicitly represented. Therefore for a given design specification, BRIDGER derives appropriate values for the attributes through interpolation of suitable design cases.

In the case of CONCEPTOR, the empirical relationships of design attributes of a given design domain is represented as an empirical network. The empirical network captures both quantitative variables and qualitative variables as nodes, which can be linked by empirical formulae (for quantitative variables) or design patterns (for qualitative variables). The empirical formula represents proportional relationships among the quantitative variables. Design patterns capture important association among qualitative variables on which the designer should focus their attention on. Each node represents a design feature, its default value and range value. Collectively, the empirical network of nodes linked either by empirical formula or design patterns represents the description attributes of a learnt concept.

In NODES, the numerical relationship of attributes of the design object is described by characteristics and formulae. Characteristics represent numerical attributes of concepts and are always associated with a single object. They are also associated with formulae and together form a constraint network where the values of characteristics are constrained by the values of other characteristics appearing in the same formula. Formulae represent mathematical equations that allow the values of characteristics to be calculated from other characteristics. Formulae, like characteristics are also associated with objects but unlike characteristics can be associated with more than one object.

Learning knowledge of design constraints and design expectation

Both design constraints and design expectation play a crucial role in controlling the design process. A design expectation is a designer's prediction about a design attribute's value. Expectations may consist of a default value, a range of possible values or a simple relation between design attribute values. Relationships between key design attributes and dependent design attributes are expressed as expectation rules by Chabot and Brown [216]. An expectation rule may be triggered when a

value for a key attribute has been decided. These expectations and expectation rule(s) are important source of design knowledge that can be learned from past design experiences and past design cases by induction [216].

Design constraint knowledge provides the means for detecting design failures [216]. Although there are a large number of constraint types, categorising them into hard or soft constraints aid in detecting design failures. Hard constraints are either satisfied or violated. If a hard constraint is violated, then the design decision becomes unacceptable. Values for soft constraints are bounded by degree of error tolerance that is acceptable or allowable. So while expectation rules provide values to progress a design, constraint(s) test that value and detect any design failure.

Since different constraints and different expectation rules are called into play during a design process, Chabot and Brown use the mechanism of constraint inheritance as the machine learning method of knowledge compilation [217] whereby new knowledge of constraints are incrementally formed or updated.

Learning design decomposition knowledge

Liu and Brown [218, 219] posit that decomposition knowledge can be generated from design object knowledge, functional knowledge, design cases, design heuristics, general problem-solving knowledge and domain knowledge. In order to extract knowledge appropriate for a given design context, Liu and Brown introduce the concept of decomposition factors. A decomposition factor is a suggestion about how to partition an entity that may be a design object, a component or a set of attributes. The learning mechanism used is that of knowledge compilation [217]. Knowledge compilation is a type of learning in which existing knowledge is converted into new forms with the intent to improve problem-solving efficiency.

The final decomposition knowledge is represented in a tree structure. Each node of the decomposition tree contains a problem description and a list of possible alternative competing decomposition hypotheses, as well as a set of links to sub-problems, knowledge sources, interaction information among sub-problems and other relevant information.

From design object knowledge, Liu and Brown identified three types of interaction among the components of an object. The first type of interaction deals with explicit relationships among the components whilst the second type of interaction arises from the relationships between the object's attributes and the components' attributes. The third type deals with interactions arising from the inter-component relations giving rise to interactions among the sub-components.

Representing the components and interactions as nodes and links respectively in a graph, several pivotal nodes with many links converging to them can be identified. Each pivotal node plus its connected nodes forms a decomposition factor.

Design cases in which specific design problems with their associated design solutions can be another source for generating decomposition hypotheses. Given a design problem, a set of relevant design cases can be retrieved by using the object's case index. A suitable case is selected if every constraint of the current design problem matches the constraints of a design case to predetermined degree; then the design case becomes a candidate for decomposition factor seed. All the selected cases are converted into decomposition factors. If a selected case has a matching quality above some limit, it can be used for generating decomposition hypotheses.

Learning knowledge for performance evaluation

Performance evaluation of a design in the various stages of the design process often determines if the design should be progressed further. But performance evaluation is not an exact science. Multiple criteria that are conflicting in nature are used to evaluate the performance of a design or designs. Current methods of evaluation are often either too simple or too complex for configuration design decisions [220]. Several researchers [220, 221] have posited that known configuration solutions in terms of its topological or geometric layouts, components, and materials can be analysed and evaluated in terms of their performance so that good aspects of these designs may be reused and poor ones changed or discarded. That is, knowledge of performance evaluation can be learned from past design configurations. Several machine learning approaches [220, 221] have been developed to elicit the acquisition of design rules that map from the design solution space (i.e. structure or form) to the

design behaviour space.

McLaughlin and Gero [221] formalise the relationship between decisions about values of design variables and their consequent performances as a mapping between a decision space and a performance space. Design variables can be represented as axes in a design (or decision) space, where each point in the space will represent a particular combination of design decisions regarding the design variables. By identifying the *best* feasible point in the performance space and mapping back to the decision space, the variables and their values that resulted in this performance can be identified. For performance space associated with multiple criteria, Pareto optimisation is used to isolate the set of best solutions in terms of the criteria defining the performance space.

The knowledge acquired either through the Pareto/ID3 [221] or GA/Kohonen Feature [220] Map is:

- Knowledge of the mapping between performance space (described by evaluation criteria) and the design decision space (described by design variables and their values).

Learning function knowledge and the causal models

Bhatta and Goel [222] posit that generic teleological mechanisms (GTMs) are one type of design abstractions that can be learned from past designs through cross-domain analogies. Cross-domain analogy involves the recognition of similarity between two problems from two different domains and determining what knowledge to transfer and how to transfer between them. GTMs take as input the functions of a desired design and a known design and suggest patterned modifications to the structure of the known design that would result in the desired design.

Dynamic learning of implicit design knowledge

So far, the systems that use machine learning methods to acquire the design object knowledge for a given domain have been based on a fixed viewpoint that is pre-determined by the knowledge engineer. Since it is not possible to completely predict designers' knowledge requirements as each designer has different knowledge needs at different times and for different reasons, Kerr [109] presents a new approach to

utilise experiential knowledge called *Customised Viewpoint*.

The key concept behind this flexible approach to knowledge utilisation is that it generalises experiential knowledge directly from specific experiences, according to designers' knowledge needs and subsequently utilises this knowledge in design. To illustrate the concept, Kerr developed, tested and evaluated the utility of the customised viewpoint approach within the realm of numerical design. PERSPECT is supported by four sub-systems, namely DESIGNER, ECOBWEB, S-PLUS and GRAPHER [108, 109].

PERSPECT uses ECOBWEB, a concept formation system to generate and organise the generalisation of a set of past designs (described by attributes and values) into a conceptual hierarchy. It classifies each example (past design) and incorporates it permanently into a hierarchy by incrementally changing the hierarchy's structure. The knowledge acquired by the approach is:

- Multiple forms of experiential knowledge of a domain (described by name, meaning and units of attributes) in terms of empirical equations, numerical generalisation and heuristics. For example, empirical equations quantify the relationship between these attributes together with a measure of the unreliability of these equations.
- Multiple forms of implicit experiential knowledge through generalisations of past design information and identifying the most suitable generalisation of past design that support the *Customised Viewpoints* for a new design.

6.2.2 Design Process Knowledge

Learning about the design process provides just as important a wealth of knowledge as learning from past designs. A design process usually consists of design decisions made which results in design actions taken in order to progress the design. Design actions interact with the evolving design. Learning about the design process invariably involves learning about the decisions made, the rationale for the decisions made and the effect of that decision on the evolving design. Certain phases of the design process are exploratory in nature and the use of control knowledge to manage the exploration is important.

Systems that capture design rationale [31] and design history thus provide vital sources of knowledge for learning about the design process. While capturing design rationale usually involves recording successful design decisions, design history records both successful and failed decision lines. Both successful decisions and failed decision lines are objects of learning about the process.

Knowledge of the design process can be captured in the form of design plans or as a hierarchy of activations (condition and action pairs) on the blackboard of a blackboard system [106, 223]. In artificial intelligence (AI) parlance, a sequence of dependent actions is called a plan. As such, systems such as ADAM [224], BOGART [225], ARGO [226] and DONTE [227] have used the AI approach of planning to describe and capture the design process, whereas only DDIS [228] has used the blackboard concept. Design plans usually do not capture failed lines of design actions whereas the blackboard approach usually captures both successful as well as failed design actions, resulting in capturing the design history.

If the design plan is expanded to lower levels of abstraction it also defines the hierarchical decomposition of the abstract actions into the primitive actions, the outcome of which change the states of the evolving design. The evolving design is expressed initially in terms of functional requirements and in its lowest level of abstraction as design objects that collectively synthesise into design configuration or structure. The design plan therefore captures not only design history, design decisions (i.e. design intent and alternatives and rationale), design strategies but also design object knowledge.

The knowledge change of design plans can be accomplished by acquisition of new design plans, modification of existing plan through design re-use and generation of new plans or generalised plans from existing plans at the end of one or more similar design processes. The review of systems such BOGART, ARGO, DONTE, CDA [238] and DDIS provides evidences of such knowledge change.

Acquiring Design Plans

A knowledge change in a design plan can be learned through the process of acquisition. This simply involves being told by the designer or recording the series

of actions taken by the design, by inferring the design plan from previous design case or solution [112].

Both BOGART and ARGO record user inputs as a means of acquiring new plans. BOGART uses VEXED to record successive design steps in a tree-like design plan that consists of nodes representing design modules. Each module can be decomposed or refined into sub-modules by a catalogue of “if-then” refinement rules provided by VEXED. If VEXED lacks any of these rules, it will learn the manual decomposition step by generalising that step into a new rule. This learning facility within VEXED is provided by LEAP (for “Learning Apprentice”) [235].

ARGO acquires a plan as it solves a problem and is represented using an acyclic graph of dependencies among plan steps (instantiated rules). If one plan step adds an assertion that satisfies the condition part of a rule, the second step becomes dependent on the first. The dependency graph may contain independently solvable sub-problems or dependent sub-problems with justifications maintained by the truth maintenance system.

In ARGO, the design plans are represented as schemas of corresponding preconditions and post-conditions which is represented as a database of assertions stored as slots of frames in a truth maintenance system. A module is represented as a collection of assertions describing its specification, components, interconnections and so forth, each with a belief status of IN or OUT supported by a set of justifications. The OUT status is caused by actions that fail and the corresponding rule instances are not included in the dependency graph for representing the design plan. Therefore ARGO does not learned plans that incorporate failed lines of reasoning.

Acquiring case dependent design plan

DDIS integrates both domain-based reasoning and case-based reasoning in its strategy in solving design problems. It uses case-dependent knowledge that it acquires from current design session(s) to supplement its domain independent knowledge for future design(s) or redesign. DDIS therefore records all design actions as knowledge source activation records (KSARs) and the design history (a

sequence of executed knowledge sources (KSs) and their bindings) on its design blackboard. By analysing these records, DDIS abstracts case-dependent plans and goals. These plans and goals can be directly posted on the control blackboard by case-dependent control actions during subsequent design sessions. These case-dependent actions compete with case-independent knowledge sources at every design cycle to allow case-based reasoning to influence domain-based reasoning so that past design actions leading to dead-ends or failures are avoided.

DDIS therefore generates case-dependent design knowledge after the completion of the design session. The knowledge generated consists of the following:

- Control knowledge of a particular session is also abstracted to a global design plan and several redesign plans so that they can be used separately in a flexible manner according to the new situations encountered.
- Knowledge of constraint violations that can be applied to new cases in order to focus early on critical constraints most likely to cause problems.

Generalised design plans

BOGART, ARGO and DDIS also generate generalised plans from several plans or by abstracting new plans. The details of how the generalised plans are generated through various knowledge transformers are given in Section 6.4.

6.2.3 Summary of the types of knowledge learnt

Table 6.2 gives a summary of the various types of design knowledge that can be learnt from the MLinD systems. The Table shows the knowledge input I_k , the product/process knowledge learned O_k and the reason for learning that knowledge, L_g .²³

²³ This is usually inferred from the context of the design problem described.

Table 6.2: Relationships between input knowledge, output knowledge and learning goal.

Input Knowledge I_k	Output knowledge O_k	Learning Goal L_g
Instance(s) of past design(s) together with existing taxonomic or compositional knowledge.	• Taxonomic knowledge of design concepts (e.g. BRIDGES). • Compositional knowledge of design concepts (e.g. NODES)	• Expedite synthesis of preliminary design concepts.
Records of past designs in terms of attributes and attribute values	• Empirical knowledge of quantitative information (e.g. CONCEPTOR, NODES). • Design patterns of qualitative relationships(e.g. CONCEPTOR, NODES).	• Expedite preliminary definition of form and structure of design concept.
Knowledge of current design constraints. Records of failed constraints	• Knowledge of new/updated design constraints (e.g. Chabot and Brown). • Knowledge of anticipated crucial constraints (e.g. DDIS).	• Streamline design process by detecting and avoiding design failure. • By checking crucial constraints early in the design leading to shorter design cycle.
Past design cases of design problems and corresponding solutions.	• Design decomposition knowledge in terms of decomposition factors (e.g. Liu & Brown).	• Streamline design process by focussing on interrelated systems and/or components.
Past design configurations and performance evaluation criteria.	• Knowledge of mapping between performance evaluation space and design decision space (e.g. McLaughlin & Gero; Murdoch & Ball).	• Excellence driven to achieve better design.
Past design concepts described by attributes and values.	• Multiple forms of explicit design knowledge (e.g. PERSPECT). • Multiple forms of implicit design knowledge (e.g. PERSPECT).	• Excellence driven by utilising knowledge from multiple sources.
Records of design actions described by preconditions and postconditions.	• Abstracted Design plan (e.g. ARGO) • Case dependent design plan (e.g. DDIS).	• Streamline design process by replaying a similar plan.

6.3 Knowledge Transformers in MLinD Systems

In this section, the purpose is to show the evidence of the knowledge transformers considered in Section 6.2 that MLinD systems used to transform input knowledge into new or modified design knowledge. Although these knowledge transformations are implemented in terms of various machine learning methods (either symbolic or sub-symbolic), it is in the nature of the characteristics of each type of knowledge transformation that made their identification within the MLinD systems possible. Each machine learning method used may be either symbolic or sub-symbolic in nature depending on the nature of the representation of the input/output knowledge. In describing the various knowledge transformers, no distinction is made as to

whether the knowledge transformed is product design knowledge or design process knowledge.

6.3.1 Group rationalisation (or Clustering)/Decomposition (Ungroup)

BRIDGER and CONCEPTOR are examples of machine learning systems that learn design concepts using the knowledge transformer of group rationalisation or clustering. While CONCEPTOR uses the concept clustering system called COBWEB, BRIDGER uses ECOBWEB [213],²⁴ which is an extension of COBWEB. Both machine learning systems for concept formation generate design concepts from the characteristics of similar past designs. Using past bridge designs as training examples, each system generates hierarchical classification structures which can be used to assist in the synthesis of similar bridge designs.

CONCEPTOR not only learns design concepts from past designs through the knowledge transformers of clustering (or concept aggregation - a term used in CONCEPTOR) it also derives numerical relationships (or concept characterisation in CONCEPTOR's terminology) among the quantitative design attributes of a concept. Derivation as a knowledge transformer is described later in Section 6.3.4.

COBWEB/ECOBWEB

To build the hierarchical structure, COBWEB/ECOBWEB integrates the processes of classifying examples and incorporating them into a hierarchy. It employs five operators to determine how best to incorporate an example into the hierarchy. Each resulting partition (classification) is evaluated using a utility function to determine the category utility [229] which is a measure quantifying the similarity between members of a partition. It selects the classification that results in the highest category utility value, incorporates the example permanently into the hierarchy, and generates the appropriate conceptual description that suits the new incorporated example. Each partition is described as a conjunction of attribute-value pairs and each partition has a probability to indicate its frequency of occurrence in the training examples. As a result, a hierarchy represents only one concept, while the nodes in the hierarchy represent subsets of the concept.

²⁴ ECOBWEB is a modified version of the concept clustering system called COBWEB.

Hence, in learning taxonomic-type design concept(s) from past designs, the knowledge change involved is the clustering of past designs into a hierarchical structure of concepts, each node in the hierarchy representing subsets of a main concept. Knowledge of the design artefact such as functionality, structure or behaviour is not explicitly represented. The knowledge of the artefact's decomposition structure or composite structure in terms of part-of links is not reflected in BRIDGER's/CONCEPTOR's classification hierarchy structure.

Another example where the knowledge transformer of clustering is used is in the context of learning design performance evaluation knowledge. The design solution space has to be categorised into different classes of solutions for the purpose of mapping them with the performance evaluation space. To elicit knowledge for performance evaluation, meaningful mappings between the decision space and the performance space modeled must be established. While McLaughlin and Gero [221] concentrate on the solutions near the Pareto boundary as sources of knowledge in the evaluation space, Murdoch and Ball [220] suggest that the entire solution set both from the design space and evaluation space represent valuable design information that must be analysed for effective reuse. The entire evaluation space and the design spaces must be analysed to identify different classes of solution and trends in design practice. Because of different areas of the design solution space and performance space considered for mapping, McLaughlin & Gero [221] chose ID3 to distinguish solutions that are Pareto optimal (i.e. near the Pareto boundary) and those that are not, while Murdoch & Ball [220] chose the clustering capability of the self-organising neural network called Kohonen Feature Map to categorise the entire design solution space.

ID3

The induction algorithm ID3 is used as a means of inferring general statements about the nature of solutions that exhibit Pareto optimal performance in terms of a set of performance criteria. The positive example set consists only of decision and performance data of solutions that are Pareto optimal in terms of chosen criteria. The negative example set is generated combinations of design decisions that are inferior in performance. The heuristic rules that best represent the concept to be

learnt are those with the most positive examples and least negative examples.

Kohonen Feature Map

Kohonen Feature Map is a neural network that can learn clustering patterns unsupervised. That is the mapping between criteria in the performance evaluation space and the configuration parameters (i.e. component and materials) in the design space can be clustered without pre-categorisation of the design space.

Each solution consists of a set of parameterised components that can be applied as a training example to the network. The network uses an unsupervised learning algorithm to generate a mapping between the high-dimensional design space of component parameters and the neurons in the network. The map generated provides a topological (i.e. nearest neighbour) relationship among the component parameters. The network topology can then be inspected to identify clusters or archetypes that span the original set of design solutions. By decoding the nodes within each cluster back to the performance evaluation space, a mapping between the two spaces is achieved. The archetype solutions are then analysed to identify the characteristics of configurations that contribute to high or low technical merit²⁵.

6.3.2 Similarity/Dissimilarity comparison

Knowledge change of a design plan through similarity comparison is made possible if there exists an original target design plan. The machine learning method used is learning by analogy that involves a transfer of information/knowledge from a base domain/plan to a target domain/modified plan.

Having acquired the history of design decisions made in a previous design BOGART [225, 230] uses the derivational analogy method by Carbonell [231, 232] to change the design plan by reasoning from the previous plan. The derivational analogy method represents a problem-solving plan as a hierarchical goal structure, showing how and why each goal was decomposed into sub-goals. It solves a new problem by replaying this plan top-down. When the sub-plan for a sub-goal fails, the plan is modified by solving that sub-goal from the user input of a new solution. By this

²⁵ Technical merit combines in one generic measure of design merit three fundamental criteria: performance (duty index), reliability (reliability index) and economy (cost).

process of similarity/dissimilarity comparison, a new design plan is constructed.

6.3.3 Association/Disassociation

NODES learns/builds a model of the conceptual design linked by a compositional network of concepts through part-of and kind-of association and numerical network of characteristics through association between object and formula. In the compositional network the nodes denote objects or assemblies and the arcs denote the directed relation (or association) part-of between two nodes. In the numerical network, nodes represent characteristics of objects or formulae and arcs represent the link or association between two nodes, one of which is a characteristic and the other of which is a formula in which the characteristic appears [215].

6.3.4 Derivation/Randomisation

The concept characterisation phase of CONCEPTOR is an example of deriving new knowledge based on some dependency between them. After concept aggregation, CONCEPTOR derives two types of relationships within a concept: empirical formulae among quantitative design attributes and design patterns among qualitative attribute-value pairs that frequently appear in past examples.

NETSYN

For some design problems, there is a need to apply the probability estimation function to acquire the probability estimate of each value of each unassigned design property (e.g. in estimates of loading, the material behavioural properties are statistical in nature).

Ivezic and Garrett developed a system called NETSYN [184] to learn the Bayesian a posteriori probabilities of design properties. NETSYN uses the feed-forward back-propagation neural network as the machine learning technique to acquire and represent the probability estimation function. The probability estimation function is acquired through inductive learning using past designs to train the neural network to estimate the desired probabilities. The trained network estimates Bayesian a posteriori probabilities.

To use conventional classifiers, one has to estimate the conditional probabilities $P(DIH_i)$ for each design property and the a priori class probabilities. The main

difficulty lies in the elicitation of conditional probabilities $P(D|H_i)$ that reflect the actual design knowledge. This is estimated by assuming some idealised probability distribution (e.g. Gaussian distribution). The neural network approach estimates the Bayesian probabilities in a direct way offering an approach where prior assumptions on probability distributions need not be made.

The computational model estimates the probability for each value of each property being used in a given design context. Each design context involves several design properties for which values have to be assigned. Therefore the construction of NETSYN architecture is modular, i.e., for each design property a neural network structure is assigned to act as a probability estimation function for that property.

6.3.5 Generalisation/Specialisation

Knowledge derived through generalisation has greater problem solving scope. This is because generalised rules or knowledge can be generally applied to a wider range of problems for a given domain or complex problem. Different types of knowledge related to the design product/process can be derived through the process of generalisation. For example, NODES enriches its design knowledge base called the concept library by progressively accumulating solutions of problems defined within a particular domain. It uses generalisation as a knowledge transformer so that new concepts are reflected in all concepts which are generalisations of that concept. By integrating with the Designer system, numerical aspects of the concept (i.e. characteristics and associated formulae) can be analysed. Both BOGART [225] and ARGO [226] acquire generalised design plans from several plan instances or by abstracting new plans. Design action interacts with the evolving design. The interaction between design action and design could be generalised into useful design rules.

Generalised design concepts

NODES generalises knowledge from the most comprehensive concepts within a concept library to the less specific. Numerical parameter ranges and compositional knowledge are generalised to all associated super-classes to ensure that there is no contradiction between a particular concept and its specialisations. The generalisation mechanism that is responsible for the updating of knowledge is invoked automatically whenever a new concept is saved in the concept library.

When a design has been completed, the evolved model in NODES is used to increase its knowledge by acquiring the relevant knowledge of the new design. The mechanism involved is the decomposition of the design into its constituent concepts (or specialisations), along with appropriate constituent and connective relations and merges each concept with its corresponding library.

NODES uses a machine learning technique called maximal conjunctive generalisation (MCG) [233]. MCG ensures that no item of knowledge is associated with a concept unless it is associated with all the concepts that are a specialisation of that concept. In terms of set theory, this means that the set of items of knowledge associated with a concept is the intersection of the sets associated with the specialisation of that concept.

Generalised design plans

BOGART uses VEXED's ability [0] to interactively record decisions in terms of general rules that can be easily replayed in a new context, rather than specific operations which cannot be generalised. In BOGART, a design plan contains a node for each module. When the module is refined, the node is annotated with the name of the decomposition rule and the values of its parameters and connected to a new child node for each sub-module.

Generalised design rules

BOGART uses LEAP [235] to learn new design object knowledge. In the domain of circuit design, the training example consists of a description of the function to be implemented, a description of the known characteristics of the input signals and a circuit provided by the user to implement the given function for the given input signals. LEAP generalises the specific example into a new refinement rule. By using a variant of explanation-based learning (called verification based learning), LEAP computes a refinement rule precondition (i.e. the left-hand side) by using its theory of circuits to analyse the single training example. LEAP explains (verifies) for itself that the circuit does in fact work. It generalises from the example by retaining only those features of the signals that characterised this class of input signal. LEAP generalises the right-hand side by verifying that the circuit correctly implements the desired function. The verification involves determining the general class of circuits and functional specifications to which the same verification steps

will apply.

Specialised (compiled) design constraint knowledge

Design constraint knowledge is the primary method of detecting design failure [216]. Past design knowledge can be expressed as design expectation rules that relate the key design attribute with the dependent design attributes. When the key design attribute's value is decided, the design expectation rule is triggered. This results in the numerical or symbolic computation of the expected values of dependent design attributes. The expected values of either type (expected value range or expected symbolic value) are compared with the corresponding design attribute values. An *expectation violation* [216] occurs when an inconsistency is noticed between the two set of values resulting in the creation of a DSPL²⁶ expectation violation structure. The information in the expectation violation structure is used by the Generic Object Knowledge Base (GOKB) Reasoner to transform the relevant constraining knowledge in the GOKB into a DSPL constraint. The knowledge compilation process consists of four sequential sub-processes of namely, Reasoner, Transformer, Inheritor and Executor. The Reasoner analyses the role descriptors of the dependent (target) attribute into either potential numeric values or a member of a list of symbolic values and the structural descriptors of the design attributes. Constraints are inherited from the GOKB when a relevant explanation has been found by the Reasoner. The Transformer component supervises the transformation of the relevant GOKB knowledge into a DSPL constraint structure. The newly inherited constraint is tested by the Executor component. A “successful” test ensures that the new design attribute is valid and the newly inherited constraint knowledge is learnt as a DSPL entity for future use.

Chabot and Brown therefore view constraint inheritance as a form of failure-driven learning that transforms a less efficient generalised deep object knowledge into surface knowledge that are highly specialised, tuned and effective for the given design problem.

²⁶ Design Specialists and Plans Language (DSPL) is a domain-independent expert system building language for expressing routine design knowledge.

6.3.6 Abstraction/detailing

Abstraction in empirical equations

Abstraction in empirical equations may become necessary in the event that that no useful empirical equations exist or because not enough attribute values are known. PERSPECT can be used to estimate the values of the unknown attributes. Using their own or PERSPECT's knowledge of design attribute dependency, designers using PERSPECT can define a perspective consisting of unknown attributes and related attributes, generates a viewpoint of experiential knowledge that can be used to find a past design or group of designs similar to the current design, and use associated similar attributes as values for the uninstantiated attributes in the current design.

If the domain model is too complex (i.e. described by too many empirical equations), designers can delete unwanted variables from empirical equations to generate simpler equations which can be used to estimate values of the design model. PERSPECT achieves this capability by the process of *abstraction*. Abstractions of empirical equations mean that dependent attributes can be assigned with fewer required attributes. DESIGNER can be used to determine the least influential input variable of the equation and suggest the variable as most suitable for deleting from the empirical equation.

Design plan abstraction

Both ARGO and DDIS use the knowledge transformation of abstraction to generate design plans. In ARGO, this task is accomplished by computing macro-rules for increasingly abstract versions of the plan and inserting these rules into a partial order according to some abstraction relation. Macro-rules, consisting of relevant preconditions and post-conditions are computed for each plan and stored in a partial order according to an abstraction scheme. These macro-rules are built by compiling through the instances rules of the plan using a variant of explanation-based generalisation (EBG) [236, 237]. The abstraction is accomplished by incrementally merging each set of edge macro rules into a set of cumulative macro rules for previously merged rules. The plan abstraction scheme consists of deleting all of its leaf rules that have no outgoing dependency edges since these leaf rules are those that deal with design details.

At the end of each design session, DDIS abstracts the control knowledge recorded on the control blackboard to one global design plan and several redesign plans. The processes that DDIS used to abstract design plans are as follows:

- All Knowledge Source Activation Records (KSARs) that modified the solution blackboard are identified and unnecessary design steps that led to unsuccessful alternatives or that do not contribute directly to the design process are filtered or removed.
- For each identified major action (KSAR), a case-dependent goal is created in order to prefer the same KS or same type of action in the future.
- The major design actions are classified into design and redesign actions. Design actions are those that lead directly to the eventual solution. Redesign actions are those that are executed when a constraint violation is present on the blackboard. The goals corresponding to design action are grouped into the design plan, while the redesign goals are grouped into redesign plans corresponding to each backtracking episode resulting from constraint violations.
- The intentions of a global plan and case-dependent redesign plan are stated. The intention of a global plan is to generate design values for all the design attributes and to satisfy all the applicable constraints of the design. The intention of a case-dependent redesign plan is to satisfy all unsatisfied constraints that triggered the redesign process.
- The critical constraints that were violated and caused backtracking are recorded so that they can be considered early in future design.

Detailing to reconstruct design history

Unlike BOGART and ARGO which rely on records of past design decisions in the form of a design plan, CDA first reconstructs from a similar solution, a design plan using predefined rules. So while BOGART uses derivational analogy to solve a new design problem, CDA uses a Reconstructive Derivational Analogy (RDA) algorithm to automatically reconstruct the design plan(s) from a large collection of past working [238] design(s). Using the knowledge base of circuit design domain rules, information about the new circuit problem, CDA's Reconstruction Expert finds applicable rules, selects and applies the preferred rule to the current circuit, adding design components each time until the final circuit meets the design requirement. In this process of

detailing, CDA reconstructs the design plan and acquires the rules for the composition of the design.

6.3.7 Explanation/Discovery

DONTE illustrates the learning of control knowledge to explore the design space through the process of discovery about the design space. Through the process of discovery, IDEAL [239] demonstrates learning physical principles of a “concept” description from examples without knowing the target concept a priori. The process of discovery is generally considered to consist of two distinct phases: hypothesis formation and hypothesis testing.

Learning design control knowledge in DONTE

The discovery learning task is initiated by hypothesis formation in which a current hypothesis on the design space is represented as a set of subproblems that are presumed to be independent. Through design decisions made on these independent subproblems guided by control heuristics, interactions of subproblems are discovered and these interactions are aggregated into what is referred to as a macro-decision. The objective of a design decision is to minimise a cost evaluation function that favours certain design solutions (e.g. NAND gates are preferred over other gates). The current hypothesis is updated by the formation of the macro-decision, resulting in a new hypothesis about the design search space. Through this process of hypothesis formation of the design search space, information assimilation through each design decision made and updating the hypothesis, DONTE learns control knowledge to optimally search the design space.

Learning models of physical principles in IDEAL

Using hypothesis formation on past designs structure-behavior-function (SBF) models of physical devices, Bhatta and Goel [239] show how BF models of physical principles can be acquired for future use in design. Discovering physical principles from abstract design models of physical devices is implemented as a learning component of IDEAL (Integrated DEsign by Analogy and Learning).

The models of specific devices (SBF models) provide both the content and constraints for learning the models of physical principles (BF models) by incremental generalisation over design experiences. In particular, Bhatta and Goel

showed that the function of a device determines what to generalise from its SBF model, the SBF model suggests how far to generalise, and the topology of functions indicates what method to use for generalisation. By using content and constraints of the model, IDEAL is able to discover physical principles using fewer examples.

Table 6.3 gives a classification of MLinD systems in terms of the knowledge transformers used to generate new design knowledge and implementation of that knowledge change through the machine learning methods supplemented by other methods (e.g. Pareto optimisation) that results in certain types of knowledge structure.

Table 6.3: Knowledge transformers used in various MLinD systems and the related machine learning or other methods.

Knowledge Transformer	MLinD Systems involved	Machine learning /Other methods	Design Knowledge Represented
Group Rationalisation /Decomposition	• BRIDGES • CONCEPTOR • McLaughlin & Gero • Murdoch & Ball	• ECOBWEB/ EPROTOS • COBWEB • ID3/Pareto • Kohonen neural network/GA	• Hierarchical structure of concept/sub-concept. • Decision tree of rules for Pareto optimum design. • Clusters or archetypes of design solution mapped to performance evaluation space.
Similarity/ Dissimilarity comparison	• BOGART • DDIS	• Derivational analogy • Case-based reasoning	• Design plan as a hierarchical goal structure. • Design plan/history
Association /Disassociation	• NODES	• Semantic links in network.	• Compositional network of concepts. Numerical network of characteristics.
Derivation /Randomisation	• CONCEPTOR • NETSYN	• Concept aggregation. • Modular back-propagation neural network.	• Empirical formula among quantitative attributes. Design patterns among qualitative attributes. • Bayesian a posterior probabilities of design properties represented as network of weights in neural network structure.
Generalisation /Specialisation	• NODES • BOGART /LEAP • DSPL	• Maximal conjunctive generalisation (MCG). • Generalisation using EBL Generalisation. • Knowledge compilation through constraint inheritance.	• Generalised rules of design concepts. • Generalised design rules. • Generalised design plan. Constraint rules.
Abstraction /Detailing	• PERSPECT • ARGO • DDIS • CDA	• COBWEB/ DESIGNER • Merging edge macro rules into cumulative macro by removing leaf rules. • Identify, classify all activated KSARs into two types of design plans. • Reconstructive Derivational Analogy.	• Abstracted empirical equations. • Abstract plan of macro-rules. • A global design plan and several redesign plans. • Detailed design plan built bottom-up.
Explanation /Discovery	• DONTÉ • IDEAL	• Hypothesis formation /hypothesis testing. • Hypothesis formation /hypothesis testing.	• Discovery of macro-decision rule to reduce search. • Discovery of physical principles from abstract design models.

6.4 Types of Triggers for Machine Learning

This purpose of this section is to show evidences of what can trigger learning and when that trigger is likely to occur. Knowing what these triggers are and when these triggers initiate the learning process are important questions that must be answered if machine learning capability is to be incorporated in design support systems. To discuss these triggers by themselves and not relate them to the context of the knowledge learnt and the knowledge transformer involved would not show the relationship between these elements of learning. Sections 6.4.1 to 6.4.3 give some examples of learning design knowledge under different types of triggers: namely the retrospective, in-situ and provisional triggers that are implemented/implicit in MLinD systems reviewed. Since these examples do not represent exhaustively the range of the types of triggers and their related triggering events, Table 6.4 gives a summary of what can trigger learning and when these triggers occur in relation to the knowledge learnt and the knowledge transformer involved in the MLinD systems reviewed.

6.4.1 Retrospective triggers

Retrospective triggers for learning design knowledge can occur at the end of a design process. The source of knowledge for retrospective triggers are past design and their corresponding past design processes. So whilst learning about the design process may occur in-situ, provisionally or retrospectively by analysing the recorded design plan/design history, learning from past designs is triggered only retrospectively (i.e. at end of the design process). The examples of product knowledge learned in retrospect are described in Section 6.2.1.

6.4.2 In-situ triggers

In-situ triggers of learning occur during the design process when design decisions are made. Design decisions are made in relation to the design object or design process. These design decisions may lead to a successful design action or to failure. Learning can occur under such design decisions and actions. Some of these examples of in-situ triggers implemented in MLinD systems are discussed below.

Failure in achieving behavioural specification

Design adaptation is a common practice in conceptual functional design. Design adaptation usually occurs in several phases: in adapting a design retrieved from past design cases to satisfy the new behavioural specification and in diagnosing and redesigning a failed design to achieve the desired behaviour. Thus Ashok and Stroulia consider the design adaptation task as learning [240]. This process of learning takes place in-situ as the design adaptation processes. Ashok and Stroulia identifies three types of diagnosis in design adaptation that could trigger the learning process during design:

- The design does not achieve the desired function of the device. The device fails to achieve the desired function because of incorrect specification of one/more of the components.
- The design results in undesirable behaviour. The undesirable behaviour is due to the under/over specification of the attribute of the component that influences its behaviour.
- The specified structural component in the design misbehaves. The component fails because of the over specification of another component whose behaviour has an adverse effect on the specified component.

They attributed the design failure in the three types of diagnosis to structural causes. Therefore, while knowledge of function to structure ($F \rightarrow S$) mapping is useful for new conceptual design which may be learned retrospectively, knowledge of structure-behaviour-function (SBF) models can be learned in-situ during design adaptation. Each of the three different diagnosis tasks requires different schemes for accessing the internal behaviours that result in the device functions. In particular, task (1) requires the use of design functions as indices into the internal behaviours that result in the device functions; task (2) requires indices from the primary behaviours of the device; and task (3) requires the use of structural components of the device as indices into the internal behaviours in which they play a functional role. KRITIK2's SBF models [240] support all three kinds of indexing schemes. This enables the system to flexibly access the internal behaviours relevant to the current diagnosis task and to thereby localise the diagnostic search.

Violation of design expectations

The DSPL system with constraint inheritance implemented by Chabot and Brown [216] is an example of in-situ triggers which occur when design expectations are violated. Whenever design expectation violations occur, constraint inheritance as a form of failure-driven learning is activated. When an inconsistency is detected in the evaluation of design expectation rules, the relevant design object knowledge is identified for knowledge compilation into new constraint surface knowledge through the constraint inheritance learning mechanism. The new constraint knowledge is then used to test the value of the design attribute for which there is an expectation. The addition of the new constraint knowledge to the existing design constraint knowledge base leads to expectation-failure-driven learning [216].

Customised Viewpoints

The concept of Customised Viewpoint is an example of learning design knowledge in situ. Depending on the design perspective that best suits the designer's current problem solving situation, PERSPECT can generate in-situ multiple forms of implicit experiential knowledge through generalisations of past design information and identifying the most suitable generalisation of past design that support the current *Customised Viewpoints* for the design.

6.4.3 Provisional triggers*Control knowledge*

To explore the design space, DONTE learns by gathering and assimilating information and generalisation during its use. The control knowledge to search the design space is learnt provisionally in anticipation of reducing the complexity of search. This is achieved by examining of a small portion of the design space and the information gained generalised and applied to control the search of the entire design space. DONTE achieved this by aggregating primitive fine-grained sub-problems into larger macro-decisions when evidence gathered during the design exploration suggests that these sub-problems interact.

Failed Constraints Anticipated in Re-design

DDIS uses the blackboard framework to integrate case-based reasoning and domain-based reasoning. Of the knowledge modules/sources in the case-based reasoner, a

module called failure anticipator checks for potential failures (i.e. violation of design constraints) and avoids them in the new design, either by recognising paths leading to dead ends or posting information about constraints that were critical in a previous design. When appropriate, DDIS checks for design constraints during the design process. Whenever constraint violations are found, DDIS's failure anticipator marks them as the major constraints to be check in a new design by placing them as goals on the blackboard. All other constraints are deactivated at this time. This action assures that the constraints most likely to be critical are checked as soon as they become checkable.

Table 6.4: Learning Triggers in relation to knowledge learnt and knowledge transformer involved.

Knowledge Transformer K_t	Knowledge learnt O_k	What triggers learning T_{lw}	When is learning trigger T_{lt}
Group Rationalisation /Decomposition	• Taxonomic knowledge of design concepts. • Clusters of design configuration map to performance evaluation space.	• New concept. • Performance trends in new design.	• Retrospective • Retrospective
Similarity/ Dissimilarity comparison	• Knowledge of design plan. • Knowledge of case-based design plan.	• New but similar design. • New design case.	• In-situ • In-situ
Association /Disassociation	• Compositional knowledge of design concepts.	• New design configuration.	• In-situ
Derivation /Randomisation	• Empirical formula among quantitative design attributes • Design patterns among qualitative attribute-value pairs. • Posterior probabilities of design properties.	• New/updating empirical relationship(s). • New/updating design patterns. • New knowledge of posterior probabilities.	• Retrospective • Retrospective • Retrospective
Generalisation /Specialisation	• Generalised design concepts. • Generalised design plans. • Generalised design rules. • Specialised design constraint knowledge.	• New concept saved. • Module(s) in plan refined. • No existing design rules. • Constraint violation.	• In-situ • In-situ /Retrospective • In-situ • In-situ
Abstraction /Detailing	• Abstracted empirical equations. • Abstracted design plan by removing leaf nodes from plan. • Abstracted from session control knowledge the following: ◆ Global design plan ◆ Related redesign plans ◆ Constraint violations. • Detailed design plan reconstructed bottom-up.	• Non-existence of useful empirical equation or insufficient knowledge of attribute values. • New abstracted design process. • Past design cases to improve design process of similar design(s). ◆ Crucial constraints that triggered redesign process. • No similar design plan existed.	• In-situ/ provisional • Retrospective • Retrospective • Provisional • Provisional
Explanation /Discovery	• Search control knowledge • Models of physical principles	• Optimal design solution • Functional driven design	• Provisional • Retrospective

6.5 Summary

This Chapter introduced and provided evidence for a formalism of learning in design in terms of five elements, namely, input knowledge, knowledge transformers, output knowledge, learning goal and learning triggers. Although it cannot be claimed that

the number of MLinD systems reviewed was exhaustive, the formalism shows that issues in machine learning in design can be studied in a systematic and structured manner that was not apparent in Grecu and Brown's "Dimensions of Machine Learning in Design" [211].

For example, Table 6.2 shows clearly the relationship between the type of input knowledge required and the type of design knowledge learned and the reason for learning that knowledge. That is, given a particular type of input knowledge, what new design knowledge can be learned and why learnt that knowledge. In this manner Table 6.2 not only exemplifies the answers to the questions of "what are the elements supporting learning" and "availability of knowledge for learning" and the "consequences of learning", but it shows that it is just as important to state what the dimensions of learning are and also to know and understand the relationships between these dimensions raised by Grecu and Brown's.

Table 6.3 shows that the methods of learning in Grecu and Brown's dimensions can be categorised into various knowledge transformers presented in this paper. These knowledge transformers represent basic types of knowledge change and provide a basis for classifying the various machine learning methods implemented in MLinD systems.

Table 6.4 not only illustrates examples of what can trigger learning but classifies these triggers into retrospective triggers, in-situ triggers and provisional triggers to provide answers to the question of "what can trigger learning" and "when does learning occur?" These questions are not answered in isolation, but in the context of what is learnt (i.e. the knowledge output) and how it is learnt (i.e. the knowledge transformer involved). Thus far, several answers to these basic questions, and supporting documentation for the formalism for learning in design (LinD) have been derived from a review of published systems in MLinD.

In summary the formalism of learning in design has been derived in [32] through the review of MLinD systems. Five key elements to the learning process have been presented: input knowledge (I_k), knowledge transformers (K_t), output knowledge (O_k), goals/reason for learning (L_g), and triggers of learning (L_t).

Chapter 7

A Model of Learning in Design

That design is inextricably linked with learning has been posited by several researchers of Machine Learning in Design (MLinD). Persidis and Duffy [35] succinctly state the relationship between design and learning:

‘Design as a problem solving activity is inextricably linked with learning.’

Chabot and Brown [216] state that “There should be no argument about the fact that designers learn while designing.” While this fact may be accepted in the MLinD research community, to-date there is a lack of knowledge as to the nature and manner in which this inextricable link exists between design and learning. Thus far, to study the phenomenon of learning in design, the approach adopted was to consider the activities of designing and learning as distinct cognitive activities. This has led to the development of the formalisms for design and learning activities in Chapter 5 and Chapter 6 respectively. The purpose of this Chapter is to propose a model that represents the inextricable link between design and learning. It provides an approach to study and analyse the complex relationships that may exist between design and learning. It is hypothesised that design and learning as cognitive activities are linked in three ways. They are linked at the knowledge level (*epistemic* link). From Chapter 6 it has already been shown that learning can occur retrospectively, provisionally or in-situ with respect to the design activity. They can thus be linked in a *temporal* manner. Both designing and learning are purposeful activities and as such, they are hypothesised to be linked in a *teleological* manner.

7.1 Epistemic link between design and learning activities

Systems (both natural and artificial) of any reasonable degree of complexity can be designed or explained at any number of different description levels (Simon [122], Newell [155], Alberts [186], DasGupta [241]). In fact, multiple description levels constitute a general (indeed, defining) characteristic of all complex systems. Recognising that complex systems can be described by functionally autonomous

levels and yet organised hierarchically, DasGupta [241] extends the notion of description levels to the cognitive system - the mind-brain complex. For the cognitive systems, these widely accepted levels are:

- The knowledge level - wherein, cognition is described or explained in terms of goals, actions, knowledge and intended rational behaviour.
- The symbol level - in which cognitive processes are described or explained in terms of symbols (and symbol structures), memory (in which symbols are held), operations (on symbols), and interpretations (of the operations).
- The biological level - wherein cognition is described or explained in terms of biological structures or structures that are abstractions of biological systems (e.g. neural systems).

The knowledge level is based on Newell's Knowledge Level Hypothesis which states that [242]:

"There exists a distinct computer system level lying immediately above the symbol level, which is characterised by knowledge as the medium and the principle of rationality as the law of behaviour."

According to Newell [155], the Principle of Rationality is stated as follows:

"If an agent has knowledge that one of its actions will lead to one of its goals then the agent will select that action."

Dietterich [243] has shown that learning can be described at the knowledge and symbol level while Kocabas [244] has extended the description to include the device level²⁷. DasGupta develops his computational theory of scientific creativity by showing that knowledge on concepts of micro-programming described at the knowledge level led Wilkes [245, 246] to the creative invention of micro-programming. The development of micro-programming led to an entirely new form or architecture known as the controller unit, a component in the computer responsible for activating its internal operations as it executes a program. It is therefore reasonable to hypothesise that both design activities and learning activities being cognitive activities can be described and operationalised at the knowledge level. As shown in Chapter 5, the knowledge input into the design activity is

²⁷ In device-level learning, the methods and outcomes of learning can only be described in reference to a particular device e.g. connectionist system, neural networks.

transformed into output knowledge through the design action. When learning occurs, the input knowledge and the output knowledge serve as input into the learning activity. Hence, not only are the design and learning activities being described at the knowledge level, it can be seen they are linked through the relationships between the input and output knowledge of the design activity and the input knowledge and output knowledge of the learning activity. Thus it can be said that both design and learning activities are linked at the epistemic level (i.e. at the knowledge level).

7.2 Temporal Link between design and learning activities

In Chapter 6, it was shown that the learning triggers can be activated retrospectively, in-situ and provisionally. It is therefore suggested that the design and learning activities can be linked temporally in three ways. In retrospective learning, the learning activity occurs after the design activity so that knowledge learnt can serve as input knowledge for future design problems. In-situ learning occurs when knowledge is learnt while designing is in progress. In provisional learning, the learning activity precedes the design activity in anticipation of the knowledge required as input into the design activity.

7.2.1 Retrospective learning

In retrospective learning or post design learning, learning is triggered after the completion of certain design activities so that the knowledge learnt is current and most up-to-date and therefore of great utility value for future designs. Past design cases, especially successful ones, are good knowledge sources for retrospective learning. The knowledge learnt is stored as experiential design knowledge in memory. The purpose of learning in hindsight is to learn design patterns that led to the successful designs, relationships in design parameters, to understand and predict underlying trends in classes or types of designs and to extrapolate design attributes of future designs. Feedback from design failures [40] either during testing and commissioning of the product or during the operational life of the product also provides opportunity for retrospective learning. McMahon [247] identified five modes of design change in the maturing of products: they are parameter space exploration, improved understanding of explicit-implicit attribute relationships, change in product design specification, modifying the feasible design space and changing the design

principle. These design changes can lead to the increment or growth of design knowledge of the product and hence the opportunity of learning of that knowledge in retrospect.

Retrospective learning can be triggered by the specific goal of learning. For example, one of the useful consequences of retrospective learning is what Bushby called design transfer [18] or what Duffy et al. [107, 110] called design re-use. Busby [18] calls design transfer as an important practical outcome in design activity. Transfer provides a means of tackling increasing complexity, of limiting risks and costs, and of capitalising on experience.

Retrospective learning is only effective when the output knowledge from the design process is definite. Therefore only knowledge output from certain design activities such as synthesis, quantitative model analysis/quantitative evaluation of final designs, optimisation or full-scale trial can be considered for retrospective learning. The recording of the design process in terms of design rationale and/or design history can be considered for retrospective learning so as to expedite design processes of similar products.

Through a qualitative analysis of 50 unstructured interviews of designers and design managers, Busby classifies inductively design transfer goals as associativity-improving, effort-reducing, error averting and mitigating motivations to innovate rather than transfer and preserve knowledge gained from historical experience.

Busby identified associating-improvement practices as that geared towards the likelihood that a designer would identify a new problem with an existing solution. Some of the practices included a prior research on related previous work, maintain personal hoards of standards and component designs, keep photograph log of past designs, maintain same labelling systems to promote familiarity with previous design and consult prior project plans to stimulate recall of outcomes.

Effort reducing practices would include making modular specifications to avoid

rewriting when reusing a part, the use of standard designs for subsystems where individual differences exist but no strong preference or the use of automated derivations to reduce modification cost of reusing product structures.

The error averting practices have a number of related goals:

- Ensuring that specific, experiential knowledge is transferred between individual designers and subgroups of the design organisation.
- Ensuring that when designs are transferred they are not used inappropriately because the designer has overlooked some specialised element of the transferred design.
- Ensuring that designs that performed poorly are not transferred, or at least are not transferred without modification.
- Ensuring that known, necessary elements of a design are simply not forgotten.

Learning from past designs can help avert errors. When one does something right, one only confirms what is already known: i.e., how to do it. A mistake is an indicator of a gap in one's knowledge. Learning takes place when a mistake is identified, its producers are identified and it is corrected.

Knowledge preserving practices are concerned with avoiding the potentially harmful consequences of reusing the last actual design instead of reusing a canonical generalised design.

These practices identified by Busby provide evidence in support of the existence of retrospective learning in design. The goal of retrospective learning may be error averting practices [18] or effort reducing practices [18] and knowledge preserving practices that may lead to design improvement [248] in terms of cost, efficiency and design quality [39]. Hence it is hypothesised that the design learning activity can occur in retrospect of the design activity or process.

Figure 7.1 presents the manner in which design and learning interact in retrospective learning.

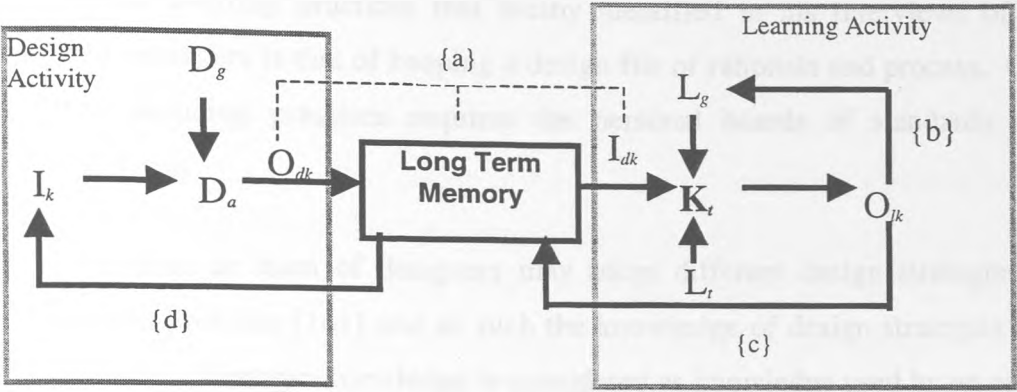


Figure 7.1: Retrospective Learning in Design

The output knowledge, O_{dk} from the design activity (D_a) becomes input knowledge, I_{dk} to the learning activity through the Long Term Memory (LTM) {a}. The output knowledge is directed by the learning goal. The output knowledge of the learning activity has an influence on the goal of the learning activity, L_g {b} in that if the output knowledge does not or partially satisfy the learning goal, then the goal may be modified and a new learning activity takes place. The knowledge gained in the learning activity is stored as experiential knowledge in the LTM {c}. This learned knowledge can then be used in the design activity as the input knowledge, I_k {d}.

7.2.2 In-situ Learning

In in-situ learning, the design activity and the learning activity occur concurrently. That is, while the design activity is in progress, the learning activity can take place.

In their study to investigate what influences the process of searching for design solutions in design organisations, Busby and Lloyd [249] attributed the success of human designers in limiting the effort they devote to search activity for design solution(s) to their ability to learn heuristics from experience. But not all heuristics can be learned in retrospect. Some heuristics must be learned while the design is in process. For example, a single predominant motivation in the conceptual design process provides a way into design without having to list exhaustively all constraints and searching for solutions that meet them [250]. While decision-making is in

progress to select the best design(s), learning of the arguments for or against the designs and final decisions taken can be learned in-situ. Hence, one of the significant error averting practices that Busby identified in his interviews of 50 designers and managers is that of keeping a design file of rationale and process. One of the effort reducing practices requires the personal hoards of standards and component designs.

Different designers or team of designers may adopt different design strategies in solving a design problem [161] and as such the knowledge of design strategies can be learned in-situ. Strategic knowledge is considered as knowledge used by an agent to decide what actions to perform next, where actions have consequences external to the agent. Strategic knowledge is a basis for expertise in domains where the management of problem solving depends upon the choice of actions to take, and this

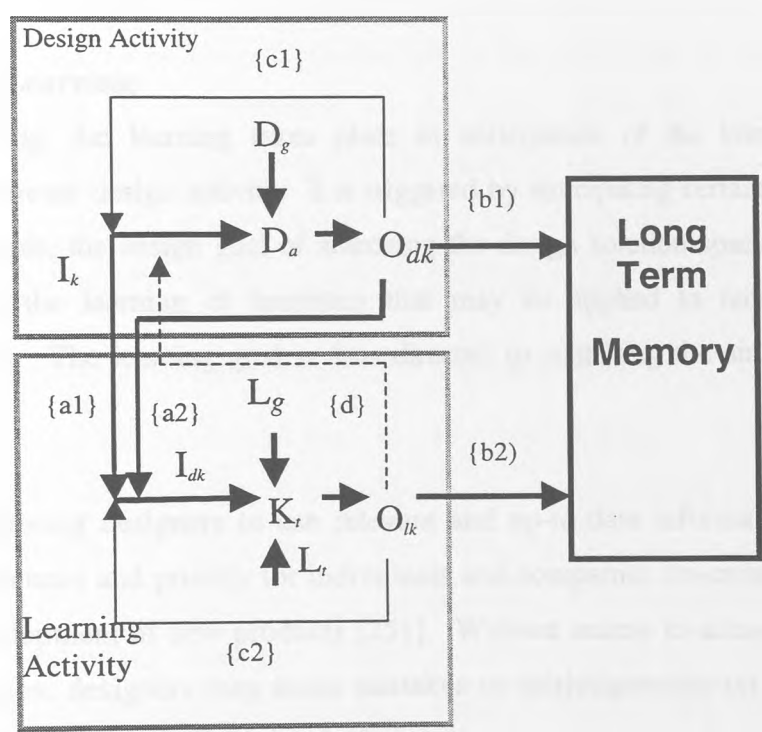


Figure 7.2: In-situ learning

choice is problematic and, therefore, a matter of considerable human expertise. Thus, all these practices are possible only if there is in-situ learning on the part of the

designer or assisted in some way by design rationale capture systems. Hence it can be argued that for certain design activities, learning must occur during the design activities.

Figure 7.2 presents in-situ learning in which the input knowledge, I_k (see {a1}), and output knowledge, O_{dk} , from the design activity, D_a , becomes input knowledge, I_{dk} , to the learning activity (see {a2}). The output knowledge from design activity, O_{dk} is stored in LTM {b1} while the output knowledge from learning activity, O_{lk} likewise stored in LTM {b2}. The output knowledge, O_{dk} , from the design activity may become input knowledge depending on whether the design goal is satisfied {c1}. The output knowledge, O_{lk} , from the learning activity may become input knowledge to the learning activity {c2}. The output from the learning activity may become input into the design activity {d}. Because both the design and learning activities occur simultaneously, this constitutes in-situ learning.

7.3.2 Provisional Learning

In provisional learning, the learning takes place in anticipation of the knowledge required for a downstream design activity. It is triggered by anticipating certain design activities. For example, the design goal of searching the design solution space would trigger provisionally the learning of heuristics that may be applied to reduce the complexity of search. The learning goal is thus directed to achieving the anticipated design goal.

The need for engineering designers to use relevant and up-to-date information has become a major influence and priority for individuals and companies concerned with the design and development of new products [251]. Without access to accurate and up-to-date information, designers may make mistakes or misjudgements on aspects of the product design. Very often the main source of relevant and up-to-date information rests with the suppliers [252, 253] of the latest components, systems or even manufacturing processes and the catalogues they provide [254]. Engineering designers are having to rely heavily upon suppliers for information and expertise throughout the engineering design process [251]. Hence, provisional learning of the

information in anticipation of the needs in the design of the product plays an important role in the development of new products.

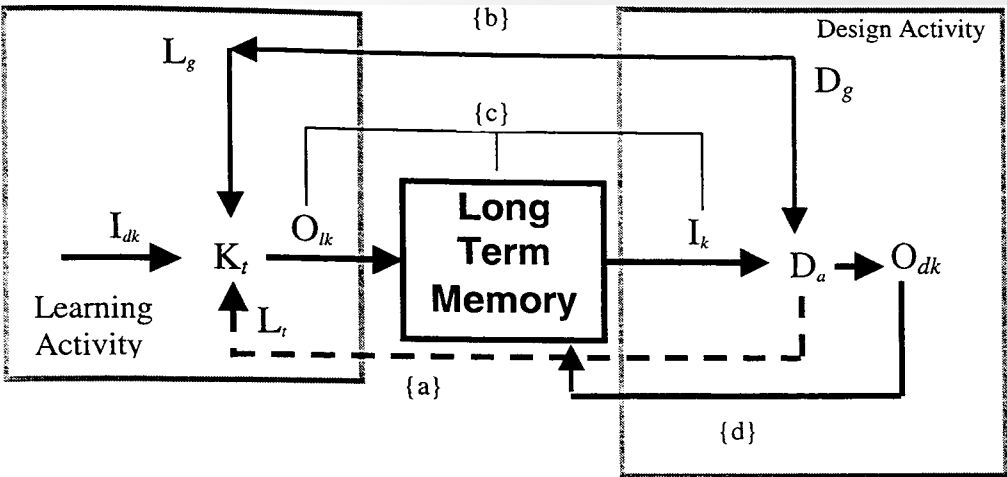


Figure 7.3: Provisional Learning

Figure 7.3 shows provisional learning in which learning is triggered prior to the design activity {a}. The needs of the design activity (i.e. the design goal, D_g) determine the nature and the source of information or knowledge to be acquired. Hence, the learning goal, L_g is dependent on the design goal, D_g {b}. The information or knowledge gained from the learning activity, O_{lk} , is stored in LTM and becomes input information or knowledge, I_k , into the design activity, D_a , {c}. The output knowledge, O_{dk} , from the design activity is stored in LTM {d}.

7.3 Teleological Link through design and learning goals

Both design and learning are considered here as purposeful activities. Each has a specific goal to achieve. But design and learning are inextricably linked in that the learning goal is related to the design goal. For example, the design goal of a preliminary synthesis activity is to reduce the complexity of the conceptual design space and the goal of learning from past design(s)' compositional and taxonomic knowledge of design concepts will expedite the synthesis design activity. The goal of a constraining activity is to reduce the complexity of the design parameter(s) space. The goal of learning knowledge on design constraints by detecting failed constraints or anticipating crucial constraints during design is to streamline the design process. Hence, it is hypothesised that design and learning goals are linked teleologically.

7.4 Summary

A model of learning in design has been presented to suggest the manner in which the design and learning activities are inextricably linked. Three hypotheses are raised here regarding the nature of these links. They are linked:

- at the *epistemic level* in that both design and learning activities are described at the knowledge level of abstraction.
- *temporally* resulting in retrospective learning, in-situ learning and provisional learning.
- *teleologically* through the design and learning goals.

Chapter 8

Protocol analysis of the model of learning in design

The formalisms for the design activity and the learning activity have been presented in Chapter 5 and Chapter 6 respectively. In Chapter 7, it was hypothesised that designing and learning are linked at the knowledge level (i.e. epistemic level), and the goal interactions are teleological in nature and that they are temporally related. There is a need to substantiate the elements of the formalisms proposed for the designing and learning activities by conducting some cognitive experiments. Cross et. al. [255] states: “Of all the empirical, observational research methods for the analysis of design activity, protocol analysis is the one that has received the most use and attention in recent years. It has been regarded as the most likely method (perhaps the only method) to bring out into the open the somewhat mysterious cognitive abilities of designers.”

Audio/video recordings of a designer at work were made to provide the experimental data for the analysis and evaluation of the formalisms for the design activity, the learning activity and the interactions between them. In order to analyse and evaluate the protocol analyses there is a need to ‘design’ a suitable coding scheme that can elicit the elements of the cognitive actions of the designer in terms of designing and learning. Several coding schemes have been studied and evaluated in terms of their suitability. The model based coding scheme was adopted since the formalisms (or models) proposed for design and learning activities serve as the basis of encoding the protocol recorded. This model-based coding scheme is then applied to the protocol and the result of the analysis and the evaluation is described.

8.1 Protocol Analysis

In essence, protocol analysis relies on the concurrent verbal accounts given by subjects of their own cognitive abilities as they perform a task (e.g. designing). It is difficult to imagine how else one might examine what is going on inside people’s heads, other than by asking them to tell us what they are thinking. The assumption made in protocol analysis is that the trace of this “verbalisation” is an accurate reflection of the mental activity that a subject is engaged in. Unlike introspection [256] which may lead to the

subjects theorising or speculating about the processes they undertook, protocol analysis should be an accurate record of the designer's reflection in action or reflective conversation within the situation. Ericsson and Simon [257] in their work on protocol analysis claimed that there is a dramatic increase in the amount of observable behaviour when a subject is performing a task while talking aloud compared to the same subject working under silent conditions. Three major objections have been raised in using verbal reporting as data. They are as follows:

- The issue of veridicality – that the reporting process might alter task performance.
- The issue of completeness – the reports may yield a very incomplete record of the cognitive process.
- The issue of epiphenomenality - the subject may, quite intentionally, give irrelevant accounts, reporting parallel but independent thoughts to those that are actually being employed in the task.

In their seminal work entitled *Protocol Analysis – Verbal Reports as Data*, Ericsson and Simon showed that these objections were unfounded. When information is reproduced in the form in which it was heeded either through talking aloud or thinking aloud, they concluded that the studies gave no evidence that verbalisation changes the course or structure of the thought processes. On the issue of completeness, they marshalled evidence for a simple hypothesis - the information that is heeded during performance of a task, is the information that is reportable; and the information that is reported is the information that is heeded. On the issue of epiphenomenality, they conclude that with great consistency, the experimental evidence they had gathered demonstrates that verbal data are not in the least epiphenomenal but instead are highly pertinent to and informative about subjects' cognitive processes and memory structures. These conclusions are important in that they provide sound rationale in using protocol analysis as a means to evaluate the proposed model of learning in design. This is corroborated by various protocol studies of the design process as evident by the number of such experiments conducted by various researchers recording designer's activities in architectural design [143] and engineering design (i.e. mechanical engineering design [144, 145, 146, 147, 148], electronic design [258]). While most of these studies

involved single-subjects thinking aloud protocol studies, there were also studies that involved cooperative or collaborative team design activities [203, 259, 260].

There is the “Behaviourist” school of thought that learning occurs unconsciously and subjects are not able to report any knowledge about their learning. But Ericsson and Simon [257] in the study of verbal reports on learning argue that a strong *a priori* case can be made that reportable information will always be involved in the learning process. These verbal reports include concept learning, learning to control a complex system (as a computer simulation and in a similar experiment as a real system) and learning of artificial grammar of letter strings. Thus Ericsson and Simon conclude that verbal reports have played and will continue to play a critical role in providing the information necessary for discriminating between different explications of the cognitive processes of learning. Based on Ericsson and Simon findings it is reasonable to propose that learning in design can be studied through protocol analysis of a designer at work. Thus both the cognitive activities of designing and learning can be studied through the method of protocol analysis.

8.1.1 Model of Information Processing

Ericsson and Simon’s model of information processing was used to explain many of the findings of many protocol analyses, not only in terms of the verbalisation of the ongoing cognitive processes, but also in terms of the storage of information in short term memory (STM) and long term memory (LTM). The model of information processing describes the cognitive processes as producing a sequence of states, each marked by a small collection of information in attention and STM. Information in attention and STM imposes constraints on accessed information and cognitive structures, even though it does not always determine uniquely which specific structure will be accessed. The general and basic point is that subjects cannot gain access to information in LTM without bringing that information into STM, hence into attention. The model of information processing is based on several assumptions. These assumptions are listed in the order given below explaining the manner in which verbalisations are the result of heeded thought and reflect the origination and integration of cognitive structures in STM:

1. The verbalisable cognitions can be described as states that correspond to the

contents of STM.

2. The information vocalised is a verbal encoding of the information in short-term memory.
3. The verbalisation processes are initiated as a thought is heeded.
4. The verbalisation is a direct encoding of the heeded thought and reflects its structure.
5. Units of articulation will correspond to integrated cognitive structures.
6. Pauses and hesitations will be good predictors of shifts in processing of cognitive structures.

The methods of protocol analysis rest on a set of assumptions about the general structure of problem solving processes, and about the verbal reporting process itself. The major theoretical assumptions that underlie these methods as given by Ericsson and Simon, reflecting the evolution of the problem solving process with the corresponding verbal reporting process, are:

- The subject's behavior can be viewed as a search through a problem space, accumulating knowledge about the problem situation as the subject goes. This gradual, step-by-step accumulation of knowledge can be represented by a problem behaviour graph, the k th node of which represents the subject's knowledge after k steps of search.
- Each step in the search involves the application of an operator, selected from a relatively small set of task-relevant operators, to knowledge held by the subject in STM. Application of the operator brings new knowledge into STM, moving the subject to a new point in the problem space.
- The verbalisation of the subject corresponds to some part of the information the subject is currently holding in STM, and usually to information that has recently been acquired.
- The information in STM, and reported by the subject, consists primarily of knowledge produced by operators, and symbols representing active goals and subgoals that are driving the activity. A goal may take the form of intent to apply an operator; in which case the protocol may contain explicit evidence for the application of the operators.

The constructs of Ericsson and Simon's model of information processing in terms of operators, knowledge and goals and the manner in which these interact to form new knowledge in the STM, find an interesting parallel in the proposed model of learning in design. The model of learning in design is based on formalisms for design activity and learning activity. These formalisms have similar constructs or elements of input knowledge, output knowledge, design goal and learning goal and the operations of these activities result in the accumulation of knowledge. Thus it can be argued that there is a strong correspondence between the model of information processing and the model of learning in design in terms of their constructs and operation on these constructs. This again affirms that protocol analysis can be used to evaluate the model of learning in design.

8.1.2 Method of Protocol Analysis

The key phases in a protocol analysis method consist of data segmentation, coding of data, analysis of data and interpretation of data [261].

It is now standard procedure to make careful verbatim transcripts of the recorded tapes, thus preserving the raw data in as "hard" a form as could be wished. At the same time, models of information processing of the cognitive processes provides a basis for making the encoding process explicit and objective, so that the theoretical presuppositions entering into that process can be examined objectively.

Protocols are segmented or categorised in some way depending on the purpose of the analysis. The traditional approach has been to segment the original data by inferring the categories that may exist in the recording or part of it and then analyse the frequencies of occurrence in the full record [262]. There is a move away from the traditional approach of analysing protocols towards imposing an externally derived structure on the protocols [262]. Segmentation of the original protocol is not generated by the data but based on other criteria, such as pauses in the flow of words or on semantics/syntactics for recognising discrete utterances or based on some hypotheses that researchers have of the models of the design process [258,123].

8.2 Case Study

In order to study and understand the phenomena of learning in design and to evaluate the proposed model, it was envisaged that the protocol should be conducted in the normal working environment of a designer. A protocol of a Senior Ship Designer with more than ten years of working experience was recorded in a company that provides consultancy service in the design and supervision of construction of high-speed naval crafts and warships. A brief description of the protocol is given below and the transcripts of which can be found in Appendix 1. The protocol took about 2 hours 45 minutes.

The verbalisation and drawing activities of the Senior Ship Designer were recorded using a digital video camera recorder. The AutoCAD system was used to record all the drawings pertaining to the design tasks.

In the protocol, the designer was working on the general arrangement of a 60 metre offshore patrol vessel with a speed of 38 to 41 knots. The vessel is to be armed with weapon systems controlled by a sophisticated fire control system with various surveillance capabilities. At the time of the recording there was a vessel of similar length under construction. So the designer took advantage of the experience gained in the design and construction of a current vessel as he performed the design task of completing the general arrangement (i.e. general layout) of the offshore patrol vessel.

At the start of the protocol the designer was preoccupied with the issue of configuring the engine room compartments with six main engines, gearboxes and shafting system. This resulted in the generation of many concepts of engine layout, each evaluated to ensure that there is room for maintenance of the engines, the routing and removal of engines for major overhaul. This has impact on the location of the engine removal casing which influence the siting of the Harpoon missile systems on the main deck. The other consideration with regards to the engine layout was the shafting installation angle with respect to the ship buttock lines²⁸. This has impact on the head-room available for

²⁸ A buttock line is derived by the intersection of a ships' surface with a longitudinal vertical plane defined parallel to the ship's centre line plane.

the main engines, the clearance of the gearbox sump from the bottom frame structure of the vessel and hence the location of the main engine. The shaft angle installation influences the location of the propeller and the propeller tip clearance²⁹ influences the propeller-induced vibration of the bottom ship structure which tends to increase the ambient noise level of the aft accommodation. The experience gained in fitting a vibration attenuation device on previous vessels built was put to good use in restricting the propeller tip clearance to 15 % of the propeller diameter.

The next major task was the location of the Otto Melara gun at the fore-deck in relation to the superstructure and the accommodation below. The gun should ideally be not too near the superstructure front in order not to restrict the training arc of the gun and too far forward where ship motion is more pronounced. Where the gun is located is also influenced by the need for sufficient room under-deck for the gun magazine and control.

8.3 Segmentation and Coding

Protocol data is very rich in cognitive information but highly unstructured [261]. A video/audio protocol of a design session represents a particular form of qualitative data. It can provide a deep insight into the design process from a perspective of interest leading to a better understanding of the process. However, the protocol and the transcripts are not the actual data on which analysis is performed. Since design involves the application of declarative and procedural knowledge, the purpose of protocol analyses in design has been to elicit the knowledge structure present in the protocols through some coding schemes. Hence a number of coding schemes have been evaluated so that either a suitable coding can be used or a variant of it devised. For a discussion of other coding schemes see Appendix 2.

From Appendix 2, it can be seen that the purpose of imposing structure on protocols through a coding system has been to test existing hypotheses and models of the design process. Wide variations in the categorical structures imposed may actually inhibit comparisons between protocol experiments and the development of theory. Hence

²⁹ A propeller tip clearance is the distance from the tip of a propeller blade to the underside of the hull surface.

Purcell et al [262] use the dual approach of theory generated categories collaborated by data generated categories derived from protocol analysis. Ericsson and Simon [257] posit that encoding and analysing empirical data – whether verbal reports or other kinds of data – always takes place in some kind of theoretical context. Ericsson and Simon therefore advocate the use of model based coding to identify relevant information heeded in a protocol. Based on this recommendation, it is proposed that a coding system based on the formalism of design activity and learning activity developed in Chapters 5 and 6 be formulated for the protocol analysis of experiment reported in this Chapter. No distinction is made here as to whether the coding should be process-based or content-based. This is because both aspects of design are relevant here in terms of the designing activities which are process based in nature and the knowledge change in these activities which is inferred from the information/knowledge content of each activity.

8.3.1 Coding Scheme for Design and Learning Activities

The coding scheme is derived from the understanding of what constitute a design activity and a learning activity. From Chapter 5, the key elements in a design activity are: the input knowledge I_k , the design goal D_g , the design activity D_a and the output knowledge O_k . The elements of the learning activity have been defined as the input knowledge I_k , the learning goal G_l , the learning trigger, knowledge transformer and the output knowledge O_k . Thus in analysing protocol data for the phenomena of learning in design, the coding scheme should incorporate the elements of the design activity, the learning activity and the interaction between them. The coding based on these elements will aid in the identification of the information heeded in the protocol. That is, such a coding scheme will allow the elements hypothesised in the design and learning models to be identified if they exist in the protocol analysis.

Coding scheme for Design Activity

The coding scheme for the design activity consists of the following:

- The nature of the design activity, (Element D_a).
- The input knowledge to the design activity, (Element I_k).
- The output knowledge of the design activity, (Element O_k).
- The goal of the design activity, (Element D_g).

The nature and list of the design activities postulated in Chapter 5 serves as a basis to differentiate the design activities into sub-categories, the list of which and the coding is given in Table 5.1.

Coding Scheme for Learning Activity

The coding scheme for the learning activity consists of the following:

- The types of design knowledge acquired (Element O_k) and the associated learning goal, (Element L_g).
- The knowledge change, ΔK as a result of the difference between input knowledge (Element I_k) and output knowledge (Element O_k).
- The knowledge transformers (Element K_t) and the corresponding machine learning method(s) used in implementation.
- What and when trigger that learning activity (Element L_t).

The knowledge transformers are further sub-categorised and Table 8.1 shows the list and their corresponding codes.

Table 8.1: Codes for Knowledge Transformers

Knowledge Transformers	Codes
Group Rationalisation (or Clustering)/ Decomposition (Ungroup)	Gr / De
Similarity comparison / Dissimilarity comparison	Sc / Dc
Association / Disassociation	As / Ds
Derivations (Reformulation) / Randomisation	Dr / Ra
Generalisation / Specialisation	Ge / Sp
Abstraction /Detailing	Ab / Dt
Explanation /Discovery	Ex / Di
Acquisition	Ac
Modification	Mo

The learning triggers that activate the learning process are categorised into temporal triggers and rational triggers to distinguish when learning is being triggered (temporal) and what triggers the learning (rationale). The temporal triggers are:

- in-situ, L_{ti} to denote learning during the design activity.
- provisional, L_{tp} to denote learning in anticipation of the design activity.
- retrospective, L_{tr} to denote learning in hindsight of the design activity.

The rational trigger is denoted by L_{rw} .

8.3.2 Segmentation into design activities

One way of segmentation is to divide protocols based on verbalisation events such as pauses, intonations as well as syntactic markers for complete phrases and sentences [257]. Pauses or syntactic markers flag the start of a new segment. Van Someren et al [263] describe a process of aggregation of segments into episodes.

Goldschmidt [264] and Gero and McNeill [261] defined a segment based on the subject's intention such that a change in the subject's intention or the contents of their thoughts or their actions flags the start of a new segment. Consequently, a segment sometimes consists of one segment, and sometimes of many sub-segments. These sub-segments are then aggregated in the coding process. A change in intention flags the start of a new segment.

Criteria for segmenting protocol into design activities

Since the purpose of the protocol analysis is to understand the relationships of the elements in the design activity and also in the learning activity and the interaction between them, it is important to first identify the design activities that the designer used in the recording of the protocols. The identification of a particular design activity then serves as the basis to segment the protocol into different segmentations depicting the various design activities that the designer engaged in during the design task. Hence the segmentation of the protocol is based on the intent of a particular design activity (i.e. the context of that particular design activity). However the task of identifying these design activities is fraught with difficulty and ambiguity given the variety and complex nature of the design activities. However there are cues such as key words or phrases uttered by the designer that may suggest the nature of the activity and hence may aid in their identification. Tables 8.2, 8.3 and 8.4 therefore give a summary of the key words, phrases or criteria used in analysing the nature of the design activities in order to identify them.

Table 8.2: Key words and/or criteria used in identifying design definition activity

Segments of protocol	Key words or criteria or sequence context	Design activity implied
• It looks as though there are some space allocation below the mounting for <i>associated</i> electronics. So just need to make a note of the likely area.	Key word 'associated' is used to relate two or more design objects	Associating
• By the time we accommodate propulsors, the auxiliary and maybe weapons systems what little space is left we've to squeeze in logistics requirements of the vessel in terms of the stores that it need to carry to survive at sea ...	The notion of 'breaking down' some design object into its parts	Decomposing
• ... arbitrary put it as a starting point at 150 mm. • So we just nominally barricade 900 mm on the ship's side.	Key phrases like 'put it at'	Defining
• How to get 6 engines, 6 propellers in a row? We could have 6 lines of shafting, 6 separate shafts.	The context in which a problem is stated followed by a proposed solution.	Generating concept
• ... the structure distribution at the bottom of the vessel will probably be of the order of 300 to 350 mm pitch. And chances are we could have a much broader spacing on the main deck. We probably stick to 350 in order to maintain some alignment characteristics.	The intent of maintaining some standard design entity in terms of parameter or value.	Standardising
• So, where do we start to put the engines? Put gearbox and forward set of engines in a forward machinery space the engines will be close coupled to the gearbox minimizing any space take up. ...	The context of establishing relationships between design objects to satisfy some functional requirements	Structuring/ Configuring
• Wherever possible trying to position supporting, shaft supporting equipment close to frame structure knowing that down the line we have good routing for individual equipment.	The context of integrating one subsystem to another so that the whole system functions as a unity.	Synthesis

Table 8.3: Key words and/or criteria used in identifying design evaluation activity

Segments of protocol	Key words or criteria or sequence context	Design activity implied
• 16V 4000 series rating 2320 kW. We will know this will be de-rated to suit ambient air conditions in the theatre of operation. Typically this would be of the order of 2,000 kW, 2000 kW as anticipated. Wow, this is 2720 kW maybe 2,500 kW, about 2,500 kW.	• The notion that design values need not be precisely calculated; an approximation is suffice.	Analysing (approximation)
• And what on the face of it there is sufficient clearance of the gearbox sump from ... it looks reasonably clear. • This is not the best position to site the missile because there is ... • We could find ourselves with keeping slightly more positive foundations for missiles in way of the smaller bolted hatches than what we do have for the machinery hatches. • This would enable the center of the generators to be lifted.	• Explicit or implicit comparisons of alternatives, easily identified by key words of “than” or of results of evaluation like “no,” “yes,” “not”	Evaluating
• So, that means that the shaft angle will have to increase. • These engines can’t move much further aft because we may get out of balance of the boat.	• Key word, “so” suggests decision-making following the sequence of evaluation. It is a command given as a result of decision made.	Decision making –
• Now I need to determine where the baseline of the ship is. • Check what clearance we got available.	• Key words such as “determine” and “check” identified in the activity.	Determining

Table 8.4: Key words and/or criteria used in identifying design management activity

Segments of protocol	Key words or criteria or sequence context	Design activity implied
• The most we can fit in the width of the ship appears to be three at any one time. • Where else can we consider putting the engines? The engine should be put in such a manner that where the propulsion shafting angled with respect to the buttock lines of the vessel in which the shaft line is arranged 6 to 12 degrees maximum in this category. • As the engine removal hatch saddles the full width of the ship it doesn't seem any clear way that we can avoid not putting these missiles on the bulkhead section unless they are placed somewhere else on the ship which may well have to be the consideration.	The notion or statement imposed on the design object or on some relationship between design objects.	Constraining
• This falls into two major areas of selection either 6 engines or 4 engines of a larger capacity. • We must look at how we will put the 6 engines in the vessel. ... There are perhaps one or two ways to put into the vessel sensibly. • It's may be better for us to consider what we do now is going to have an effect on accommodation and how best to organise this accommodation. Let's just see what else have we got. What we'll do is introduce another layout to this arrangement to include the another deck. • There is another thought that we can organise the missiles to be arranged at the forward end of the superstructure. But, this will have the effect of raising or the need to raise the height of the superstructure which again will compromise the optical signature of the vessel. Taking hold of the missiles, let's find out what we can do here.	The intent of this activity is to explore possibilities. The common similar recurring phase of "Let's try and see what it means" or "Let's find out what we can do here?" suggests intention of exploring the design space.	Exploring
• But there are maintenance considerations that we'll have to consider - the ease of the operatives to move about the engine spaces to conduct maintenance in and out of vessel. The ability to get them in and out of the engine space. • Now we've to look at the engineering considerations of the engine attachments to the gearbox. For the forward machinery space the engine is flexibly coupled to the gearbox direct. The engine in the aft machinery space will require different consideration. • This eh could have been overcome at the initial design stage had more attention been paid to the stiffness of the structure. But eh, we are confronted with this problem.	This activity involves the identifying design problem, the requirements of a design (e.g. crew make-up to determine accommodation requirements, list of weapon fit) constraints to apply, statutory rules, list of equipments, particular design item to focus on.	Identifying
• What we must do is communicate with gearbox manufacturers to find if there is anything suitable for this particular application and for this particular power. • Collision bulkhead in table 2.42. Consideration will be given to the proposal for positioning the bulkhead slightly further aft of the arrangement. The craft, blur, blur,.....The minimum is 0.05 length on waterline. The maximum is 0.08.	The notion of communicating with appropriate source for information or acquiring information from design codes.	Information gathering

• All we’re trying to do at this stage is to get some approximate idea of space envelope. How much room do we need to allow for the machinery and once we have decided on that we then can move on to other aspects of the arrangement to see from how we can best fit in those needs. • First of all let’s see about putting Otto Melara Gun.	From a list of design actions, one is selected for immediate focus of attention.	Prioritising
• These enable us to fit propellers with clearances a lot less to the hull than what would ordinarily be, perhaps to avoid vibration being transmitted to the hull.	Identifying conflicting design requirements and a solution to resolve it.	Resolving
• Right, what I got to do is to choose one which have the right torque range. Previous experience has driven us to use the 190 Series which have been renamed to 390 Series. This being a heavy duty application. So we go to the 190 Series.	Key word like “choose”	Selecting
• So I might have to find my drawing. So I need to find my sub-contractor drawing. I need only to draw the bottom contours of the gearbox. It’s only the bottom of the gearbox that is important in the checking of the clearance. Where is my sub-contractor drawing? Over there. • Just so happen in this particular catalogue it doesn’t have what I’m looking for. I need to consult yet another catalogue which is in the library. • So search for library cells for that Gun Otto 76.	The key word or phrase is “to find”, “to look up”, “need to consult yet another”, “was looking at”, “need to look at”, “search for”.	Searching

Using the coding scheme developed in Section 8.3.1 the protocol is segmented and coded based on the design activities identified.

8.4 Results of Protocol Analysis

The purpose of this protocol analysis is to evaluate the proposed model of learning in design in terms of the following hypotheses:

- There exist elements that characterised the design activity (e.g design goal, input knowledge and output knowledge).
- There exist elements that characterised the learning activity (e.g. learning goal, learning trigger, input knowledge, output knowledge and knowledge transformer).
- The design activity and the learning activity are motivated by design goals and learning goals respectively and that the learning goal may be influenced by the design goal (i.e. the teleological link).
- There is a knowledge change that occurs as a result of a design activity and appropriate knowledge transformer(s) is inferred by which this knowledge may be learnt (i.e. the epistemic link).

- The learning activity is triggered retrospectively, in-situ or provisionally with respect to the design activity (i.e. the temporal link).

Hence in this section, the presentation of the results of the protocol analysis is directed at evaluating the model of learning in design in terms of the hypotheses raised.

8.4.1 Identifying elements in the design activity

For each segment of the protocol, the design activity and/or learning activity is identified.

Ericsson and Simon [257] found overwhelming evidence in support of the hypothesis of sequential verbalisation for non-recognition processes. The Hypothesis of Sequential Verbalisation states that:

Information required as input to some process or operation will be verbalised before the output of that operation is verbalised.

In the context of a design activity or learning activity, this implies that information required as input to the activity will be verbalised before the output of that operation is verbalised. This serves as a guide in identifying the elements of input knowledge and design goal in the design and in learning activity.

In some segments, there is only a single statement and thus the semantics of that statement has to be interpreted in conjunction with the preceding statements in order to identify which element it may represent.

The design goal

Design goals are usually expressed as a set of objectives [265]. The objectives could be related to some description of the design (e.g. portability, or some constraints that the design must satisfy or properties that the design must exhibit) or to satisfy some criteria of evaluation of the design (e.g. minimise the level of vibration). Design goals could be related to the intentions of achievement of certain milestones in the design process (e.g. “I want to” or “I need to”). Baykan [266] suggests that the design goal statement may be in the form of question or a statement (e.g. “How to get 6 engines, 6 propellers in row?”). Thus with these possible descriptions of what

design goals might entail, the protocol was analysed to identify the design goals (see Tables 8.6, 8.7 and 8.8).

Since the design goal is usually considered as input into a design activity, based on hypothesis of sequential verbalisation, the design goal can be identified from the input part of the segment of a verbalisation. In the segment of verbalisation under analysis, it is more likely that the design goals will be verbalised first before the designer verbalises how he/she is going to satisfy the goal.

Problem decomposition occurs by setting up sub-goals. As a result there may be a hierarchy of goals and sub-goals. The hierarchy of relationship between goal and sub-goals has been observed in the protocol recorded. For example, before the generation of various concepts to configure six engines, the overall goal was verbalised as follows:

“First things first. We must look at how we can put six engines in the vessel.”

This was subsequently followed by the expression of a sub-goal “How to get six engines, six propellers in a row?”

The input knowledge

The element of input knowledge can be identified from the context of the goal statement and the subsequent statement or even from preceding statement(s). For example, in segment 7 (see Table 8.5), the element of input knowledge of six engines and six propellers comes from the goal statement “How to get six engines, six propellers in a row?” This is followed by the generation of a concept in the next statement “We could have six lines of shafting, six separate shafts.”

Sometimes, the input knowledge may come from the knowledge of past design case(s) that the designer applied to the current design situation. For example, in segment 9, the protocol statement is as follows:

“We know from past installations that it is possible to put in a U-drive gearbox which enable the engine to be placed in reverse arrangement ...” So the use of U-

drive gearbox becomes the element of input knowledge to the design activity for generating code.

The output knowledge

With the identification of the goal statement and the input knowledge, it is conceivable that the element of output knowledge should normally be in the subsequent statement. So with the goal statement “How to get six engines, six propellers in a row?” the element of output knowledge is in the second statement of segment 7 “We could have six lines of shafting, six separate shafts.”

Table 8.5: Coding the segment of protocol into elements of the design and learning activity.

Seg. No	Segment of Protocol	Coding into elements of design activity	Coding into elements of learning activity
7	How to get 6 engines, 6 propellers in a row? We could have 6 lines of shafting, 6 separate shafts. But the practicality is that the vessel could not accommodate 6 propellers in a row, let alone six engines.	•D_g = 6 engines and 6 propellers in a row. D_a = Generating concept. I_k = 6 engines & 6 propellers. O_k = Layout showing 6 engines connected to 6 propellers in a row. •D_g = Is concept feasible. D_a = Evaluating concept. I_k = Layout showing 6 engines connected to 6 propellers in a row. O_k = concept is not feasible as it is not possible to arrange 6 propellers in a row.	Knowledge of constraints on the design space (i.e. space constraint -not feasible to arrange 6 engines in a row). K_r = Derivation T_r = In- situ
8	Can we put 4 engines abreast? A pair of engines driving a single propeller and arrange engines in staggered format - in one area 4 engines abreast and in another area, 2 engines. Inspection of these prospects is shown on screen and as can be seen the ability to put 4 engines in a row is just grossly impractical. There is no room, there is no room for the ship structure, engines have to go in at heights which made them difficult to maintain and access, would raise the CG and the impact of the overall stability of vessel. So, at a glance we can see that this is not a sensible approach.	•D_g = 4 engines abreast. D_a = Generating concept. I_k = 6 engines & 6 propellers. O_k = Layout showing two pairs of engines connected to a propeller each and 1 pair of engines to a propeller in another area. •D_g = Is concept feasible. D_a = Evaluating concept. I_k = Layout showing two pairs of engines connected to a propeller each and 1 pair of engines to a propeller in another area. O_k = concept not feasible because of space constraint.	Knowledge of constraints on the design space (i.e. space constraint -not feasible to arrange 4 engines in a row). K_r = Derivation T_r = In- situ Knowledge of reasons for infeasible concept K_r = Derivation T_r = In- situ
9	So we'll look at another consideration. We know from past installations it is possible to put engine in U-drive gearbox which enable the engine to be placed in reverse arrangement with a stub shaft coming from the engine to the gearbox and then the main propulsion shaft coming out underneath the engine and running aft to the propellers. But this in itself may be a solution but as previously mentioned we can't accommodate 6 propellers.	•D_g = 4 engines abreast. D_a = Generating concept. I_k = 6 U drive connected to gearbox through stub shaft. O_k = Layout showing 6 U drives to drive gearboxes through stub shaft. •D_g = Is concept feasible. D_a = Evaluating concept. I_k = Layout showing 6 U drives to drive gearboxes through stub shaft. O_k = concept is not feasible as it is not possible to arrange 6 propellers in a row.	Application of past design concept to generate a similar concept.
10	The most we can fit in the width of the ship appears to be three at any one time.		Learning of a rule based on evaluation of several design concepts. K_r = Generalisation T_r = In- situ

8.4.2 Identifying elements in the learning activities

Unlike the design activities which can be identified from the context of protocol segments, the same cannot be said of the learning activities. This is because it is not obvious from the protocol segment that learning is occurring or has occurred. Hence this had caused great difficulty in the identification of the elements of the learning activity.

There is however some scope of inferring these elements if segments of the protocol implied the application of knowledge to the current design problem. The application of knowledge implies that knowledge has been learnt in the past; this knowledge is now applied in the current design problem. The identification of these segments of verbalisation which suggests learning in design can be easily identified by statements such as “We know”, “We already decided from previous experience”, “experience gained from previous craft”, “This is a lesson learnt” etc. These phrases (e.g. “we know”) suggest the context in which learning has taken place in the past. Hence, the elements of the learning activity has been derived by implication or inference from segments of the protocol that suggest the application of knowledge learnt.

The learning goal

In general, having identified the portion of the segment that contained the application of knowledge learnt, using the hypothesis of sequential verbalisation, the learning goal and the input knowledge can then be inferred before the verbalisation of the knowledge learnt.

Like the design goals, the learning goals may be identified by phrases such as “we need to,” “we want to” or such similar phrases like “there’s a need for” etc.

The nature of a learning goal is usually influenced by the design goal and the interaction between the two goals is discussed in Section 8.4.3 in which many examples of learning goals are given.

The input knowledge

Just as it is difficult to know when learning has taken place, it is difficult to know

the content and nature of the input knowledge to the learning activity. A similar approach to identifying the learning goal has been used to identify the input knowledge. That is, in a segment of the protocol where application of knowledge learnt has been inferred, that segment has been analysed for input knowledge.

The learning trigger

The retrospective trigger is inferred from the phrases such as “We know” or “experience has shown” suggesting the learning of knowledge from past design cases. The provisional trigger is inferred from the need of acquiring information or knowledge before the design activity can proceed. The in-situ trigger is inferred from the knowledge change that occurs during a design activity which may lead to the learning of new knowledge.

The output knowledge or knowledge learnt

The output knowledge is inferred from the protocol segment which suggests that knowledge has been learnt in the past. This knowledge is now being applied to the current design activity. This is usually expressed in the learning goal statement or in a subsequent statement.

An example in which the knowledge learnt is expressed in the learning goal statement was “But we know that *the magazine can be arranged under-deck*”, the portion in italic expressed the knowledge learnt which then becomes the input knowledge to a current configuring activity which was expressed in an action “so we could pull that down at that level where it’s slightly there ...”

In another example, the knowledge learnt “*It’s normally of 3m square envelope required below the mounting*” was expressed after the learning goal “So just need to make note of the likely area.”

The knowledge transformer

Because of the lack of knowledge of the context and the input knowledge involved in a learning activity, it is not possible to identify the knowledge transformer used for the segment of protocol that implied the application of knowledge. Again, by studying the knowledge change between the input knowledge and the output

knowledge and with the assumption that learning did occur in-situ, the knowledge transformer can be inferred. The details of such inferences are given in Section 8.4.4.

8.4.3 Goal interaction in design and learning activities

It has been postulated that a design activity is motivated by a design goal. Further, it is also postulated that a learning activity is motivated by a learning goal that may be a consequence to the design goal. The design goal could be further decomposed into sub-goals thus giving rise to a hierarchy of goal, sub-goal relationships. Thus in analysing the protocol it is important to identify design goal and sub-goal interactions in the design activities and also the learning goals. The identification of these goals posed significant problems as the designer would normally not explicitly express the design goal or learning goal with such statements like “this is the design goal or that is the learning goal.” Where the goals are not explicitly stated, these goals are therefore inferred from the context of the protocol segments. It should be emphasised that in studying the interactions between design goal and learning goal, the segments of protocol analysed relate to those that involved the application of knowledge. The assumption made is that application of knowledge to the design problem under current focus of attention implies that knowledge has been learnt in-situ or retrospectively.

Tables 8.6, 8.7 and 8.8 show the relationships between the design goal and the learning goal. The design goal and the learning goal inferred from the design segment and the learning segment are shown as italic and bold respectively. It appears that the goals (i.e. design goal and learning goal) interact in two possible ways:

- The learning goal precedes the design goal.
- The design goal precedes the learning goal.

Learning goal precedes the design goal

For design activities of configuring, defining, standardising, identifying, evaluating and decision-making, the learning goal was uttered before the design goal. It may be reasonable to explain that the designer relies on his/her past knowledge and experience in such instances. This is conceivable for activities such configuring,

defining and standardising where there is expedience in relying on past design knowledge. For activities such as identifying, evaluation and decision-making, the designer relies on his/her past successes in making right judgements and assessments.

Table 8.6: Interaction between Design and Learning Goals for Design Definition Activities

Seg .No	Design Activity	Design Goal	Learning Goal
132	Associating	[1] So relying on previous vessel's eh design where same equipment have been used. So just going into that vessel and I am going to extract a drawing. Also I can see on this particular arrangement as to <i>whether there are other considerations towards other associated electronic equipment that may be installed below the mounting.</i>	[2] ³⁰ Emm. It looks as though there are some space allocation below the mounting for associated electronics. So just need to make a note of the likely area. It's nominally of 3m square envelope required below the mounting.
136	Configuring	[2] <i>So we could pull that down at that level where it's slightly there..</i>	[1] But we know that the magazine can be arranged under-deck.
3	Decomposing	By the time we accommodate propulsors, the auxiliary and maybe weapons systems what little space is left we've to squeeze in logistics requirements of the vessel in terms of the stores that it need to carry to survive at sea for tens days.	
23	Defining	[2] So what we have manage to do now is to <i>dictate the position of the propeller centre.</i>	[1] We know from the clearance point of view between the rudder and propeller we want to set the propeller as far aft as possible and clear of the rudder.
54	Defining	[1] <i>So we are looking for something which has something in the region of ...</i>	[2] Again, experience has shown that something in excess of that capability is typically chosen.
11	Generating	[1] <i>We need to fit some sort of gearbox that enables 2 engines to be connected to the one gearbox with a propulsion shaft.</i>	[2] So, we know from commercial catamaran construction in the market place that very narrow hulls eh, have to accommodate a lot of engine installation and we know that there are combined gearboxes that eh, enable so many engines to be joined into one drive train.
141	Standardising	[2] We probably <i>stick to 350 in order to maintain some alignment characteristics.</i>	[1] Right, we know that for our vessel of this type that the structure distribution at the bottom of the vessel will probably be of the order of 300 to 350 mm pitch. And chances are we could have a probably a much broader spacing on the main deck. [3] We know in that deckhouse construction, the aluminium could limit the thickness to 350 mm distribution.
80	Synthesising	Just show the other pedestal bearing. Em, using the frame structure to back it up. <i>Whenever possible trying to position supporting, shaft supporting equipment close to frame structure</i> knowing that down the line we have good, good routing for individual equipment.	

³⁰ [1] indicates the first segment of the verbalisation and [2] indicates the second segment.

Table 8.7: Interaction between Design Goal and Learning Goal for Design Evaluation Activities

Segment No	Design Activity	Design Goal	Learning Goal
20	Analysing	[2] But as an initial design consideration if we work on what we know for ease of quick calculation we can conclude that 1,800 mm diameter is <i>what is necessary to absorb the power</i> and we're assuming propeller speed propeller shaft speed of something in the order of 700 rpm.	[1] What is the other consideration? We know that from separate consideration that the propeller of the order of 1.8 m. This is typically derived by calculation to absorb power of the combined engines . This is taking a typical 3 bladed propeller we know that there are greater advances and more efficiency to be obtained from a 5-bladed propeller or more.
40	Decision Making	[2] <i>so in this new application we'll be looking to doing the same thing.</i>	[1] But, this is overcome by doing away with the bulkhead bearing and the introduction of a second pedestal bearing on what would be the forward side of the bulkhead replacing the bearing and thus taking away from the flexible drive shaft, from the bulkhead any out of balance forces which is connected to the bottom hull foundation. A pedestal bearing which is originally connected is more able to cope with the forces. This is a lesson learnt and ...
95	Determining	[1] So we need to say for a nominal say displacement of 550 tonnes, ...[3] <i>we can determine the design waterline would be.</i>	[2] we know from the past vessels that we've built, we've a set of hydrostatic
115	Evaluating	[2] <i>And a minimum of 4 m in terms of compartment would be required</i> as we know from the experience of previous installation.	[1] One thing that is known about this particular gun is that the under-deck carousel which carries the ammunition is such that there's a need for clearance in order to load ammunition on to it.

Design goal precedes learning goal

For activities such as associating, defining, generating, constraining and exploring, the design goal precedes the learning goal. This may be due to the fact that these activities by their nature as expressed in the design goal encourages the generation of feasible design solutions with minimal constraining. The learning from past experience offers some avenues of feasible solutions. For determining, there is no learning goal expressed but a reliance on past design information. For selecting, the learning from past experience only serves as conformation of the design goal currently being pursued.

Table 8.8: Interaction between Design Goal and Learning Goal for Design Management Activities

Seg. No	Design Activity	Design Goal	Learning Goal
10	Constraining	[1] <i>Where else can we consider putting the engines?</i>	[2] The engine should be put in such a manner that where the propulsion shafting angled with respect to the buttock lines of the vessel in which the shaft line is arranged 6 to 12 degrees maximum in this category.
117	Exploring	[1] So if the gun is placed a little bit further aft it might improve matters. But the further aft it comes it comes to the superstructure. The superstructure will have the effect of restricting the training arc of the gun. Thus limiting the target acquisition area. <i>So again we have to seek a compromise, we got to look at everything that is going on in the boat and see what's good at the end of the day to move things around.</i>	
38/39	Identifying	[2] This eh could have been overcome at the initial design stage had <i>more attention been paid to the stiffness of the structure.</i> But eh, we are confronted with this problem.	[1] And then a flexible Cardan shaft, universal drive shaft was installed at the stub end at the bulkhead and at the gearbox. The out of balance forces from the flexible drive shaft caused vibration resonance at the bulkhead which led to localised cracking of some tank structures that are connected to the bulkhead.
98	Information gathering	Right now the requirement of the rules with regard to collision bulkhead is that the bulkhead be placed not less than 97 % of ship's length and not greater than 94 %. Something of that nature.	
55	Prioritising	All we're trying to do at this stage is <i>to get some approximate idea of space envelope.</i> How much room do we need to allow for the machinery and once we have decided on that we then can move on to other aspects of the arrangement to see from how we can best fit in those needs.	
52	Selecting	[1] Right, <i>what I got to do is to choose one which have the right torque range.</i> [3] This being a heavy duty application. So we go to the 190 Series	[2] Previous experience has driven us to use the 190 Series which have been renamed to 390 Series.
56	Searching	So I was looking at arbitrary 900 mm long and it appears quite satisfactory that I can get for the Cardan shaft application which range from 430 mm long up to 1150 mm if required. So my 900 sits comfortably in the middle. And in terms of the torque carrying capacity, it would suggest that eh 190-55 series is quite suitable.	

8.4.4 Knowledge change in design activities

One of the tenets of the model of learning in design is the knowledge change or knowledge increment that results from the cognitive actions of the designer in the design process. The protocols recorded provide us with a rich source of data to study this knowledge change. The knowledge change could serve as input into the learning activity in which the generation of new knowledge by what is termed as a knowledge transformer may occur. This is achieved by identifying the segment of protocol, the design activity depicted in that segment, the difference in the input knowledge and the output knowledge giving rise to the knowledge change. Based on the characteristics of the knowledge transformers given in Chapter 6, the knowledge transformer that gives rise to the knowledge change is identified. The review in Chapter 6 showed that there are various types of design knowledge that can be learnt. Since Grecu and Brown [40] have a list of types of design knowledge that are preferentially targeted within a learning process that is more comprehensive than that shown in Table 6.2, it is used as a guide to categorise the types of design knowledge that can be learned. The details of segmentation of the protocol, the identification of the elements of the design activity (i.e. the design goal, the input knowledge and the output knowledge), the knowledge change inferred and the corresponding knowledge transformers are given in Appendix 1.

From Appendix 1, the knowledge change, the knowledge transformers and the types of design knowledge learnt for the various design activities in the three categories of design activities (i.e. design definition activities, design evaluation activities and design management activities) are summarised in Tables 8.9, 8.10 and 8.11 respectively.

Knowledge Change for Design Definition Activities

The types of knowledge change as summarised in Table 8.9 relate to the generation of design concepts in the use of the ship space envelope and the reduction of uncertainty and ambiguity in terms of the attributes associated with the design concept.

Knowledge increment for activity of associating

The knowledge transformer Association is commonly used in design activities where components interact with one another. For example, the size of shaft influences and is therefore associated with the size of pedestal bearing to select. There was a need to associate space for magazine and electronic equipment below the *RAM missile mounting*. Hence the knowledge transformer inferred is that of 'Association'.

Knowledge increment for activity of decomposing

The optimal allocation of ship space envelope into various functional spaces (e.g. engine room, accommodation, operations room) motivated the designer's actions. Due considerations must be given to the many requirements and constraints imposed on the design of a warship. The knowledge change associated with decomposition of the ship space envelope into functional spaces was identified at the beginning of the protocol. The appropriate knowledge transformer identified is 'Decomposition'.

Knowledge increment for activity of defining

In order to study the influence of the location of the engines in relation to various factors (e.g. the installation and removal of engines, ease of maintenance, etc.) various design attributes of related design objects (e.g. angle of shaft line) have to be defined. In these cases, the knowledge transformer involved is that of 'Detailing'.

There are instances in which definition of design attributes of a design object are made in consideration with the spatial or geometric relationships of other design object(s). For example, the extent (i.e. the width) of the walk-way around the missile installation on the main deck has to be derived visually in relation the location of the missile installation. Hence the knowledge transformer involved is that of *Derivation*.

In some instances, the definition of design attributes is based on knowledge learnt from previous designs. The decision on where to locate the rudder in relation to the propeller was based on the knowledge of a past design in which the locations of rudder and propeller are known.

Knowledge increment for activity of generating concepts

With the focus on arranging six engines in the engine room, the knowledge change involved generating various concepts for propulsion systems involving the engines and the associated gearboxes and shafting systems. The knowledge transformers identified were ‘*Derivation*’ and ‘*Similarity Comparison*’. Through interacting with the visual display of various concepts to arrange six engines, the feasible engine layout was derived visually. One possible layout was generated based on a past engine room configuration using a “U- drive” through the knowledge transformer, ‘*Similarity Comparison*’.

Knowledge increment for activity of standardising

To ensure that loads imposed on the structure are transmitted effectively, there is a need for adjoining structural members to be aligned. Hence in standardizing the deck longitudinal spacing to that of the bottom longitudinal spacing, structural members connecting the deck longitudinals to the bottom longitudinals can transmit the load effectively. The knowledge transformer is that of ‘*Similarity Comparison*’.

Knowledge increment for activity of structuring/configuring

In configuring a system the components in the system are associated to one another by some relationships (e.g. connectivity relation, part of relation, etc.). So in configuring the shafting system, the association of connectivity of the gearbox to the Cardan shaft, the stub shaft, the pedestal roller bearing, the watertight seal to ensure integrity of the compartment in the engine has to be specified. The knowledge transformer inferred is that of ‘*Association*’.

Knowledge increment for activity of synthesising

In the synthesis activity in which the shafting support structure has to be aligned (integrated) to the bottom structure, the knowledge change was the shaft mounting structure was associated (supported by) the bottom structure. The knowledge change inferred is that of ‘*Association*’.

Table 8.9: Knowledge Change and knowledge transformers for Design Definition Activities

Design activity	Knowledge change	Knowledge Transformer implied/Category of knowledge learned
Associating	• Knowledge that the shaft diameter influences the selection of suitable size of pedestal bearing.	• Association/ Constraint relating design parameters.
Decomposing	• Knowledge that the ship space envelope has to be divided into functional spaces (e.g. propulsion, weapons systems etc.)	• Decomposition
Defining	• Definition of a specific value of the angle of shafting to progress the design • Definition of specific value(s) for design attribute(s) of component (e.g. length, flange diameter, flange thickness of Cardan shaft and stub shaft). • Definition of locations of collision bulkheads using rule requirement. • Definition of locations of fore peak bulkhead using knowledge of positions of collision bulkhead of previous vessel. • Knowledge of arrangement of missiles one bank facing one direction and other in the opposite direction. • Definition of ship's side walkway around the bank of missile. • Definition of clearance around cartridge based on its width. • Definition of space envelope by defining positions of collision bulkheads.	• Specialisation taken from a range of values/Constraint relating design parameters. • Detailing/ Dependencies between design parameters. • Detailing/Design rule • Derivation/ Analogical association • Detailing/ Consequence of design decision. • Derivation/ Dependencies between design parameters. • Derivation/ Dependencies between design parameters. • Association/Analogical association.
Generating	• Knowledge of possible engine and propulsion arrangements. • Knowledge of increasing redundancy by placing 6 engines into 2 compartments. • Knowledge of coupling engine with U-drive gearbox.	• Derivation/ Dependencies between design objects. • Derivation/ Consequence of design decision. • Similarity comparison (based on past design)/ Analogical association.
Standardising	• Standardising the deck longitudinal spacing to be in line with bottom longitudinal spacing based on past design.	• Similarity comparison/ Dependencies between design parameters.
Structuring/ Configuring	• Knowledge of composition of shafting system and its components and their interfaces.	• Association/ Dependencies between design objects.
Synthesising	• Knowledge of integrating shaft supporting equipment to bottom frame structure.	• Association (1)/ Dependencies between design objects.

Knowledge Change for Design Evaluation Activities

Under the category of design evaluation activities, the activities for which knowledge change is inferred are analysing, determining, decision making and evaluating (See Table 8.10).

Knowledge increment for activity of analysing

There are few examples of analysis in this particular protocol. The nature of the analysis is that of analysis by approximation. This is used in evaluating the influence of ambient condition on the rating of the engine. The knowledge transformer is that of *Derivation*.

Knowledge increment for activity of determining

Before evaluation and decision-making, the designer had to determine qualitative or quantitative values of design attributes or properties. Table 8.10 gives various examples of the knowledge change as a result of the activity of determining. The examples invariably involved some calculation and/or measurement of some design parameters associated with the design object or between design objects (e.g. distance of gearbox sump from the bottom frame). The knowledge transformers identified were *Derivation*, *Detailing* and *Similarity Comparison*. In the case of similarity comparison, the design attribute (the design waterline for a given displacement) was based on information obtained from the hydrostatic information of an existing vessel.

Knowledge increment for activity of evaluating

The protocol illustrated many examples of knowledge change as a result of the evaluating activity of the designer. Because of the multi-faceted nature of the design problems detailed in the protocol requiring different perspectives of consideration, the nature of the knowledge change here reflects the nature of the perspectives considered. In general the knowledge increment relates to that of the evaluation criteria that the designer used to evaluate various design concepts, design attributes or properties related to the design. In several of the examples the knowledge transformer identified is *Derivation* (visually). This is due to the nature of the design problem in which the designer had to evaluate the design before him 'visually' through checking of distance (for problems involving clearance between design objects), or compatibility between design objects (mating of flange and coupling). In some instances, the knowledge change involves the explanation of the physical phenomena used in evaluating the design problem. For example, the influence of noise source decreasing with distance is a factor to evaluate the suitability of the accommodation near to the engine or generator rooms. The knowledge transformer

used in this case is *Explanation*.

Knowledge increment for activity of decision-making

The knowledge gained in the decision-making activity gives an insight into the reason or rationale for arriving at the decision. For example, the knowledge that increasing the angle of the shaft line will increase the clearance between the gearbox sump and the bottom structure and so the decision was made to increase the angle of the shaft line marginally. The rationale for moving the engine block and the gearbox up the shaft line is to clear the bottom frame structure. This is derived visually by depicting them on the screen and visually inspecting the outcome of that design decision. So while the knowledge transformer for increasing the angle of the shaft line is *Explanation*, that for moving the engine up the shaft line is *Derivation* (though derived visually).

Based on past installation of a reverse reduction gearbox coupled to a water-jet as the propulsion system on a catamaran, a decision was made to use a reverse reduction gearbox for the propeller application, the feasibility of which had to be confirmed with the gearbox manufacturer.

Table 8.10: Knowledge Change and knowledge transformers for Design Evaluation Activities

Design activity	Knowledge change	Knowledge Transformer implied/Category of knowledge learned
Analysing	•Estimation of the size of propeller based on past design. •Knowledge of influence of ambient condition on rated engine power of catalogue.	•Similarity comparison (1)/ Analogical associations •Derivation (1)/ Dependencies between design parameters.
Evaluating	•Knowledge of space constraint on engine arrangements (i.e. not possible to put 6 engines). •Knowledge to check clearance between gearbox sump and bottom structure. •Knowledge that shaft angle of 5 degree is sufficient to provide clearance between gearbox sump and structure •Knowledge of checking for sufficient headroom for engine installation and removal given the location of engine and gearbox. •Knowledge of evaluating space requirement for officer accommodation and crew accommodation •Knowledge of satisfying a design constraint (feasible shaft angle to buttock line - 8 degrees). •Knowledge of interfacing compatibility between design objects (e.g. flange and coupling) •Knowledge of evaluation criteria for design object (e.g. clearance allowance for pedestal bearing) •Knowledge of evaluation criteria - the influence of noise sources on location of accommodation. •Knowledge of factors in locating generators in relation to hatch size and deck side structure. •Knowledge of factors in evaluating locations to site missile installation (not to site missiles over engine removal hatch but acceptable over smaller generator hatch).	•Derivation visually (3)/ Constraint relating elements of the design. •Derivation visually (3)/Dependencies between design parameters •Derivation visually (1)/ Constraint relating elements of the design. •Derivation visually (1)/ Constraint relating elements of the design. •Derivation (3)/ Dependencies between design parameters •Specialisation (1)/ Constraint relating elements of the design. •Association (3)/ Dependencies between design parameters •Association/ Constraint relating elements of the design. •Explanation (4)/Preferences •Derivation spatially/Consequence of design decisions •Derivation & Explanation/ Consequence of design decisions.
Decision making	•Knowledge of associating reverse reduction gearbox with propeller version based on past design. •Knowledge of relationship between missile and ship's side walkway. •Knowledge that increasing shaft angle can increase clearance between gearbox and bottom structure. •Knowledge that removal of bulkhead bearing and introduction of second pedestal bearing can remove out of balance forces on bulkhead (in retrospect) •Knowledge of alternative site for missile in a less	•Association (2)/ Analogical association •Explanation/ Consequence of design decisions. •Explanation/ Dependencies between design objects. •Explanation in retrospect/ Learning from past failure •Derivation/ Support in

	restrictive location. •Knowledge of rationale for locating one bank of missile over generator hatch. •Knowledge of not siting engine further aft due to trim problem. •Knowledge to allow stub shaft to protrude beyond bulkhead for maintenance accessibility. •Knowledge of moving engine and gearbox up the shaft line in order to clear bottom frame.	favour of a decision. •Explanation (2)/ Consequence of design decision. •Association/ Support in favour of a decision. •Derivation/ Consequence of design decision. •Derivation(2)/ Consequence of design decision.
Determining	•Knowledge of geometric relationship between design objects (e.g. distance of shaft from centreline). •Knowledge of the spatial relationship between design objects (e.g. clearance between gearbox sump and bottom structure, clearance of engine from bulkhead). •Knowledge of equation to calculate design parameter. •Knowledge of design attribute of component (e.g. distance between knuckle points of Cardan shaft, cartridge of roller bearing, dimensions of pedestal bearing). •Knowledge of design attribute (e.g. design waterline for given displacement). •Knowledge of location of collision bulkhead from stem.	•Detailing/ Dependencies between design objects. •Derivation(3)/ Dependencies between design objects. •Derivation/ Design method. •Detailing(6)/ Dependencies between design elements. •Similarity comparison using past design/ Analogical association •Derivation/ Dependencies between design elements.

Knowledge Change for Design Management Activities

The various examples of knowledge change for design management activities are shown in Table 8.11.

Knowledge increment for activity of constraining

Constraining sets limits on the feasible design space. In exploring the different concepts of engine room configuration for six engines, the designer was able to generalised from the visual inspections of the various configurations that at any one time it is not possible to have three engines abreast given the width of the vessel under consideration. The knowledge transformer here is clearly that of *Generalisation* resulting in a rule of thumb that it is not possible to put three engines abreast in the given vessel. In another example of the constraining activity, the designer used a rule of thumb learnt which states that the angle of the shaft line is to be between 6° and 12°. A choice of a particular value of angle of the shaft line (say 8°) would be an example of the knowledge transformer of *Specialisation*.

Knowledge increment for activity of exploring

The optimal location of the main engines in the machinery room is influenced by considerations of ease of installation, removal and maintenance. As the main engine is connected to the gearbox, shafting system and propeller, the angle of the shaft line has significant impact on the location of the main engine in terms of its clearance from the bottom structure and also the engine room main deck structure. So in locating the main engine, the designer explored the effects of the changing shaft angle before arriving at the final location of the engine and gearbox. For example, increasing the shaft angle will raise the engine and the gearbox that in turn will ensure there is sufficient clearance between the gearbox sump and the bottom frame structure. If there is a need to move the engine aft, this can also be achieved by increasing the angle of the shaft line. There is an upper limit to the angle of the shaft line. If there is insufficient clearance of gearbox sump from the bottom frame, the designer can move the engine and gearbox up the shaft line. Because of the interaction between the design objects (engines, gearbox, bottom frame) and the design attributes involved, the knowledge transformer involved in this exploration is '*Derivation*'.

Another example of knowledge increment through the activity of exploring was in exploring the optimal location of the *Otto Melera gun* at the fore deck structure. The gun should be located where the ship's motion is minimal in order to minimise constant correction by the gun's gyro system and yet not too far aft to the superstructure which would limit the firing arc of the gun. Where it is located has an impact on the accommodation in the deck below. In exploring the optimal location of the gun, the transformer inferred is that of '*Explanation*'; the pros and cons were given for deciding the final location of the gun.

Knowledge increment for activity of identifying

Identifying the design requirements from the design specification and formulating the key issues or factors to consider in design often requires good understanding of domain knowledge and experiential knowledge. Thus the activity of identifying provides an avenue for a knowledge increment is evident from the protocol recording of identifying activities. For example, in the protocol the key issue in machinery

room configuration are the factors to consider in deciding the location of the main engines and the propulsion system. The factors include sensible routing for the ease of installation, removal and maintenance of the main engines, which have an impact on the location of the engine removal hatch which, in turn, is constrained by the other equipment or structures on the main deck. The knowledge transformer inferred is that of 'Association'; that is, a set of criteria or factors that influence association with the specific locations of the main engines and the propulsion system.

Knowledge increment for activity of information gathering

In order to progress further in generating a concept of a particular engine room arrangement using combined gearboxes which have been used in a catamaran application, the designer was engaged in the activity of information gathering. In this activity, the designer wished to establish if there were combined gearboxes with power adequate for the particular naval vessel under design. The knowledge transformer is that of 'Acquisition'.

Knowledge increment for activity of prioritising

Prioritising enables the designer to focus on design tasks that may have significant impact on other downstream design tasks. Although the designer may not always verbalise the design tasks on his agenda, nevertheless he did single out important tasks to be accomplished first in the protocol. The knowledge gained from these examples may be useful for future design tasks. In allocating ship space envelope, priority was given to the needs of the machinery space. Having defined details of the shaft, priority was given to the selection of a suitable pedestal bearing. Similarly, among the weapons systems requirements, the priority was to consider the design requirements of the Otto Melara gun.

Except for the design activity of prioritising, it is interesting to note that all suitable knowledge transformers were identified among the seven pairs of knowledge transformers. Hence this protocol analysis highlights a deficiency in the range of knowledge transformers. The appropriate transformer that must be included in the proposed list of knowledge transformers given in Chapter 6 must include the

knowledge transformer pair of 'Sorting/Unsorting'.

Knowledge increment for activity of resolving

Resolving design conflicts involves identifying the conflicts and a solution that removes the conflicts. In one of the protocol segments, there was a need to resolve the conflicting requirements of reducing propeller-induced vibration and providing angle of shaft line that has small inclination to the buttock lines. Reducing propeller-induced vibration requires an increase in the clearance between the propeller tip (i.e. propeller tip clearance) and the bottom structure and this can be achieved by increasing the angle of the shaft line; the conflict arises because increasing the shaft line will imply that the design requirement that the shaft line should have small inclination to the buttock lines might not be satisfied. By using a vibration attenuation cave which reduces propeller induced vibration, the propeller tip clearance was reduced to 15% of the propeller diameter. The knowledge transformer is that of 'Explanation' since it is based on the understanding of the physical phenomenon of absorbing vibration by the elastomeric membrane used in the vibration attenuation cave.

Knowledge increment for activity of searching

The search activity involved either looking for information in the CAD databases or catalogues found in the consultancy company's library. The knowledge gained through the information provided serves to increase the knowledge of the design objects. For example, the search in the catalogue on a product called Cardan shaft identifies which Cardan shaft is suitable in providing the required torque and other design attributes of the Cardan shaft. Similarly, a suitable pedestal bearing was identified and the details of which provided information to progress the design. The knowledge transformer inferred is that of 'Detailing'.

Knowledge increment for activity of selecting

The end result of the selecting activity is a definite design object. A suitable Cardan shaft (i.e. the one with the right torque) was selected from the catalogue on Cardan shafts. In so doing, the designer has a specific knowledge of a particular Cardan shaft. In this instance, the knowledge transformer is that of 'Detailing'.

Table 8.11: Knowledge Change and knowledge transformer for Design Management Activities

Design activity	Knowledge Change	Knowledge Transformer
Constraining	•From three engines configuration, generalised knowledge that the maximum number of engines in a given width of ship is not more than three. •Application of the rule of thumb for angle between shaft line and buttock. •Knowledge locating missiles at bulkhead section rather than at engine removal hatch.	•Generalisation (2)/Design rule of constraint for this particular vessel. •Specialisation /Constraint relating design parameters. •Derivation/explanation(1)/Constraint between design elements
Exploring	•Knowledge of design space for possible engine room configurations. •Knowledge of moving engines aft by increasing shaft included angle. •Knowledge that increasing the shaft angle will raise the gearbox and engine. •Knowledge to explore the distance between the gearbox sump and the frame line. •Knowledge of moving the engine and gearbox up the shaft line for gearbox sump to clear frame •Knowledge of possible locations for generator. •Knowledge that locating accommodation aft is nearer to noise source due to engine. •Knowledge of locating FP bulkhead at half frame •Knowledge of problem structuring in terms of factors that affects the optimal location of the gun (i.e. ship motion and the possibility of constant correction of gun gyro system, training arc of gun). •Knowledge that location of gun has influence on deck accommodation below. •Knowledge that locating missiles at forward end of superstructure (SS) will raise height of SS and optical signature •Knowledge that bulkhead position, superstructure and engine hatch can constrain the location of engine. •Knowledge of locating generators in terms of multiples of longitudinal stiffener spacing for routing of structure.	•Discovery/Feasible design space •Derivation/Dependencies between design parameters. •Derivation/ Dependencies between design parameters. •Derivation/ Dependencies between design parameters. •Derivation/ Dependencies between design parameters. •Derivation/ Dependencies between design parameters. •Derivation/ Feasible design space. •Explanation/Preference •Explanation/ Preference. •Explanation(2)/Support in favour or against a decision. •Explanation/Dependencies between design objects. •Derivation/ Dependencies between design objects. •Derivation/Constraint between design objects •Derivation/Preference
Identifying	•Identified the key issues in locating ship's equipment (i.e. sensible routes for installing and removal of equipment). •Identified other issues including ease of operatives to move around the engine spaces to conduct maintenance. •Identified damage stability criteria that vessel must survived damage of nay two compartments. •Identified Frame 37 to check clearance of gearbox from frame structure. •Identified the problem of insufficient stiffness of bulkhead structure. •Identified the need to configure the structure above the main deck. •Identified the need also to lift engines in and out	•Association (2)/ Factors in favour of a design decision. •Association/ Factors in favour of a design decision •Association/ Design rule •Derivation/ Dependencies •Association/ failure of design object as feedback •Similarity comparison/ •Explanation/ Dependencies

	of vessel in relation to other structures on main deck. • Identified the need to know engine power in order to size the drive shaft. • Identified the noise problem as having an impact on location of crew accommodation due to need to keep generators running when ship is in harbour most of the time as there is no shore supply; generators are hence placed further aft.	between design object • Explanation/ Dependencies between design parameters. • Explanation/ Dependencies between design parameters. Preference on location explained.
Information gathering	• Enquired from gearbox manufacturer regarding the availability of a combined gearbox with reversing capability to which the reply is the one suitable for water-jet application may not have the capability needed for propeller version. • Read from Lloyd's Rule for Special Service Craft regarding rule's requirement for location of collision bulkheads.	• Acquisition/ Analogical association. • Detailing(2)/ Design rule
Prioritising	• In allocating ship space envelope, priority was given to the needs of machinery space. • Having defined details of the shaft, priority was on pedestal bearing. • Having identified the weapons systems requirements, priority was to consider the design requirement of the Otto Melara gun	No suitable knowledge transformer identified³¹/ Preference in design tasks / Preference in design tasks
Resolving	• Learning that vibration attenuation cave reduced propeller induced vibration, the propeller tip clearance is reduced to 15 % of propeller diameter.	• Explanation/Constraints relating design parameters overcome.
Selecting	• From the Catalog on Cardan shaft, the 190 series was selected	• Detailing/ Dependencies between design parameters.
Searching	• Search the CAD database to locate the cell library on a particular gearbox so as to draw the bottom contours of the gearbox. • Search in the library for a particular catalogue on Cardan shaft. • Search to find a Cardan shaft of suitable length and torque carrying capacity. • Search catalogues on pedestal bearing and found a suitable cooper pedestal bearing of correct diameter to suit the shaft • Search the CAD database to locate the cell library for Gun Otto 76. • Search the CAD database on missiles to locate the cell library for RAM missile and key to the cell library	• Detailing/ Dependencies between design parameters. • Detailing(2)/ Dependencies between design parameters. • Detailing/ Dependencies between design parameters. • Detailing/ Dependencies between design parameters. • Detailing/ Dependencies between design parameters. • Detailing(2)/ Dependencies between design parameters.

Summary

Having identified the knowledge change that occurred as a result of various design activities, the corresponding knowledge transformers have been inferred from the nature of the knowledge change. Although knowledge change in itself is not learning *per se*, it is the hypothesis of this thesis that knowledge change and the

corresponding transformer could be the basis of learning either by the human designer or by a system.

From Tables 8.9, 8.10 and 8.11 it has been observed that there is a relationship between design activities and knowledge transformers. Hence, from these Tables a mapping between design activities and knowledge transformers for the three categories of design activities (i.e. design definition activities, design evaluation activities and design management activities) are derived and summarised in Tables 8.12, 8.13 and 8.14. The mappings show that for each design activity, there is a corresponding set of knowledge transformers. It is on this basis that it is inferred and shown that design activities and learning activities are inextricably linked. This inextricable link between the design activity and the learning activity is referred to here as an epistemic link.

The mapping between design activities and the knowledge transformers gives an insight into how design knowledge can be learnt by design support systems. That is, the knowledge of the mapping between design activities and knowledge transformers will ensure that the appropriate knowledge transformers and the associated machine learning methods are called upon in supporting designers when they perform a specific design activity.

Table 8.12: Mapping between design definition activities and knowledge transformers

Design activity	Knowledge transformers
Associating	Association.
Decomposing	Decomposition.
Defining	Association, Derivation, Detailing, Specialisation.
Generating	Derivation, Similarity comparison.
Standardising	Similarity comparison.
Structuring/Configuring	Association.
Synthesising	Association.

³¹ The pair of knowledge transformers identified for this design activity is *Sorting/Unsorting*.

Table 8.13: Mapping between design evaluation activities and knowledge transformers

Design activity	Knowledge transformers
Analysing	Derivation, Similarity comparison.
Decision making	Association, Derivation, Explanation.
Determining	Association, Derivation, Detailing, Specialisation.
Evaluation	Derivation, Detailing, Similarity comparison.

Table 8.14: Mapping between design management activities and knowledge transformers

Design activity	Knowledge transformers
Constraining	Derivation/explanation, Generalisation
Exploring	Derivation, Discovery, Explanation.
Identifying	Association, Derivation, Explanation, Similarity comparison.
Information gathering	Acquisition, Detailing.
Prioritising	Sorting, Unsorting
Resolving	Explanation
Selecting	Detailing
Searching	Detailing

8.4.5 Triggers of learning

Four types of triggers have been identified in the protocol segments. Three triggers are temporal in nature and they are retrospective, in-situ and provisional; these indicate when learning takes place. The fourth denotes what triggers learning is referred to as rationale trigger. The four triggers of learning are not explicitly stated in the segments of the protocol. These have to be inferred from the context and semantics of the various verbal segments. Learning takes place visually when the designer interacts with the objects being drawn or with objects already drawn.

Retrospective triggers

Retrospective triggers can be identified by the phrase “we know” or “from previous design” or “from our experience”. The word “know” can be defined as:
“to have knowledge of (something), especially as a result of personal experience; have (information) in the mind; to have learnt (and be able to do so); to be familiar with (a person, place etc); to be able to recognise.”³²

Thus Tables 8.15, 8.16 and 8.17, show examples of verbal segments that suggest that retrospective triggering of knowledge is learnt from past design cases. These Tables also show the interaction between knowledge learnt in retrospect and applied in a

³² From Longman Dictionary of Contemporary English, 1988.

particular design activity.

For activities such as searching, there is no example of learning in retrospect. It is concluded that this is because the nature of these activities is such that the designer has to interact with the design object that is currently under focus of attention.

Table 8.15: Protocol segment of knowledge learnt in retrospect and applied in design definition activities

Design activity	Knowledge learned in retrospect	Application of knowledge	Seg No.
Associating	<i>It looks as though there are some space allocation below the mounting for associated electronics.</i> So just need to make a note of the likely area. <i>It's nominally of 3m square envelope required below the mounting.</i> Let's just check in the schedule to see whether compartment 91 - CIWS equipment room. It's the same. This is an optional fit. It's either CIWS or Rolling Air Frame.	This is an example of design re-use of a past design case and the recognition of the space requirement for associated electronics.	132
Defining	We <i>know</i> the rudder maybe perhaps a similar size maybe slightly different. We <i>know</i> from a clearance point of view between the rudder and propeller we want to set the propeller as far aft as possible and clear of the rudder. We use all the advantages that we have on the <i>previous boat</i> . So what we have manage to do now is to dictate the position of the propeller centre.	Knowledge of the size and location of the rudder used in a previous craft is now used to defined the size and location of the rudder in relation to the current design	23
Generating	We <i>know</i> from <i>past installations</i> it is possible to put engine in U-drive gearbox which enable the engine to be placed in reverse arrangement with a stub shaft coming from the engine to the gearbox and the main propulsion shaft coming out underneath the engine and running aft to the propellers.	Knowledge learned of past U drive gearbox installation used in generating concept for engine room layout	9
Synthesising	I think we can feel reasonably confident with the arrangement as it stands now. Just show the other pedestal bearing. Em, using the frame structure to back it up. Whenever possible trying to position supporting, shaft supporting equipment close to frame structure knowing that down the line we have good, good routing for individual equipment.	Application of the principle of integrating the equipment mounting structure in alignment with bottom structure which serves as the foundation.	82

Table 8.16: Protocol segment of knowledge learnt in retrospect and applied in design evaluation activities

Design activity	Knowledge learnt in retrospect	Application of knowledge	Seg No.
Analysing	We know that from separate consideration that the propeller of the order of 1.8 m. This is typically derived by calculation to absorb power of the combined engines. {by calculation}. But as an initial design consideration if we work on what we know for ease of quick calculation we can conclude that 1,800 mm diameter is what is necessary to absorb the power and we're assuming propeller speed propeller shaft speed of something in the order of 700 rpm.	Based on past design of 3 bladed propeller, 1.8 propeller diameter is considered suitable. As an approximation without detailed calculation, a 1.8 m diameter is used here.	20
Evaluating	One thing that is <i>known</i> about this particular gun is that the under-deck carousel which carries the ammunition is such that there's a need for clearance in order to load ammunition on to it. And a minimum of 4 m in terms of compartment would be required as we <i>know</i> from the <i>experience of previous installation</i> . Right. Based on existing bulkhead positions, the forward most compartment was 5m long. So that would appear to be more than sufficient.	Knowledge of the space requirement learnt from past installation is used in evaluating the space required for the under-deck carousel of the gun.	115
Decision-making	But, this is overcome by doing away with the bulkhead bearing and the introduction of a second pedestal bearing on what would be the forward side of the bulkhead replacing the bearing and thus taking away from the flexible drive shaft, from the bulkhead any out of balance forces which is connected to the bottom hull foundation. A pedestal bearing which is originally connected is more able to cope with the forces. This is a <i>lesson learnt</i> and so in this new application we'll be looking to doing the same thing.	The bulkhead bearing in a previous installation give rise of out of balance forces. This was overcome by removing the bulkhead bearing and providing a second pedestal bearing on the forward side of the bulkhead. This arrangement will be used in the current design.	40
Determining	So we need to say for a nominal say displacement of 550 tonnes, we know from the past vessels that we've built, we've a set of hydrostatic we can determine the design waterline would be.	This is an example of design re-use of the hydrostatics of a past vessel. To generate it for current design would require time.	95

Table 8.17: Protocol segment of knowledge learnt in retrospect and applied in design management activities

Design activity	Knowledge learnt in retrospect	Application of knowledge	Seg. No.
Constraining	Where else can we consider putting the engines? The engine should be put in such a manner that where the propulsion shafting angled with respect to the buttock lines of the vessel in which the shaft line is arranged 6 to 12 degrees maximum in this category.	It can be implied that the angle of the shaft line of 6 to 12 degrees maximum has been generalised from past cases and used here to set limits on the shaft angle to investigate.	17
Exploring	At best this FP bulkhead could move $\frac{1}{2}$ frame as we're likely to have half frames in the fore end of the boat. Because it's getting so fine and we can't practically run longitudinals and won't be able to add any more structure than what we have there in any case because of the shape of the vessel.	Because of the fineness of the hull form in the bow region, the designer recognised from past experience that this region is usually framed transversely at half frame interval.	104
Identifying	So relying on <i>previous vessel's eh design where same equipment have been used</i> . So just going into that vessel and I am going to extract a drawing, also I can see on this particular arrangement as to whether there are other considerations towards other associated electronic equipment that may be installed below the mounting. So, see to do that.	In identifying design consideration for RAM missile, re-use of past drawings brought to attention that associated electronic equipment may be installed below the missile mounting.	130
Information gathering	Combined gearboxes are used in water-jet application for commercial catamaran	Can combined gearbox be used for conventional propeller version.	13
Resolving	And <i>more recently, another innovation</i> which is the vibration attenuation caves which are fitted directly over the propellers. These enable us to fit propellers with clearances a lot less to the hull than what would ordinarily be, perhaps to avoid vibration being transmitted to the hull. With this facility or characteristic we assume a propeller tip clearance of 15% of propeller diameter to work from.	To minimise propeller-induced vibration, a high propeller tip clearance is usually used. But with the experience gained in reducing vibration using a vibration attenuation cave fitted directly over the propellers, a reduction in tip clearance is possible hence overcoming a conflicting requirement. A tip clearance of 15 % is used for this new installation.	22
Selecting	Right, what I got to do is to choose one which have the right torque range. <i>Previous experience has driven us to use the 190 Series</i> which have been renamed to 390 Series. This being a heavy duty application. So we go to the 190 Series	In selecting a suitable Cardan shaft that can delivered the torque range, experience of similar design led to the choice of 190 series.	52

In-situ triggers

In-situ triggers of learning suggest that learning occurs while the designer is interacting with the design. The protocol provides many evidence of in-situ learning.

In attempting to arrange six engines in the machinery space the designer generated

many concepts as to how these engines can be configured. In three of the concepts generated, by visually inspecting the arrangement of the engines on the screen, the designer inferred (or learnt) that these solutions were not feasible. The first concept generated had six engines in a row, each connected to a shaft and propeller. In the second concept, four engines, driving two propellers, and two engines driving one propeller, are configured. The third concept had six engines connected to U-drive gearboxes, driving propeller shafts. These three concepts are not feasible because it is impossible to arrange four, let alone six engines, in the confines of the width of the vessel.

The knowledge of evaluation result of the first concept that used six engines and six propellers was applied in the evaluation of the third concept using six propellers but connected to 'U-drive' gearboxes. The details of the protocol segment clearly illustrate this application of knowledge; the segment is 'But this in itself may be a solution but as previously mentioned we can't accommodate 6 propellers'. This implies that the knowledge of the evaluation result of the first concept has been learnt in-situ and applied in the evaluation of the third concept. In general, this implies that the segments of the protocol illustrate that learning can occur in-situ as the design progresses.

The three infeasible solutions led the designer to generalise in the statement:

"The most we can fit in the width of the ship appears to be three at any one time."

This illustrates an example of learning a generalisation in-situ.

The details of the protocol segments and their coding illustrating in-situ learning discussed above are given in Table 8.18. It will be too lengthy to state all of the examples of learning in-situ here. One can refer to Appendix 1 for details of these examples.

Among the examples of the learning in-situ, it is possible that some of the knowledge increment will transform into long term experiential knowledge and some may be considered as transient knowledge used in the problem-solving and the evolution of the design to a final solution [24]. For example, a generalised rule of limiting the

number of engines to three in the width of this particular vessel is an example of experiential knowledge learnt in-situ that could be reused in later design scenarios of vessels of similar width. The infeasible solution concepts would represent the transient knowledge learnt during the concept generation.

Table 8.18: Protocol segments and coding of examples of in-situ learning

Serial No.	Segment of Protocol	Interpretation of segment
7	How to get 6 engines, 6 propellers in a row? We could have 6 lines of shafting, 6 separate shafts. But the practicality is that the vessel could not accommodate 6 propellers in a row, let alone six engines.	•The sub-goal of generating a concept to accommodate 6 engines was given followed by a possible arrangement •The phrase “<i>the vessel could not accommodate 6 propellers in a row</i>” indicates a negative evaluation of the concept.
8	Can we put 4 engines abreast? A pair of engines driving a single propeller and arrange engines in staggered format - in one area 4 engines abreast and in another area, 2 engines. Inspection of these prospects is shown on screen and as can be seen the ability to put 4 engines in a row is just grossly impractical. There is no room, there is no room for the ship structure, engines have to go in at heights which made them difficult to maintain and access, would raise the CG and the impact of the overall stability of vessel. So, at a glance we can see that this is not a sensible approach.	• An alternative arrangement using 4 engines abreast and 2 engines was on the screen – another possible concept. •The negative evaluation of the concept is summarised by the phrases “<i>4 engines in a row is just grossly impractical</i>” and “<i>this is not a sensible approach</i>”. Reasons for rejecting this concept was also given.
9	So we’ll look at another consideration. We know from past installations it is possible to put engine in U-drive gearbox which enable the engine to be placed in reverse arrangement with a stub shaft coming from the engine to the gearbox and then the main propulsion shaft coming out underneath the engine and running aft to the propellers. But this in itself may be a solution but as previously mentioned we can’t accommodate 6 propellers.	• A possible configuration based on a past installation is suggested in which a u-drive gearbox is connected to the engine. •The phrase “<i>as previously mentioned we can’t accommodate 6 propellers</i>” suggests in-situ learning of a negative assessment of an earlier concept and the knowledge is now applied to the current concept.
10	The most we can fit in the width of the ship appears to be three at any one time.	Having considered the possible concepts to be infeasible, this statement suggests a generalisation of a rule of thumb for this particular type of vessel.

Provisional triggers

The examples of provisional triggers of learning are not as numerous as examples of retrospective or in-situ learning.

Table 8.19 illustrates a few examples.

In the first example, the price per kilowatt of the four engines and six engines was learnt provisionally before evaluating the two options. In the second example, although a combined gearbox had been used in the water-jet application, an enquiry was made provisionally regarding the availability of a combined gearbox with reverse capability for that of propeller version of propulsion. The information or knowledge learnt provisionally enabled the designer to pursue the concept of using combined gearbox in the propeller version further.

Table 8.19: Protocol segment and coding of examples of provisional learning

Seg. No.	Segment of Protocol	Interpretation of segment
2	This falls into two major areas of selection either 6 engines or 4 engines of a larger capacity. We're advised that the \$/kW price of the 4 engines applications is prohibitive and that we should concentrate our effort on 6 engines application.	• Two possible engine configurations based on either 4 or 6 engines are being explored. • The phrase "We're advised" suggests that there was an activity of information gathering which results in the finding that the \$/kW price of the 4 engines is prohibitive compared with the 6 engine configuration. The information gathering was done prior (i.e. provisionally) to the decision to concentrate effort on the 6 engine application.
12	What we must do is communicate with gearbox manufacturers to find if there is anything suitable for this particular application and for this particular power.	• The phrase "What we must to is communicate with" suggest an activity of information gathering provisionally to find out the suitability of a particular gearbox (the combine gearbox in this case) for a particular application and a given engine power.
13	We know that these are typically available for water-jet application. What needs to be done is to establish if reverse reduction capability for conventional propeller arrangements. The gearbox manufacturer has come back to us and has made a recommendation but it's only a tentative one as it happen to be a gearbox suitable for water-jet which doesn't have a reverse requirement eh, because for reversing in water-jets, manipulation of the reversing bucket over the jet nozzle gives the reversing capability.	• The phrases "What needs to be done is to establish.." and "The gearbox manufacturer has come back to us and has made a recommendation .." suggest that there is an activity of information gathering provisionally prior to applying the combined gearbox which has been used in water-jet application to the propeller version which requires a reversing capability.

Rationale learning trigger

Rationale trigger denotes the reason why learning is triggered. The reasons identified were novelty driven, expedience driven, conflict avoidance and failure avoidance. Examples of these are shown in Table 8.20 below.

The first example of rationale learning trigger is *novelty driven*. The designer knew of the application of combined gearboxes connected to waterjets in the narrow hulls of commercial catamaran design. Being faced with a similar problem of space constraint in the design of a conventional hull, he was led by the novelty of the idea to inquire more knowledge from the combined gearbox manufacturer whether combined gearbox can be used in conventional propeller design.

The second example of rationale learning trigger is *expedience driven*. As the designer had to progress his shafting arrangement, which is influenced by the position of the propeller and propeller size, he needed to know the diameter of the propeller for the current design which absorbed similar horsepower at a similar RPM. By learning about the diameter of a similar propeller design, he did not have to perform detailed propeller design calculation and thus was able to progress the design.

The third example of rationale learning trigger is driven by conflict avoidance. A recent innovation of using vibration attenuation cave (VAC) fitted directly over the propellers reduces propeller-induced hull vibration. The common approach to reduce propeller-induced hull vibration is to increase the propeller tip clearance (i.e. increase the distance of the propeller tip from the hull). This approach however, is in conflict with the requirement to maintain low angle of shafting line as the increase in propeller tip clearance will lead to an increase in the angle of shafting line. So by learning from the experience of the installation of VAC, the designer was able to resolve conflicting design requirements.

The fourth example of rationale learning trigger is driven by failure avoidance. Tank structures connected to the bulkhead had localised cracking. This was caused by

out-of-balance forces from the flexible drive shaft giving rise to vibration resonance at the bulkhead. The problem was solved by the removal of the bulkhead bearing and the introduction of a second pedestal bearing and hence this removed the source of the out of balance forces. When faced with a similar design situation, the designer learned from the design solution what is the modified arrangement to support the flexible Cardan shaft in order to avoid a similar design failure in the tank structure.

Table 8.20: Protocol segment and inference of examples of rationale triggers

Seg No.	Protocol segment	Interpretation
11	We need to fit some sort of gearbox that enables 2 engines to be connected to the one gearbox with a propulsion shaft. So, we <i>know</i> from commercial catamaran construction in the market place that very narrow hulls eh, have to accommodate a lot of engine installation and we <i>know</i> that there are combined gearboxes that eh, enable so many engines to be joined into one drive train.	This is an example of rational learning trigger that is <i>novelty driven</i>. The designer knew of the application of combined gearboxes in narrow catamaran hulls and thought that the concept can be used for the current design where space is a constraint which resulted in him inquiring the gearbox manufacturer whether the combined gearbox can be used in conventional hull.
20	What is the other consideration? We <i>know</i> that from separate consideration that the propeller of the order of 1.8 m. This is typically derived by calculation to absorb power of the combined engines. {by calculation}. This is taking a typical 3 bladed propeller. We <i>know</i> that there are greater advances and more efficiency to be obtained from a 5-bladed propeller or more. But as an initial design consideration if we work on <i>what we know</i> for ease of quick calculation we can conclude that 1,800 mm diameter is what is necessary to absorb the power and we're assuming propeller speed propeller shaft speed of something in the order of 700 rpm.	This is an example of rational learning trigger that is expedience driven. The designer had to progress his shafting arrangement which is influenced by the location of the propeller and the size of the propeller. He needed to know the diameter of the propeller without resorting to detailed calculation. By learning from the case of similar propeller design, he applied the knowledge learnt of the propeller diameter to expedite the progress of the design.
22	And more recently, another innovation which is the vibration attenuation caves which are fitted directly over the propellers. These enable us to fit propellers with clearances a lot less to the hull than what would ordinarily be, perhaps to avoid vibration being transmitted to the hull. This attenuation panel made from elastomeric membrane is suspended from watertight box directly in way of each propeller. With this facility or characteristic we assume a propeller tip clearance of 15% of propeller diameter to work from. What is the membrane face (i.e. the elastomeric membrane face) to what is the centreline of the propeller? This gives us a datum point to the aft end of the vessel.	This is an example of rationale trigger that is driven by conflict avoidance. The requirement to reduce propeller induced vibration without increasing the propeller tip clearance triggered what was learnt regarding vibration attenuation caves (VAC). Hence the VAC was applied to the current design problem to resolve the conflict of maintaining small shaft angle and the need to reduce hull vibration which is normally achieved by specifying higher propeller tip clearance.
38, 40	And then a flexible Cardan shaft, universal drive shaft was installed at the stub end at the bulkhead and at the gearbox. The out of balance forces from the flexible drive shaft caused vibration resonance at the bulkhead which led to localised cracking of some tank structures that are connected to the bulkhead. But, this is overcome by doing away with the bulkhead bearing and the introduction of a second pedestal bearing on what would be the forward side of the bulkhead replacing the bearing and thus taking away from the flexible drive shaft, from the bulkhead any out of balance forces which is connected to the bottom hull foundation. A pedestal bearing which is originally connected is more able to cope with the forces. <i>This is a lesson learnt and so in this new application we'll be looking to doing the same thing.</i>	This is an example of rationale trigger that is driven by <i>failure avoidance</i>. The lesson was learnt of localised cracking of some tanks connected to the bulkhead structure caused by vibration resonance arising from the out of balance forces of the flexible drive shaft at the bulkhead. By learning how the problem of localised cracking was solved, he applied the solution to the current design.

8.5 Summary of results of protocol analysis

In the proposed model of learning in design, it has been hypothesised that design and learning are inextricably linked at the teleological level, at the epistemic level and temporally.

In Sections 8.4.1 and 8.4.2 it has been shown that the elements of a design activity and the elements of a learning activity can be identified in the segments of protocols. However, it is not possible to identify the elements of the learning activity; they were inferred from the protocol segments.

In Section 8.4.3 it has been shown that learning and designing are linked at the teleological level. This was achieved through the identification of the elements of the design goal and learning and the analysis of the interaction between them. It was not possible to identify both design goals and learning goals in all the segments of the protocol as these elements are not explicitly stated. Nevertheless in segments where they are identified, the evaluation of their interaction suggests that these goals interact in two manners which are:

- The learning goal precedes the design goal.
- The design goal precedes the learning goal.

It has also been shown in Section 8.4.4 that the protocol segments of various design activities can be analysed into the elements of input knowledge and output knowledge and the knowledge change or knowledge increment that was inferred is achieved through knowledge transformers. This knowledge change or increment may lead to the learning of new design knowledge. Thus there is subjective evidence to conclude that design and learning are linked at the epistemic (knowledge) level.

In Section 8.4.5 the protocol analysis also provides evidence of the triggers of learning, explaining what activates learning and when learning could take place retrospectively, in-situ and provisionally. The rationale trigger indicates what activates learning. The temporal triggers show that there is a temporal relationship between design and learning activities; that is there is a temporal link.

Chapter 9

Evaluation of the model of learning in design

The aim of this Chapter is to evaluate and evolve, if required the proposed model of learning in design through peer critique and the results of the protocol analysis. It is envisaged that through this evaluation a more generic model of learning in design can be derived. The peer critique plays an important role since the intention of developing a model or theory of learning in design is that it must be a generic and consequently accepted by the research community and developers of design support systems. Thus the peer critique was seen as an important contribution towards the evolution of a better model of learning in design. The protocol recording of a designer at work provides another avenue by which the phenomenon of learning in design could be studied. Therefore, by comparing the results of the protocol analysis of a designer at work in relation to the proposed model, one is able to propose modifications to the model of learning in design. Hence the evaluation through the peer critique and the protocol analysis has resulted in an evolved model. Given the set of criteria used in Chapter 2 that revealed that there was no generic model of learning in design, the same set of criteria has been used to evaluate the evolved model of learning in design.

9.1 Evaluation through peer critique

The model of learning in design described in Chapter 6 was presented at the Workshop of Machine Learning in Design in 1998 for critique [126]. In the original model of the learning activity, the elements presented in Chapter 6 are (See Figure 9.1):

- Existing knowledge as Input knowledge, I_k .
- Knowledge transformers, K_t .
- Output knowledge, O_k .
- Learning goal or reason, L_g .
- Learning trigger, L_t .

After much debate and discussion, two additional elements were added (See Figure 9.2) as summarised by Duffy [267]. They are:

- Knowledge of the knowledge transformers, K_k .

- General domain knowledge which is the accumulation of output domain knowledge, ΣO_k .

The differences between the two models are discussed below.

Firstly, in the original model the input knowledge feeds into the knowledge transformer and the learning goal that activates the learning activity through the knowledge transformer. That is, the learning goal does not activate the learning trigger. Learning is considered as a goal directed activity. For learning to be effective or proactive, learning should be triggered by the learning goal. For example, when there is a failure in a design, the learning goal would be to avoid the failure in a similar design. The goal of avoiding the failure would trigger the learning process. As there is no link between the learning goal and the learning trigger, this would imply that the learning process will not be triggered. Herein lies the weakness of the original model in that there is no direct link between the learning goal and learning trigger.

This weakness is overcome in the peer evolved model. In the peer evolved model the learning goal, L_g activates the learning trigger and directs the learning process through the knowledge transformer. Hence, in the peer evolved model, the input knowledge not only feeds into the knowledge transformer but also the learning goal which activates the learning process through the learning trigger.

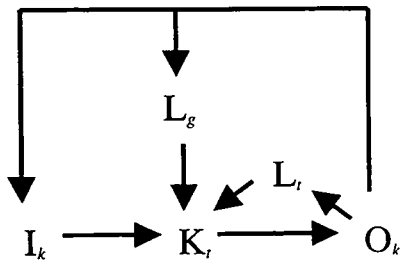


Figure 9.1: Original model of the learning activity but redrawn to highlight differences

The input knowledge is transformed into new output knowledge that may act as a reason or goal for additional learning activity. The output knowledge can then feed back into the learning activity as input knowledge in both the original and evolved models through the learning goal. But there is a significant difference in the manner the output knowledge is fed back as input knowledge in the two models. In the original model, it is the output knowledge that activates the learning trigger and re-iterates the learning

process through the knowledge transformer. In the peer evolved model, the output

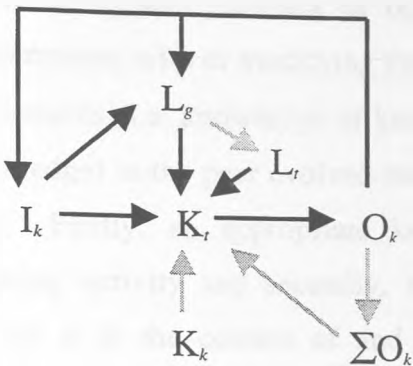


Figure 9.2: Peer evolved model of the learning activity

knowledge is fed back via the learning goal (for comparison to establish satisfaction of the goal or to modify the learning goal) before triggering the learning process. By comparing with or modifying the learning goal, the peer evolved model ensures that the knowledge learnt is consistent with or updates the goal of the learning activity.

The knowledge of the transformers is added in the evolved model to show its influences on which transformer to apply, where and when to apply a particular transformer and what would be the expected result. Through the review of MLinD systems using the formalism for the learning activity proposed in Chapter 6, it has been shown that different knowledge transformers are used to generate the new design knowledge. Hence, the addition of the element of knowledge of transformers is consistent with the finding in Chapter 6 on the influence that knowledge transformers have on the design knowledge learnt.

The original model features learning in the context of a specific design activity. In the evolved model, by including the additional element of accumulation of knowledge, ΣO_k , learning is seen as taking place in the context of and building upon what is already known (i.e. experiential knowledge) and the learning of specific knowledge adds to the repository of experience.

In summary, the peer evolved model of the learning activity has the following features that are lacking in the original model:

- The activation of the learning trigger of the learning process through the

learning goal when input knowledge is first transformed into output knowledge and in the subsequent feedback of output knowledge into the learning process by comparing with or modifying the learning goal.

- The two additional elements (i.e. knowledge of knowledge transformers and accumulation of knowledge) in the peer evolved model depict two important aspects of learning. Firstly, an appropriate knowledge transformer is involved in the learning activity and secondly, the knowledge gain in a specific design activity is in the context of and increases the knowledge already learnt.

9.2 Protocol analysis

Protocol analysis has been used to evaluate a case study of a designer at work as described in Chapter 8. The purpose of the protocol analysis was to evaluate and/or evolve the original model of learning in design. The original model of learning has been expressed in the formalisms of a design activity and a learning activity. The formalism of a design activity hypothesises what the elements of a design activity are, and how these elements interact. Similarly the formalism of learning activity hypothesises what the elements of a learning activity are, and how these elements interact. The design activity and learning activity themselves are hypothesised to interact in three manners; they interact epistemically, teleologically and temporally as explained in Chapter 7.

9.2.1 Elements of a design activity

The first hypothesis concerns the existence of elements of a design activity. The evaluation shows that the elements of a particular design activity identified are input knowledge, design goal and output knowledge. But it is by the process of inference using the hypothesis of sequential verbalisation that the input knowledge and design goal are considered as inputs to the design activity and the output knowledge is the output as suggested by the formalism of a design activity.

The ontology of design activities (see Chapter 5) was derived through a comprehensive though not exhaustive literature survey of the corpus of knowledge reported in journals, books and conference proceedings of design, cognitive science, machine learning and machine learning in design. Tables 9.1, 9.2 and 9.3 below

show a comparison between the ontology of design activities proposed in the model and those identified in the protocol analysis.

Table 9.1: Design definition activities identified in the protocol analysis

Design activity	Design activities identified in the protocol analysis
Abstracting	
Association	✓
Decomposing	✓
Defining	✓
Detailing	✓
Generating	✓
Standardising	✓
Structuring/Configuring	✓
Synthesising	✓

Table 9.2: Design evaluation activities identified in the protocol analysis

Design activity	Design activities identified in the protocol analysis
Modelling	
Analysing	✓
Testing/Experimenting	
Evaluating	✓
Decision making	✓
Determining	✓
Verifying	

Table 9.3: Design management activities identified in the protocol analysis

Design activity	Design activities identified in the protocol analysis
Constraining	✓
Exploring	✓
Identifying	✓
Information gathering	✓
Planning	✓
Prioritising	✓
Resolving	✓
Scheduling	
Selecting	✓
Searching	✓

The comparison shows that some of the design activities are not identified in the protocol recorded. The design activities not identified in the protocol are abstracting, modelling, testing/experimenting and verifying and scheduling. This may be due to the fact that the video recording was taken of a designer engaged in a particular type of design task; in this case that of general arrangement of the vessel. Unlike conceptual

design in which abstracting of ideas or concepts are used in generating designs, for general arrangement or configuration design, the systems, sub-systems and design objects have been pre-determined. So the preoccupation of the design task was how to achieve an optimal layout or arrangement that can satisfy all the design requirements. Hence this may be the reason that design activities such as abstracting have not been identified in the protocol recording of this case study.

Nevertheless, these activities have been identified in other protocol recordings. For example, Cross and Cross [255] identify the activity, planning in the Delft's protocol recording [255] among other design activities. Thus it can be said that the ontology of design activities in the proposed model provides an appropriate basis for operational definitions of design activities.

9.2.2 Elements of a learning activity

In order to identify the elements of a learning activity, the segments of the protocol in which learning occurs must be identified. The protocol analysis reveals that it is difficult to identify when learning has taken place. In most cases where learning activities have been identified, the occurrence of learning is inferred based on segments of the protocol in which there was application of past design knowledge (i.e. design reuse) in a particular design activity. These instances suggest the learning of design knowledge but it is often difficult to infer with certainty whether learning had occurred in-situ or retrospectively after the design task.

Learning is suggested to occur in design activities in which it is inferred that knowledge change has taken place. In such instances it is inferred that learning has occurred in-situ. This is illustrated by an instance in which it is possible to infer that learning has occurred in-situ. In generating various concepts of engine room layout, for one particular concept generated involving six propellers connected to 'U-drive' gearboxes, reference was made to the knowledge learnt in the first concept generated. In that concept the evaluation result is that it was not possible to accommodate six propellers involving six engines and six propellers. The evaluation result was applied to the concept in which propellers are coupled to 'U-drive' gearboxes. This clearly shows an example of in-situ learning of the result of evaluation of the first concept. This example shows that

although the designer did not state explicitly that he/she was learning during the design process, it serves as evidence that learning had taken place though implicitly (i.e. without the designer verbalising it) after evaluating the first concept. This example also serves to illustrate that input knowledge to the design activity of generating concept is the same input knowledge to the learning activity that occurred in-situ.

Inferring learning goals

In instances where it was inferred that learning could have taken place (retrospectively or in-situ) through the application of knowledge it was possible to infer learning goals from the context of the segment of the protocol. Each learning goal was inferred with respect to the design goal identified in that segment.

Knowledge transformer

The protocol analysis shows that there was no explicit way of identifying the element, knowledge transformer. In fact, the knowledge transformer is inferred by studying the knowledge change in a design activity and the knowledge transformer is then inferred from the nature of the knowledge change. Although care was taken to interpret or infer the knowledge change in each segment of protocol correctly, one cannot claim absolute objectivity in the interpretation.

Table 9.4 shows a comparison of the knowledge transformers postulated in the formalism of a learning activity and those inferred through the protocol analysis. The comparison suggests that protocol analysis provides evidence in support of the repertoire of knowledge transformers in the proposed model of learning in design, except for one pair of knowledge transformers (i.e. sorting/unsorting) that was identified in the protocol analysis.

Table 9.4: A comparison of the knowledge transformers postulated in the model of learning and those inferred in the protocol analysis.

Knowledge Transformers	Model of learning in design	Inferred in the protocol analysis
Group Rationalisation or Clustering/Decomposition	✓	✓
Similarity comparison/ Dissimilarity comparison	✓	✓
Association/Disassociation	✓	✓
Derivations/Randomisation	✓	✓
Generalisation /Specialisation	✓	✓
Abstraction /Detailing	✓	✓
Explanation /Discovery	✓	✓
Sorting/Unsorting		✓

Knowledge of knowledge transformers

The protocol analysis reveals that there may exist a mapping between design activities and knowledge transformers or vice versa. This revelation implies that the design agent should have the knowledge of knowledge transformers by which appropriate knowledge transformer(s) is applied for a given design activity. This finding corroborates with the recommendation of the peer critique that the knowledge of knowledge transformers should be considered as an additional element to augment the original formalism of learning in design.

Temporal Triggers of learning

The three types temporal learning triggers (i.e. provisional learning trigger (L_p), in-situ trigger (L_n) and retrospective trigger (L_r)) have been inferred to exist through of the protocol analysis of the case study.

Output knowledge and Accumulation of Output knowledge

From the protocol analysis of Chapter 8, it can be seen that as a result of the knowledge change, there are numerous types of output knowledge. Learning occurs when the output knowledge is recorded in the memory of an agent. For the purpose of classification, the list proposed by Grecu and Brown [40] has been used to classify the types of output knowledge learned with respect to three categories of design activities. The protocol analysis reveals that for each design activity, there is that transformation of input knowledge into specific output knowledge through the knowledge transformer. At the end of the design task, it is conceivable that there would be an accumulation of

various types of specific output knowledge. This suggests that there is a link between the nature of the design activity, the knowledge transformer and the specific output knowledge.

9.3 The evolution of the model of learning in design

The protocol analysis has been used to study the phenomenon and to evaluate the original model of learning in design. Through the evaluation in Section 9.2, the protocol analysis has revealed several aspects of the model that were not envisaged in the original formulation. Similarly, through the peer critique in Section 9.1 the original model has been augmented with two additional elements of learning. Hence from the results of the evaluation of protocol analysis and the peer critique, the findings would contribute towards formulating an evolved model of learning in design. From the study of the interactions between the design and learning activities, it has been shown that these activities are inextricably linked at the epistemic level, teleological level and temporally. This implies that the original assumption of decomposing the two cognitive activities for the purpose of simplification of the protocol analysis may not be valid. Furthermore the learning goals and design goals mutually influence one another depending on the nature of the design activities. There appears to be a mapping between the design activities and knowledge transformers. There is therefore a need to depict mapping between the knowledge of design activities and the knowledge of knowledge transformers in the model. Hence arising from the evaluations presented in Sections 9.1 and 9.2, there is a need to propose an evolved model of learning in design.

9.3.1 Interaction between design and learning activities

In Chapter 8, it has been shown that the hypotheses raised in Chapter 7 regarding the manner the design and learning activities interact can be substantiated through the protocol analysis. It has been shown that these activities interact through three links, the epistemic link, the temporal link and the teleological link.

The epistemic link

The protocol analysis reveals that knowledge change does occur in a design activity and that this knowledge change can be attributed to certain knowledge transformers should learning take place (see Chapter 8, Section 8.4.4). Where learning occurs,

this inference from the protocol analysis suggests that there is an inextricable link between the input knowledge to the design activity and the learning activity and the output knowledge of the design activity and that of the learning activity. The element that links the input knowledge and that of the output knowledge is the knowledge transformer and the resulting knowledge change is in the context of a specific design activity. That there is relationship between a design activity and learning activity is not depicted in the original model of learning in design.

The original model of learning in design was not able to predict for a given design activity, what the likely knowledge transformers for the knowledge change to occur are; that is, what the mapping between design activities and knowledge transformers is and vice versa. From the knowledge transformers inferred in the protocol analysis, it is observed that there is a mapping between the nature of design activities and the knowledge transformers. Table 9.5 presents the mapping between knowledge transformers and the three categories of design activities (i.e. design definition activities, design evaluation activities and design management activities).

From Table 9.5, it can be observed that for design definition activities, except for the two pairs of knowledge transformers Explanation/Discovery and Sorting/Unsorting, the knowledge transformers have been inferred in these activities. This is conceivable as the nature of design definition activities is such that the knowledge transformers involved will be those that promote an increase in definitional knowledge of the design rather than knowledge transformers such as Explanation/Discovery and Sorting/Unsorting. For design evaluation activities, except for the two pairs of knowledge transformers, Group rationalisation/Decomposition and Sorting/Unsorting, the knowledge transformers have been inferred in these activities. These two pairs of knowledge transformers are less likely to be responsible for the knowledge increment relating to design evaluation activities. For design management activities, the pair of knowledge transformers, Group rationalisation/Decomposition has not been inferred in these activities.

Table 9.5: Mapping between knowledge transformers and categories of design activities inferred in the protocol analysis.

Knowledge Transformers	Design definition activities	Design evaluation activities	Design management activities
Abstraction/Detailing	Defining	Determining, Evaluation	Information gathering, Selecting, Searching
Association/Disassociation	Associating, Defining, Structuring/configuring, Synthesising	Decision-making, Determining	Identifying
Derivation/Randomisation	Generating,	Analysing, Determining, Decision-making, Evaluation.	Constraining, Exploring, Identifying
Explanation/Discovery	Not inferred.	Decision-making	Constraining, Exploring, Identifying, Resolving
Generalisation/Specialisation	Defining	Determining	Constraining
Group Rationalisation/Decomposition	Decomposing	Not inferred.	Not inferred.
Similarity comparison/Dissimilarity comparison	Generating, Standardising	Analysing, Evaluation	Identifying
Sorting/Unsorting	Not inferred.	Not inferred.	Prioritising

The mapping between knowledge transformers and design activities indicates explicitly the epistemic link between the learning and design activities. The mapping also suggests that there is an underlying knowledge that determines when appropriate knowledge transformers should be applied in specific design activities; the underlying knowledge refers to the knowledge of knowledge transformers. This therefore substantiates the recommendation of the peer review to include the additional element of knowledge of knowledge transformers as an element in the evolved model.

Figure 9.3 shows the relationship between a specific design activity, D_a , and knowledge transformer, K_t , and its relationship to mapping knowledge of knowledge transformers K_k and the repertoire of design activities D_k stored in memory.

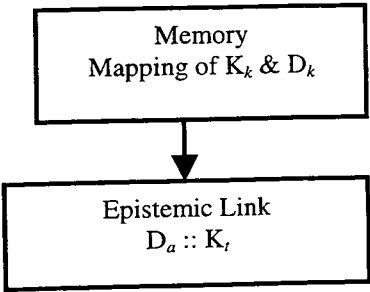


Figure 9.3: The mapping between knowledge transformers and design activities

The teleological link

The study of the interaction between the design goal, D_g , and learning goal, L_g , shows that the design goal can precede the learning goal or vice versa. The protocol study reveals that for design activities such as configuring, defining, standardising, identifying, evaluating and decision-making, the designer verbalised the learning goals before the design goals.

It is also shown in the protocol analysis in cases where there is application of knowledge learnt, it is often as a result of failure avoidance or novelty driven [32]. Where there is application of knowledge to a design situation, it implies that there is that triggering of the learning process in the first place. Hence learning occurs in these design situations because of these learning triggers. These can be considered as rationale learning triggers, L_{rw} ; that is, these triggers denote what the reason that triggers learning is. In the protocol analysis, an example of failure avoidance is the reason for triggering the learning of a similar past design situation in which a solution was given for the problem of structured induced vibration leading to crack in the structure. In another example, a novel solution applied to another problem was the reason for triggering the learning of more information from the vendor. Hence Figure 9.4 shows the rationale learning trigger, L_{rw} that activates the learning goal, which precedes the design goal. This suggests that the rational learning triggers the learning activity that precedes the design activity (i.e. that knowledge has been learnt either retrospectively or provisionally before application in the current design situation).

For activities such as associating, defining, generating, constraining and exploring, the designer verbalised the design goals before the learning goals. In Figure 9.5, the design goal, D_g precedes the learning goal, L_g (i.e. $D_g \rightarrow L_g$) in that for certain design situations, the design activities are not necessarily influenced by the learning goal. Learning may occur during the design activity so that the knowledge gained can be applied in similar design situation.

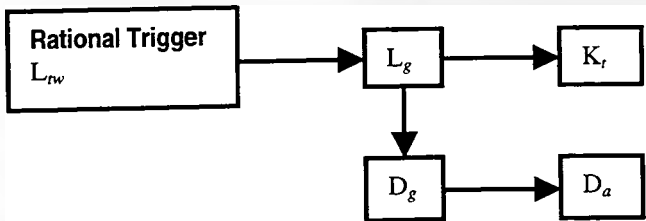


Figure 9.4: Learning goal precedes design goal.

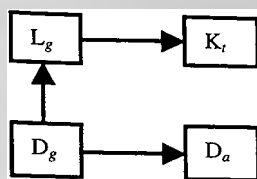


Figure 9.5: Design goal precedes learning goal.

In the original model, it is suggested that the learning goal is subservient to the design goal. The protocol analysis supports the hypothesis that the design goal and learning goal are teleologically linked but it suggests that both can mutually influence one another in activating the learning activity. Thus the protocol analysis was able to reveal an aspect of the original model that was not envisaged in the first place. In order to generalise the findings, it is necessary to conduct more protocol analyses of different design tasks (e.g. conceptual design, detail design in similar and other engineering domains). However, the findings suggest that the nature of the interaction between the design goal and the learning goal is dependent on the nature of the design activity.

The temporal link

The evaluation of the protocol analysis shows that the design activity and learning activity are temporally link (i.e. retrospectively, in-situ and provisionally) (See Figure 9.6). The protocol analysis hence supports the hypothesis that the design activity and learning activity are temporally linked as was postulated in the original model.

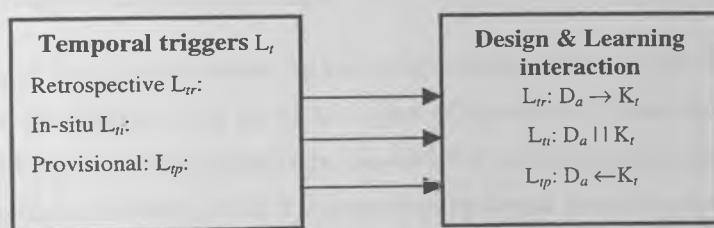


Figure 9.6: Temporal learning triggers activating the design and learning activities.

9.3.2 The evolved model of learning in design

The evolved model of learning in design (named as Model-LinD) is shown in Figure 9.7. The evolved model is derived from the evaluation of the peer critique and the

modifications arising from the protocol evaluation.

For the purpose of deriving formalisms for the design and learning activities and the analysis of protocol recording, the phenomenon of learning in design was dichotomised into two distinct cognitive activities. In the evolved model, the interactions between a design activity and learning activity are clearly shown. From the protocol analysis, it was observed, in segments of the protocol in which in-situ learning and retrospective learning were inferred, that the input knowledge into a design activity was the same input knowledge into the learning activity. Hence, there is no distinction between the input knowledge to the design activity and learning activity depicted in the evolved model.

There is a mapping between knowledge of design activities and knowledge of knowledge transformers ($D_a :: K_k$) showing the inextricable link between the design and learning activities at the epistemic level.

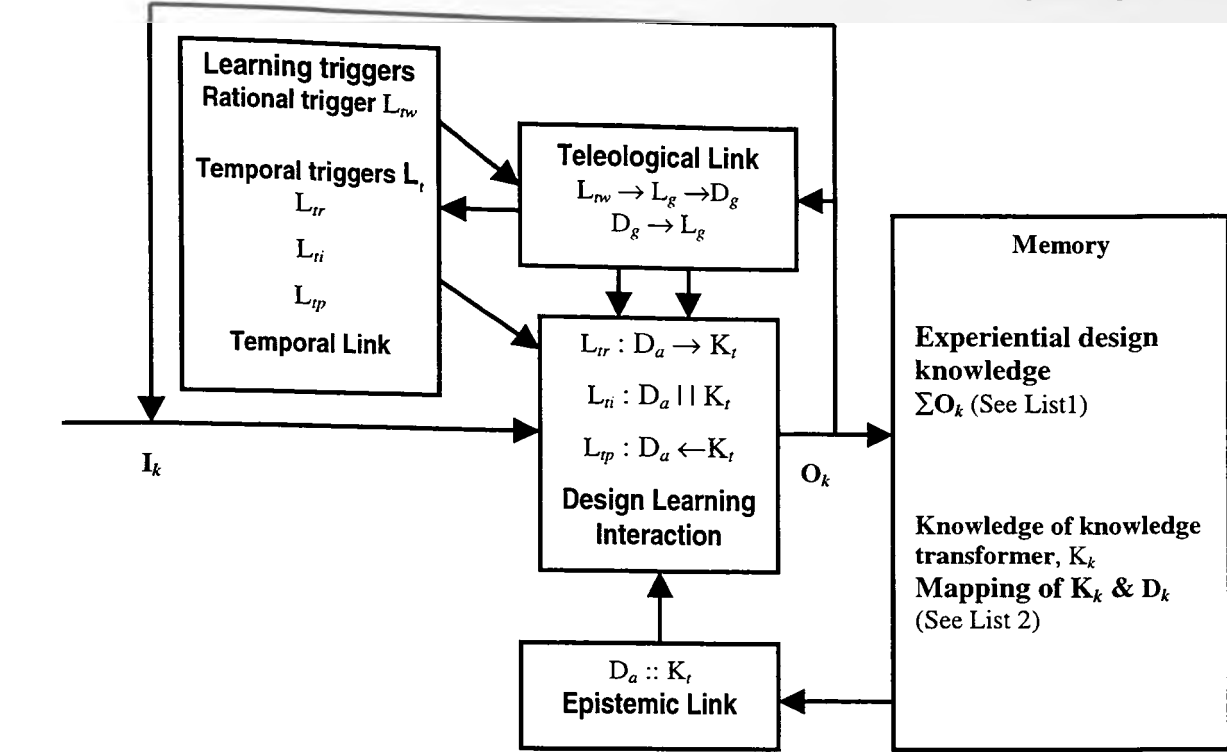
Depending on the nature of the design activity, there are two possible ways by which design and learning goals interacts with one another; this depicts the inextricable link at the teleological level. Firstly, the learning goal (which is triggered by the rationale learning trigger, L_{rw}) precedes the design goal (i.e. $L_{rw} \rightarrow L_g \rightarrow D_g$). Secondly, the design goal precedes the learning goal (i.e. $D_g \rightarrow L_g$). Depending on the learning goal, the output knowledge may act as the reason for further learning activity. Hence, the feedback of output knowledge goes as input into the learning activity depicts the iterative nature of the learning process.

For many of the design activities, the knowledge change which is a part of learning occurs in-situ or provisionally. But for certain types of experiential knowledge (e.g. trends of relationships among design parameters, knowledge of cause of failure in design) learning of the knowledge is of utility value if it is learnt in retrospect. Hence there are three types of temporal learning triggers (i.e. retrospective trigger (L_r), in-situ trigger (L_{it}) and provisional trigger (L_p)) shown in the model. For the retrospective learning trigger, the learning activity succeeds the design activity ($D_a \rightarrow K_r$). For the provisional learning trigger, the learning activity precedes the design activity ($K_p \rightarrow D_a$). For the in-situ learning trigger, the learning

activity occurs concurrently with the design activity $D_a \parallel K_i$).

The output knowledge, O_k from each learning situation is stored in the memory as experiential design knowledge that is basically an accumulation of output knowledge from the design task(s). The experiential design knowledge has been categorised into the different types of design knowledge³³ listed by Grecu and Brown [40]. Within the memory is also the knowledge of knowledge transformers, K_i and the mapping between knowledge of knowledge transformers, K_k and the repertoire of design activities, D_k .

³³ However not all the types of knowledge has been identified in the protocol analysis in Chapter 8.



List 1: Types of knowledge learnt.

analogical associations, constraints relating parameters of design, consequences of design decisions, dependencies between design parameters, design rule, design method, failures and conflicts, heuristics for failure discovery, heuristics for constraints, preferences, preconditions and postconditions for rules, successful design, successful design process, rationale for decision

List 2: Mapping between knowledge transformers and design activities (taken from Table 9.5).

Knowledge Transformers	Design definition activities	Design evaluation activities	Design management activities
Abstraction/Detailing	Defining	Determining, Evaluation	Information gathering, Selecting, Searching
Association/Disassociation	Associating, Defining, Structuring/configuring, Synthesising	Decision-making, Determining	Identifying
Derivation/Randomisation	Generating,	Analysing, Determining, Decision-making, Evaluation.	Constraining, Exploring, Identifying
Explanation/Discovery	Not inferred.	Decision-making	Constraining, Exploring, Identifying, Resolving
Generalisation/Specialisation	Defining	Determining	Constraining
Group Rationalisation/Decomposition	Decomposing	Not inferred.	Not inferred.
Similarity comparison/Dissimilarity comparision	Generating, Standardising	Analysing, Evaluation	Identifying
Sorting/Unsorting	Not inferred.	Not inferred.	Prioritising

Figure 9.7: An evolved model of learning in design, Model-LinD

9.3.3 Backus-Naur Form of the model of learning in design

The elements of the model of learning in design, its respective terms and the relations of their interaction can be expressed in the Backus-Naur Form [83]. Backus Naur Form (BNF) is a standard language for describing the syntax of typical programming language. There are several major components of a BNF description. At the lowest level, a terminal character is a character from the alphabet for the language being described. Its use in a BNF statement means that a language statement constructed according to that BNF rule must have that character embedded properly in it. At the next level is a non-terminal that represents some sub-language being described. A production rule or rewrite rule is a basic BNF statement that indicates one way of generating part of the valid sub-language for some non-terminal [83]. Its format is as follows:

$$\langle \text{non-terminal} \rangle := \langle \text{BNF-expression} \rangle$$

There may be multiple production rules (IF condition THEN conclusion) with the same left-hand side non-terminal. The right-hand side of a production rule is a BNF expression indicating the appropriate subset of valid strings for this terminal. A BNF program consists of a set of production rules. The order of rules within the program is immaterial; all ordering produce /define the same language.

The BNF can thus be used to implement a computer model of learning in design and serves as a basis to implement a design support system. Figure 9.8 shows Backus-Naur Form representation of the model of learning in design (See Appendix 3 for the meanings of the symbols used (e.g. $:=$, $\langle \rangle$ etc.).

$\langle \text{learning in design} \rangle := \langle \text{design activity} \rangle \mid \langle \text{learning activity} \rangle \mid \langle \text{interactions between design \& learning activities} \rangle \mid \{ \langle \text{learning in design} \rangle \}$

$\langle \text{design activity} \rangle := \langle \text{input knowledge} \rangle \mid \langle \text{design goal} \rangle \mid \langle \text{output knowledge} \rangle \mid \{ \langle \text{design activity} \rangle \}$

$\langle \text{design activity} \rangle := \langle \text{design definition activity} \rangle \mid \langle \text{design evaluation activity} \rangle \mid \langle \text{design management activity} \rangle$

$\langle \text{design definition activity} \rangle := \text{abstracting} \mid \text{associating} \mid \text{decomposing} \mid \text{defining} \mid \text{detailing} \mid \text{generating} \mid \text{standardising} \mid \text{structuring} \mid \text{synthesising}$

$\langle \text{design evaluation activity} \rangle := \text{modelling} \mid \text{analysing} \mid \text{testing} \mid \text{evaluating} \mid \text{decision making} \mid \text{verifying}$

$\langle \text{design management} \rangle := \text{constraining} \mid \text{exploring} \mid \text{identifying} \mid \text{information gathering} \mid \text{planning} \mid \text{prioritising} \mid \text{resolving} \mid \text{scheduling} \mid \text{selecting} \mid \text{searching}$

$\langle \text{input knowledge} \rangle := \langle \text{design drawing} \rangle \mid \langle \text{design method} \rangle \mid \langle \text{design rules} \rangle \mid \langle \text{design requirements} \rangle \mid \langle \text{output knowledge} \rangle$

$\langle \text{learning activity} \rangle := \langle \text{input knowledge} \rangle \mid \langle \text{learning goal} \rangle \mid \langle \text{learning trigger} \rangle \mid \langle \text{temporal learning trigger} \rangle \mid \langle \text{knowledge transformer} \rangle \mid \langle \text{output knowledge} \rangle \mid \langle \text{mapping} \rangle \mid \{ \langle \text{learning activity} \rangle \}$

$\langle \text{mapping} \rangle := \langle \text{knowledge of design activity} \rangle \mid \langle \text{knowledge of knowledge transformer} \rangle$

$\langle \text{knowledge of knowledge transformer} \rangle := \langle \text{knowledge transformer} \rangle \mid \{ \langle \text{knowledge transformer} \rangle \}$

$\langle \text{knowledge transformer} \rangle := \langle \text{knowledge transformer pair} \rangle$

$\langle \text{knowledge transformer pair} \rangle := \text{group rationalisation} \mid \text{decomposition; similarity comparison} \mid \text{dissimilarity comparison; association} \mid \text{disassociation; derivation} \mid \text{randomisation; generalisation} \mid \text{specialisation; abstraction} \mid \text{detailing; explanation} \mid \text{discovery; sorting} \mid \text{unsorting.}$

$\langle \text{output knowledge} \rangle := \text{analogical associations} \mid \text{constraints relating parameters of design} \mid \text{consequences of design decisions} \mid \text{dependencies between design parameters} \mid \text{design rule} \mid \text{design method} \mid \text{failures and conflicts} \mid \text{heuristics for failure discovery} \mid \text{heuristics for constraints} \mid \text{preferences} \mid \text{preconditions and postconditions for rules} \mid \text{successful design} \mid \text{successful design process} \mid \text{rationale for decision}$

$\langle \text{learning triggers} \rangle := \langle \text{rationale learning trigger} \rangle \mid \langle \text{temporal learning trigger} \rangle$

$\langle \text{rationale learning trigger} \rangle := \text{expedience driven} \mid \text{novelty driven} \mid \text{failure avoidance} \mid \text{conflict avoidance}$

$\langle \text{temporal learning trigger} \rangle := \text{provisional trigger} \mid \text{retrospective trigger} \mid \text{in-situ trigger}$

$\langle \text{interactions between design \& learning activities} \rangle := \langle \text{epistemic link} \rangle \mid \langle \text{teleological link} \rangle \mid \langle \text{temporal link} \rangle$

$\langle \text{epistemic link} \rangle := \langle \text{input design knowledge} = \text{input learning knowledge} \rangle \text{ and } \langle \text{output design knowledge} = \text{output learning knowledge} \rangle$

$\langle \text{teleological link} \rangle := \text{rationale learning trigger} \rightarrow \langle \text{learning goal} \rangle \rightarrow \langle \text{design goal} \rangle \mid \langle \text{design goal} \rangle \rightarrow \langle \text{learning goal} \rangle$

$\langle \text{temporal link} \rangle := \text{if provisional trigger then } \langle \text{learning activity} \rangle \rightarrow \langle \text{design activity} \rangle \mid \text{if retrospective trigger then } \langle \text{design activity} \rangle \rightarrow \langle \text{learning trigger} \rangle \mid \text{if in-situ trigger then } \langle \text{design activity} \rangle \parallel \langle \text{learning activity} \rangle$

Figure 9.8: Backus-Naur Form representation of the model of learning in design.

9.4 Evaluation of the evolved model of learning in design

The set of criteria used in Chapter 2 has been used to show the need of a model or theory of learning in design. This predefined set of criteria is now used to evaluate the evolved model of learning in design. To recapitulate, they are:

- Explanative capability.
- Predictive capability.
- Operationalisation.
- Comprehensiveness.
- Applicability.

9.4.1 Explanative capability

The evolved model of learning in design should exhibit explanative capability in explaining the phenomenon of learning in design.

Grecu and Brown [40] have proposed a set of dimensions as a guide for comparing research work of machine learning in design. Reich [39] has proposed a more elaborate set of dimensions for describing learning in the context of engineering practice. Reich's dimensions consist of a set of fundamental questions that one would expect a model or theory of learning in design would provide answers to. Hence it is appropriate to use Reich's dimensions (RD) or Grecu and Brown's dimensions (GBD) as criteria for evaluation of the proposed model of learning in design. But since RD is a more comprehensive set of criteria, it shall be used to evaluate the model of learning in design.

Table 9.6 shows the evaluation of the evolved model using RD as criteria. For each of the questions raised in RD, the model is evaluated in terms of what elements in its model are able to provide the answer and in what manner the elements interact to provide the answer.

Who is learning?

In Chapter 7, it is posited that it is an agent (i.e. either human or system) that is involved in the learning activity. The agent is governed by the principle of rationality. The principle of rationality dictates that the agent selects actions that will attain the agent's goal(s). In the context of design, the agent's goal is defined by design goal(s). However, the agent is not explicitly represented in the model. It is

represented implicitly in that it is governed by the principle of rationality (See Chapter 5).

Why does the learner want to learn?

In the evolved model of learning in design, the object of learning is captured by the learning goal, L_g , and its interaction with the design goal, D_g . It is the learning goal that triggers the learning process and what knowledge is to be acquired. Thus, the reason for learning (the why) is reflected in L_g , which is linked to, and gives further elaboration as to “why”, the D_g . The interaction between learning and design goal is accounted for in the model by the teleological link.

When does the learner learn?

This is accounted by the *temporal* link in the model of learning in design which shows that learning can occur retrospectively, in-situ and provisionally and these are denoted by the elements, L_{tp} , L_{in} , L_{ir} respectively.

What is the learner doing?

The evolved model of learning in design positions the learning activity in the context of a specific design activity D_a . That is, learning can occur in the context of any design activity and what is learnt and how learning occurs are dependent on the context of a design activity. A repertoire of design activities in which knowledge change can occur was postulated in this thesis. It also postulates three hypotheses regarding the interaction between the design activity and the learning activity. These activities are hypothesised and shown to be linked epistemically, teleologically and temporally (See Chapter 8).

What is learned?

In the evolved model of learning in design, the element O_k is used to denote the outcome of the learning process. The model also postulates that what is learned is dependent on the nature of the design and learning activities, the learning goal and the input knowledge. For a given design task, the accumulation of O_k results in $\sum O_k$ which represents the experiential knowledge of the designer.

Table 9.6: Evaluation of the evolved model of learning in design using RD

Reich's dimensions (RD)	Elements in the of Model of learning in design
Who is learning?	Not explicitly represented in the model although an agent (human or system) is implied.
Why does the learner want to learn?	Learning goal L_g , Rationale learning trigger L_r , Link to D_g .
When does the learner learn?	Retrospective trigger (L_{rr}), in-situ trigger (L_{in}) and provisional trigger (L_{ip}).
What is the learner doing?	In the context of particular design activity D_a
What is learned?	The output knowledge O_k denotes the outcome of the learning process. For a given design activity D_a , it is dependent on input knowledge I_k the goal of learning, L_g , and the design goal, D_g . The accumulation of O_k results in ΣO_k .
How does the learner learn?	Through the knowledge transformer (K_t) and knowledge of knowledge transformers, K_k
What are the consequences of learning?	Depends on the learning goal L_g and activated by rationale learning trigger, L_{rr} .
What resources are needed to carry out the learning activity?	Not addressed by the model.

How does the learner learn?

The evolved model of learning in design postulates that learning can take place through knowledge transformers, K_t . These knowledge transformers describe different processes by which knowledge change can take place and hence the appropriate knowledge transformer is influenced by the knowledge of knowledge transformers and the nature of design activity; that is, there is a mapping between the design activity and knowledge transformer.

What are the consequences of learning?

The consequences of learning are governed by the reason for which learning is initiated (i.e. the learning goal). The reasons for learning are represented by rationale learning trigger. These have been postulated to include reasons such as novelty driven, excellence driven and failure avoidance [32] and so the consequences of learning may lead to novelty in design, excellence and improvement in design [32].

What resources are needed to carry out the learning activity?

The resources needed for learning is dependent on what is to be learnt, the manner in which learning can take place through knowledge transformers and the goal of learning. This requires a cost-benefit analysis of different algorithms or machine learning methods for a given knowledge. However, this aspect is not addressed by

the model and it will become pertinent in implementing the elements of the model as a design support systems.

The evaluation of the evolved model or Model-LinD against RD's criteria shows that the model has explanative capability in explaining the phenomenon of learning in design of a single designer. Through the interactions of its elements, Model-LinD has the explanative capability in describing the phenomenon in terms of what knowledge can be learned and how that knowledge learning is learnt, when learning occurs and the motivation and consequence of the learning. Models of learning in design reviewed in Chapter 2 are able to explain limited aspects of the phenomenon. For example, exploration-based model, design reuse model and Design/Learning Loop are able to explain learning in retrospect and in-situ but they do not feature a comprehensive definitions of knowledge transformers to explain how learning occurs. Model-LinD features three links (epistemic, teleological and temporal) to explain the inextricable link between learning and design. However, the inextricable link that characterises the phenomenon is not apparent in models of learning in design reviewed). Hence, it is reasonable to conclude that Model-LinD displays explanative capability to explain the phenomenon of learning in design. As discussed in Chapter 2, since explanative capability is considered as the most important characteristic of a model or theory, the objective of developing a model of learning in design has been achieved in Model-LinD.

9.4.2 Predictive capability

The evolved model of learning in design has limited predictive capability. The evaluation of the protocol analysis suggests that for a given type of design activity there is a possible set of knowledge transformers. This observation however is based on one protocol analysis. Hence the mapping of design activities and corresponding knowledge transformers cannot be considered as conclusive.

The evolved model is able to predict that the learning and design goals are teleologically linked in two possible ways: the design goal precedes learning goal or vice versa. The interaction between design and learning goals is not featured in other models of learning in design.

While it is able to predict that the design activities and learning activities are temporally linked, it is not able to predict for a given design activity, its temporal relationship with the learning activity. However, through the protocol analysis, it is clear that for certain design activities, learning can only occur in-situ. For example, for the design activity of determining, it is generally observed that the related learning activity occurs in-situ. Another example is the activity of evaluating a design concept where spatial and geometric considerations are involved, the outcome of the evaluation has to be learnt in-situ.

While Model-LinD has some predictive capability, it lacks an inference structure that will enable it to predict the outcome of the phenomenon in a conclusive and comprehensive manner.

9.4.3 Operationalisation

This requires that all terms and concepts used in the new model or theory be subjected to operational definition. The Backus Naur Form (BNF) representation of the model of learning in design can be used for the operationalisation of the model; that is to implement a computational model of learning in design. While the BNF representation has been the basis of successful implementation of many programming languages, by implementing a computational model of learning in design using the BNF representation given in Figure 9.8 will ensure that it is not implemented on an ad hoc basis, but rather on a formal basis.

9.4.4 Comprehensiveness

Design and learning by themselves are complex cognitive activities. In developing the model of learning in design it is intended to explain the phenomenon of learning situated in a design context. By itself, the model must address the complexity that arises when the activity of designing and the activity of learning interact. Based on the evaluation using RD as a set of criteria (in Section 9.1.2), it has been shown that the evolved model, Model-LinD is able to provide explanations to fundamental questions that one would expect a model of learning in design to do. Thus except for the question that involved the use of resources to perform the learning activity, it is reasonable to state that the evolved model is indeed comprehensive in explaining the phenomenon of learning in design of a single designer. This conclusion is based on the evolved model of learning

in design being derived from the results of the evaluation of the peer critique and protocol analysis. The elements of the design activity and learning activity from which the evolved model of learning has been derived were identified/inferred from the protocol recording. These elements were shown to interact in such a manner that they provide answers to the fundamental questions based on RD as a set of criteria. Hence, using the notation of Miller in describing comprehensiveness of a model or theory, it can be considered as a middle-range model [69]. That is, the model developed here is able to explain the phenomenon of learning in design of a single designer in a context of a design practice, but not in the context of a team of designers. Models of learning in design that have been reviewed are only able to explain some aspects of the phenomenon.

9.4.5 Applicability

The applicability of the model of learning in design can only be gauged by how effective it is in assisting designers by implementing the concepts and hypotheses of the model in design support systems. The application is generic if the model of learning in design can be implemented in domains such as architecture. Hence this is an aspect of the evaluation that

This is prob what I'll end up doing...

9.5 Summary

The evaluation of the proposed model of learning in design through the peer critique and the results of the protocol analysis has led to an evolution of the model. Evaluated against the set of criteria used in Chapter 2, the evolved model, Model-LinD is shown to display explanative capability but limited predictive capability, has operational definitions as represented in BNF for implementation as a computational model of learning in design and is comprehensive in explaining the phenomenon of learning in design of a single designer but yet to prove its applicability in general. Hence, it is reasonable to conclude that Model-LinD is by far the most comprehensive model of learning in design on which a computational model of learning in design can be developed.

Chapter 10

Discussion and future work

Central to the research work of this thesis has been the identification of and the development of a model of learning in design. To pursue the research work, the research methodology identified was based on the research perspective of pragmatic scientism in which the underlying research methodology is the scientific method (see Section 3.2.1). This led to the development of a model of learning in design and subsequently its evaluation using peer review (by a community of researchers), protocol analysis and against a predefined set of criteria. The evaluation resulted in the evolution of the proposed model. While the model of learning in design was able to explain the phenomenon as recorded in the protocol of a designer at work, the model is found to be inadequate in many aspects as revealed by the evaluation in Chapter 9. This inadequacy may be due to the manner in which the research task has been performed (i.e. research methodology) or due to the nature of the model itself or the inadequacy of the methods of evaluation. Therefore, this Chapter has three objectives. Firstly, it is pertinent to discuss the strength and weaknesses of the research approach taken. Secondly, the results of this research work in terms of the model developed are summarised and the inadequacies highlighted. Thirdly, future work is proposed in the light of the results and inadequacies presented so that the research task of developing a generic model or theory of learning in design can be achieved.

10.1 Research approach

The scientific method used requires the formulation of hypotheses regarding the proposed model or theory and these hypotheses are to be validated or evaluated by experiment. Hence a model of learning in design has been formulated and subsequently evaluated by peer review and analysing a protocol recording of a designer at work.

While it is true that the scientific method can be applied in all disciplines, it has to be adapted to fit the specific needs of the research problem [268]. The study of science and the discovery of scientific theories and laws are achieved through a systematic, well-focussed and well-established scientific method. The formulation of a theory/model of

learning in design, however, has to deal with the complexity of the application of scientific knowledge to design, and the complexity of the human learning processes in the design process either working individually or working in a team. Hence, the nature of developing a model or theory of learning in design is such that it has to do with the cognitive processes of a designer (i.e. the designing process and the learning process). The cognitive processes being neural activities of the human brain are not directly observable. Where there were problems associated with observing and recording data that characterised the model, the history of science has shown that these problems were overcome when the appropriate equipment or devices or even measuring instruments were constructed that were sensitive enough to record the data [120]. However, the problems associated with observing the billions of neural activities in the human brain during the learning process have remained intractable.

Although scientific theories and models, models of learning in design are products of human cognitive processes, there are two distinct differences between the scientific theories and models and models of learning in design. The first is science (for which scientific theories are developed) has to do with the study of nature around us and learning in design has to do with study of the learning behaviour of a human designer. Hence the human becomes the object of the study rather than nature. The second distinct difference lies in the nature of the results of their respective experiments (i.e. experiments on validating scientific theories and case studies in evaluating models of learning in design). The scientific theories and laws can be validated by repeated experiments and the results within the limits of experimental errors are always consistent. On the other hand, models of learning in design are more difficult to evaluate or validate because the object of the investigation is a human cognitive process. Human designers become the subject of the scientific paradigm of hypothesis testing through evaluation or experimentation. But no two designers are alike in the manner they perform a design task and hence their cognitive processes during designing may be different. Although it is possible to validate scientific theories by repeated experiments, the same approach of investigation cannot be used for the model of learning in design. Hence rather than trying to validate the proposed model of learning in design, the method of investigation is that of evaluating the model by peer review and evaluation through a case study. But the manner in which the evaluation has been performed is not without any weaknesses. The weaknesses are:

- Inadequacy of peer review.
- Weakness of case study.
- Weakness of scope of evaluation.

Inadequacy of peer review

While peer critique provides collective wisdom in evaluating the model of learning in design, the changes made to the model are at best hypothetical in nature. Each of the critique is based on the critic's understanding of the phenomenon and hence it could be subjective and maybe non-conclusive in its contribution. Nevertheless, its value lies in the fact that it is a model that is acceptable by the group of researchers at the time of the critique. This can be considered in Kuhn's paradigm, as in the state of normal science until some anomalies are raised on the ground of empirical evidence or rationalisation. Hence, by itself peer critique cannot be considered as a de facto method of evaluation.

Weakness of the protocol analysis

The protocol analysis of the video recording suffers from the lack of objectivity in the manner in which it was analysed. As it was analysed solely by the author of this thesis, it is difficult to claim there is absolute objectivity. The presence of subjectivity was minimised by re-analysing the protocol recording after a period of a few weeks. But this is also recognised as inadequate in trying to ensure that the analysis is as objective as possible. Therefore the manner in which the protocol recording has been analysed can be considered as a weakness of the research approach. This would not be so had it been that the analysis been done under different circumstances where another researcher can be involved in the investigation.

Another weakness of the protocol analysis is that of identifying the incidence of learning within the segments of the protocol. That is, the designer did not state explicitly when he has learned new knowledge. The incidences of learning were inferred through identifying segments where there was application of design knowledge and a noticed knowledge change. One approach that may overcome this weakness is to interview the designer retrospectively regarding those instances where there was application of learning in order to establish how and when the design knowledge was learned in the

first place.

Weakness of scope of evaluation

There is a need to explore further how the model of learning can be modified through protocol recordings of different designers performing different nature of design tasks (e.g. conceptual design, configuration design and detail design etc.) since these protocol recordings represent the reality of the phenomenon of learning in design. To be a truly comprehensive model of the phenomenon of learning in design, there is a need to consider the social aspect of design where designers learn from one another through their interactions in the course of designing. To study the phenomenon of learning in design with designers working as a team there is a need to record and analyse the appropriate protocols.

10.2 Results and solutions

The purpose of the research work presented in this thesis has been to develop a model of learning in design that can explain its phenomenon and hence serve as a basis to develop design support systems. This has resulted in a descriptive model of learning in design that features two distinct cognitive activities (i.e. a design activity and a learning activity for the purpose of simplification) that interact in three distinct manners in order to explain the phenomenon of learning in design adequately. The two cognitive activities have been described by two formalisms.

The formalism for the design activity is described in terms of its elements, namely input knowledge, design goal, knowledge of design activities and knowledge output as a result of the design activity. The formalism has been used as a generic description for a repertoire of design activities that are necessary for the completion of a design task. The formalism has been used to analyse the knowledge change for a given design activity.

The second formalism describes the learning activity in terms of the elements of input knowledge, learning goal, learning trigger, knowledge transformer and output knowledge. The formalism was further augmented by two more elements through a peer evaluation. The two elements are knowledge of knowledge transformers and the accumulation of output knowledge as distinct from specific output knowledge that may arise from each learning activity. The formalism also depicts the manner in which these elements would interact. For example it is the learning goal that activates the learning

activity through the learning trigger. The different manners in which knowledge change can occur is represented by a repertoire of knowledge transformers.

The model of learning in design also features three hypotheses that describe the inextricable link between the learning and design activities. These links have been referred to as epistemic, teleological and temporal links. The hypothesis on the epistemic link states that design and learning as cognitive activities can be described at the knowledge level. The teleological link prescribes that design and learning being purposeful cognitive activities interact through the design and learning goals. The hypothesis on the temporal link prescribes that learning can occur in retrospect of a design activity, in-situ (or during a design activity), or in anticipation of a design activity.

The two formalisms and the three hypotheses have been evaluated by analysing a protocol recording of a designer at work. The protocol analysis shows that the elements of the design activity can be identified/inferred in segments of the recording depicting these design activities. While it was more difficult to identify when learning had taken place, in segments where there was application of knowledge it was possible to identify/infer elements of the learning activities (i.e. learning goal, knowledge transformer and learning triggers). The protocol analysis also shows that there is some evidence in support of the three hypotheses that describe the inextricable link between the design activity and learning activity (i.e. these interact through the epistemic, teleological and temporal links). The results of its evaluation as described in Chapter 9 led to further modifications resulting in a new model of learning in design. The new model of learning in design has been expressed in BNF notation. This would facilitate implementation as a computer model of learning in design. It cannot be considered as a generic model as there is a need to subject the model to more extensive and less subjective evaluation. However, it is reasonable to conclude that the research work in this thesis has introduced a novel descriptive model of learning in design and can act as an initial basis upon which to further advance the state of knowledge in the field.

10.3 Future work

The future work proposed is seen in the context of a research framework promoted by

Duffy and Andreasen [269] and Duffy and O'Donnell [125]. The research framework was based on the hypothesis that models should be built based on the “reality” of design and continually evolved to develop tools to support design. The evolution involves the building of phenomena models, information models and computational models. Phenomena models are primarily based on upon observations and analyses of the “reality” of design and the use of tools employed and hence reflect design practice. Where appropriate, these models are then developed in more detail as information models and similarly as computational models and tools [269]. Figure 10.1 shows the relationship of the future work to the phenomena model of learning in design developed in this thesis.

The research work of this thesis has resulted in a descriptive model of learning in design of a single designer. There is therefore a need to understand the phenomenon of learning of a designer in a team of designers at work and derive a generic model or theory from which information model(s) of learning in design and/or computational model(s) can be developed. In developing a computational model of learning in design there is a need to understand the issues and differences involved between human learning in design and that of implementing computational models or systems. This is because the analysis of protocol recording in Chapter 8 shows that the cognitive activities of the designer, while working on the design task, involved both verbal (symbolic) and visual (pictorial) processing of information and knowledge. The review of systems of machine learning in design in Chapter 6 shows that machine learning algorithms used are largely symbolic or sub-symbolic in nature. Hence the future work proposed here is directed at improving/modifying the model of learning in design (given in Chapter 9) so that a more comprehensive/complete generic model or theory of learning in design is derived so that effective computational model or systems can be built to support designers. In the light of these considerations, the following areas are suggested as avenues of future work:

- Enhancing the explanative and predictive capability of the model of learning in design.
- Extending the comprehensiveness of the model of learning in design by explaining not just the learning behaviour of a single designer but of a team of designers.
- Implementing an effective computational model of learning in design by

studying the issues and differences between human and computer models of learning in design so that a human/computer symbiosis can be achieved in a design support system.

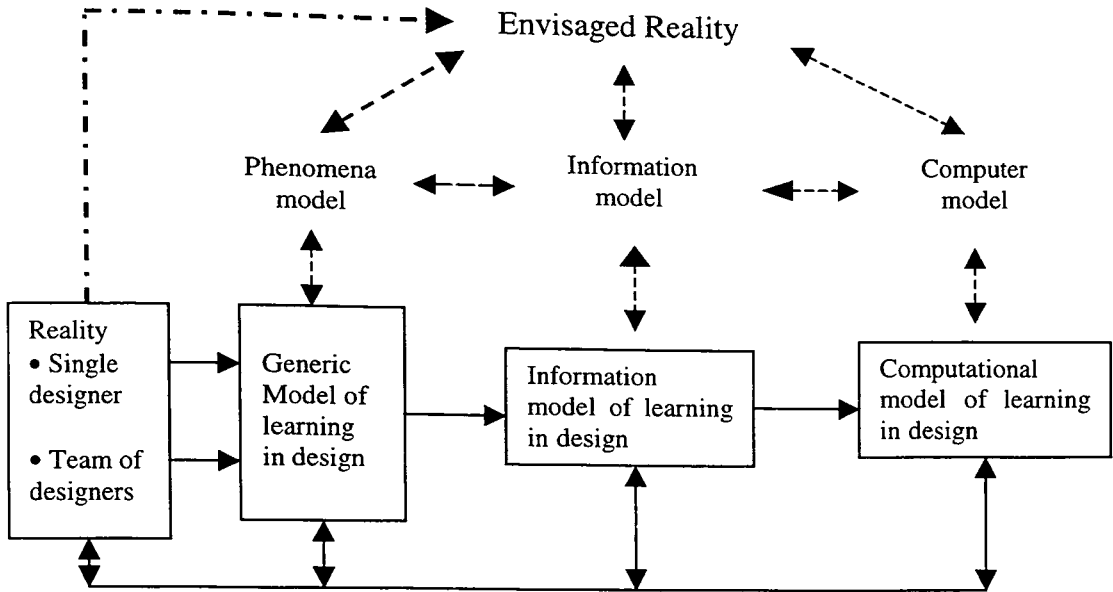


Figure 10.1: Research framework for future work (adapted from [125]).

10.3.1 Enhancing explanative and predictive capability

The model of learning in design was based on the results of peer evaluation and that of a single protocol analysis. In order to derive a more generic model of learning in design or a theory of learning in design, there is a need to study the phenomenon of learning in design more extensively through protocol analysis. By studying and analyzing protocol recordings of designers at work in different types of design tasks (e.g. conceptual design, detail design) and in different design domains (i.e. architectural, mechanical, electronics etc) it is envisaged that a more generic model of learning in design can be evolved. Through such extensive study it is envisaged that a generic model of learning in design can be formulated, evaluated and substantiated leading to a theory of learning in design. In the process, the study could substantiate or modify the formalism for design activity, the formalism for learning activity, and add/modify the hypotheses regarding the interactions between the design and learning activities.

It is envisaged that extensive study of the phenomenon of learning in design of a single designer may lead to a generic mapping of design activities to knowledge transformers. The protocol analysis of the single designer in Chapter 8 has revealed patterns of sequence

of design activities. This pattern of sequence of design activities may influence the manner in which procedural design knowledge may be learned. It is envisaged that the extensive study that is based on the reality of design will lead to a prescriptive model of the design process and the manner in which knowledge of design rationale may be learned.

10.3.2 Model of learning in team design

Design is a social activity as much as it is a cognitive activity. This is because design often requires the knowledge of more than a single discipline [1] or from a team of designers with expertise knowledge of different disciplines. Because design is a social activity that is knowledge intensive, there is sharing of knowledge among team members [270]. Hence it is reasonable to extend the model of learning in design of a single designer to that of a team. By extending the current work of developing a model or theory of learning in design to that of a design team, the model would then be in a position to explain or predict the phenomenon of learning in design in a more comprehensive manner.

In formulating a model of learning in design for a single designer the research work has revealed the challenges and difficulties encountered in such a formulation. Formulating such a model for a team of designers would present even greater complexities. These complexities are therefore avenues for further research work in order to achieve the research goal of developing a more comprehensive model or theory of learning in design. Some of the complexities faced in developing a model or theory in design are discussed here.

In the context of team design, Siemiemiuch and Sinclair [13] are involved:

- Knowledge that is of direct relevance to the development of products.
- Knowledge that is required to manage the development of the products.
- Knowledge that is required for cooperation in the application of knowledge.

Hence, in addition to learning design knowledge within a single designer's context, the model of learning in design would have to explain or predict the different types of design knowledge that can be learnt through interaction among members of a design team. This may involve recording protocols of a team of designers at work, analysing categories of knowledge learnt by each designer and knowledge learnt through interaction by the team members.

Perry and Sanderson [271] highlighted and established the importance of two key dimensions of joint design work: social and organisational interactions, and the ongoing nature of the knowledge representation and transformation work that takes place through the use of design artefacts (e.g. plans, models, prototypes and visualisation). Their case studies provide empirical observations that design work can no longer be adequately conceptualised in terms of ‘individual intelligence’, nor as a linear process with a set of design stages. But rather, it should be viewed as a situation in which joint, coordinated learning and work practices evolve, and in which artefacts help to mediate and organise communication. The observation by Perry and Sanderson adds further complication to the manner the representation and evolution of the design artefact would affect both learning of design knowledge by an individual designer and corporately as a team.

Learning knowledge of strategies for conflict detection and resolution for a team design may introduce complexities that are not envisaged for that of a single designer. Klein and Lu [272] developed a model of conflict resolution for cooperative design through protocol recording of a team of architects designing a house. They observe that knowledge acquisition in cooperative design presents special challenges and requires additional techniques compared to traditional knowledge acquisition.

10.3.3 Issues and differences between human learning and machine learning

In developing a computational model based on the phenomena model of human designer’s learning in design there are many issues and differences that have to be addressed in order that design support systems can be effective and efficient in supporting designers at work. However two issues are raised for discussion here. They are:

- The verbal and visual processing of information/knowledge during design.
- The loss of fidelity over time of human memory and hence of learning knowledge.

Verbal and visual processing of knowledge

It is fact that humans are adept at processing (i.e. remembering (learning) and recognising) complex arrays of three-dimensional objects and pictures (i.e. two-

dimensional representation of a three-dimensional object). That humans can recognise the conceptual identity of pictures and their corresponding real objects implies that picture processing is relatively simple [273].

It is also known that an average person coordinates visual and verbal processing and almost all activities involve some interaction between the two systems [273]. That there is interaction between the visual and verbal system is evident in the protocol recording of the designer at work.

Design involves the manipulation of objects or pictorial representation of design in terms of drawings and sketches. Learning in design involves the processing of both visual and verbal (symbolic) information. For example, learning of the result of evaluating whether a concept is feasible involves the visual processing of the composition/interactions between design objects before a verbal pronouncement of the result is made and recorded in memory; human designers appear to perform visual and verbal processing almost effortlessly. These observations cannot be said of the manner in which machine learning systems learn patterns or pictorial representations or objects. For machine learning systems to learn these patterns or objects requires implementation of complex algorithms. Hence a clear understanding of the manner in which the visual and verbal information interacts during learning from a human learning perspective and from a machine learning perspective must be sought in order that a human/machine symbiosis can be achieved.

Loss of fidelity of human memory over time

While humans have flexibility in learning all types of knowledge under different circumstances, humans suffer from forgetfulness and the inability to record knowledge learnt verbatim. Humans suffer from what is known as a decay hypothesis which states that memory simply weakens as a function of time and is therefore harder to retrieve [37]. Machine learning does not suffer from the human phenomenon of forgetfulness but has great fidelity in recording the knowledge learnt. But it lacks the ability to learn all kinds of knowledge unless its repertoire of knowledge transformers and machine learning methods are programmed to learn that particular kind of knowledge. While human learning may

occur consciously as well as subconsciously, machine learning has to be triggered. These differences between human learning and machine learning however need not present stumbling blocks in building design support systems. But rather they should serve as avenues in which weaknesses of human learning can be overcome by the strength of the machine learning.

Some differences between the two behaviours of learning (i.e. human learning and machine learning) are discussed here. It is not meant to be exhaustive, but to highlight the need for a thorough understanding of these differences so as to lead to a synergy between the two types of learning and which results in the building of effective and efficient design support systems.

Chapter 11

Conclusion

The theme of the research work presented in this thesis has been the modelling of learning in design. The main contributions derived and conclusions reached through the research methodology of pragmatic scientism is the focus of this Chapter. This thesis contributes towards the knowledge of modelling learning in design and has resulted in a novel and comprehensive model of learning in design.

Design practice characterised by design activity and learning activity

That the phenomenon of learning in design exists has been observed by many researchers in design and machine learning in design. That design practice is characterised by design and learning activities has been the basis on which many design support systems or Machine Learning in Design (MLinD) systems have been built (See Chapter 2 and Chapter 6). These systems have been developed on an ad hoc basis, without a comprehensive formalism to support their development. Without a comprehensive formalism (i.e. a model or theory), these systems have been developed based on specific aspects of learning in design, to describe or prescribe what learning in design is. But, until the publication of this thesis, there was no model or theory of learning in design.

The phenomenon of learning in design

This thesis describes learning in design as a phenomenon that occurs in the context of design (i.e. situated learning). It was hypothesised that the cognitive activity of designing is inextricably linked to the cognitive activity of learning. Every design situation is seen as an opportunity for learning design knowledge. In order to study the phenomenon, every design action or design activity provides an opportunity to study the phenomenon so as to answer the epistemological questions:

- What is learned?
- How does learning occur?
- What is the goal of learning?
- When does learning occur?

Critical review of existing theories, models and systems

To determine if there were indeed an existing model of learning in design, that could explain or predict the phenomenon of learning in design, existing models or theories of design and/or of learning that represented a cross section of the design domains were identified and reviewed. The review shows that there are descriptive models and theories of design that describe the nature of the phenomenon of design. There are also prescriptive models and theories that prescribe how design should be performed. Most of the models or theories reviewed describe or prescribe the design aspect per se without describing the learning aspect. In some models (e.g. exploration-based model, design reuse model and design/learning loop) and MLinD systems in which some aspect of learning is featured, the review showed that these models and systems were inadequate in describing the phenomenon of learning in design.

Criteria for evaluating theories and models

The models or theories were evaluated using a set of criteria that characterised a model or theory in general and a model or theory of learning in design in particular. The pre-defined set of criteria used in the evaluation of existing models and theories were identified as explanative capability, predictive capability, comprehensibility, operationalisation and applicability. That is, a theory or model must exhibit one or several of these characteristics before it can be considered to be a model or theory.

Non-existent model of learning in design

The review of theories, models and systems points to one of the contributions made by the research work. That is, there is indeed no theory or model that can explain the phenomenon of learning in design. Hence the need identified was to develop a model of learning in design and as a consequent, an approach to model learning in design.

Research methodology used to derive the model of learning in design

Learning in design is not strictly a scientific phenomenon, but a phenomenon of the cognitive processes of designers who apply scientific laws, principles and theories to design. Therefore, although the principal research methodology adopted was that of

scientism, the method of evaluation was not through experimentation as is the norm for traditional scientific work, but through a case study of a designer at work; this research methodology was referred to as pragmatic scientism. Hence the research methodology for modelling learning in design consisted of two complementary approaches. The first approach was to derive formalisms for modelling the design and learning activities and to present three hypotheses describing the interactions between these activities; the formalisms and hypotheses constitute a proposed model of learning in design. The second approach was to evaluate the proposed model through a protocol analysis.

Requirements for a model or theory of learning in design

The requirements for a model of learning in design were identified using the same pre-defined set of criteria to evaluate the theories and models of design and/or learning. It was determined that the model developed must answer the epistemological questions raised above in explaining the phenomenon of learning in design.

A model of learning in design

That the phenomenon of learning in design can be characterised by two cognitive activities served as basis to develop the model of learning in design (Model-LinD). From the review of published literature, two formalisms were derived for these cognitive activities. The Model-LinD was derived by studying the manner in which these cognitive activities interact and the nature of these interactions so that it explains the phenomenon of learning in design.

Formalism for design activity

From the review of the published literature on design and case studies on design (e.g. analyses of protocol recordings), the numerous design activities that designers perform were identified and classified into design definition activities, design evaluation activities and design management activities. The review also served as the basis for the formulation of a formalism for the design activity. The formalism for a given design activity consists of the elements of input knowledge, design goal and output knowledge. The formalism was used to analyse knowledge change that may occur for a repertoire of design activities; knowledge change in a design activity

is considered a part to the learning process. The formalism for the design activity and the knowledge change identified for the design activities then served as background knowledge for analysing the protocol analysis.

Formalism for learning activity

The review of the published literature on learning, machine learning and machine learning in design served as the basis for the formulation of a formalism for the learning activity. This formalism consists of the elements of input knowledge, learning goal, learning trigger, knowledge transformer and output knowledge. By the formalism, it is shown that these elements are related to one another in a logical and structured manner that contributes to the derivation of the model of learning in design. The elements of input knowledge and the output knowledge are to describe state of the knowledge transformation that occurs in a learning process. The element of learning goal represents learning as a purposeful activity directed by the learning goal. The review also identified a set of knowledge transformers that are used to generate, and modify existing knowledge into new knowledge. The knowledge transformers explain how learning can take place. Four types of learning triggers were identified; three triggers are temporal in nature and they are retrospective, in-situ and provisional triggers and the fourth is the rationale trigger. The temporal triggers and the rationale trigger are to explain when or what triggers learning.

The formalism for the learning activity that was derived served as the basis for analysing the phenomenon of learning in design in a protocol recording.

Interaction between design activity and learning activity

The cognitive activities of design and learning were presented to interact in three manners; three hypotheses were raised to describe these interactions. Firstly, they interact at the knowledge or epistemic level. Secondly, they interact teleologically through the design goal and learning goal. Thirdly, they interact temporally through retrospective, in-situ and provisional triggers. Hence the inextricable link between design and learning is described by the epistemic link, the teleological link and the temporal link respectively. These links were identified in the protocol analysis and hence substantiated these three hypotheses.

Protocol recording and analysis for the phenomenon of learning in design

A protocol recording of a designer at work provided another avenue by which the phenomenon of learning in design was studied and analysed. Through the protocol analysis, the list of design activities identified through the published literature on design and cased studies on design was substantiated. The elements presented in the formalisms for the design and learning activities were also identified and substantiated. The original repertoire of knowledge transformers was augmented with an additional knowledge transformer (i.e. sorting/unsorting) to account for knowledge change of the design activity of prioritising.

The protocol segments were analysed in terms of the interactions between design and learning activities in the related elements in order to substantiate three hypotheses raised on the interaction between these activities. The protocol segments of various design activities was analysed into the elements of input knowledge and output knowledge from which knowledge change and the corresponding knowledge transformer that may lead to the learning of new knowledge were inferred. It was observed that there is a relationship between design activities and knowledge transformers for the three categories of design activities (i.e. design definition activities, design evaluation activities and design management activities). Therefore, a mapping between the design activities and the set of knowledge transformers was derived. It was on the basis of the mapping that it was suggested that design activities and learning activities are inextricably linked. Hence, there is subjective evidence to conclude that design and learning are linked at the epistemic (knowledge) level.

The identification of the elements of the design goal and learning and the analysis of the interaction between them showed that learning and designing are linked at the teleological level. The evaluation of their interaction suggests that these goals interact in two manners which are:

- The learning goal precedes the design goal.

- The design goal precedes the learning goal.

The protocol analysis also provided evidence of the triggers of learning to explain that learning took place retrospectively, in-situ and provisionally. This shows that there is a temporal relationship between design and learning activities; that is there is a temporal link.

The findings of the protocol analysis were used to evaluate and evolve the original Model-LinD.

Evaluation of a model of learning in design

The original Model-LinD was evaluated and evolved through peer review and the findings of the protocol analysis. Through the peer review, two elements (i.e. knowledge of knowledge transformers and accumulation of output knowledge) were added to augment the formalism for the learning activity. Secondly, the protocol analysis was used to substantiate the elements of the design activity and the learning activity. The three hypotheses relating to the interactions between design and learning activities were also substantiated showing that the epistemic link, teleological link and temporal link exist between the activities. As a result of the evaluation through the peer review and the protocol analysis, the evolved Model-LinD was derived. The evolved Model-LinD features learning in design as transformation of the input knowledge into output knowledge through the interactions between the design and learning activities that are determined by teleological link, epistemic link and the temporal link. The epistemic link explains the mapping between the specific design activity and corresponding knowledge transformer which is derived from the mapping between the knowledge of knowledge transformers and repertoire of design knowledge stored in memory. The teleological link explains the interaction between design and learning goals. The temporal link explains that the phenomenon of learning in design can occur retrospectively, in-situ and provisionally. The knowledge output is stored in memory that accumulates experiential design knowledge.

The evaluation of evolved Model-LinD against the pre-defined set of theory criteria showed that the model has comprehensive capability of explaining and limited predictive capability in describing the phenomenon of learning in design.

Strengths and weaknesses of the model of learning in design

The original Model-LinD was formulated based on formalisms and hypotheses (as described in Chapters 5, 6 and 7) to satisfy the need identified of the non-existence of a model of learning in design. The evaluation of the original Model-LinD through the results of the peer review and protocol analysis resulted in the evolution of the model. The evolved Model-LinD features formalisms for both the design and learning activities that has epistemic, teleological and temporal links. The evolved Model-LinD has been evaluated against a pre-defined set of criteria. The evaluation showed that the evolved Model-LinD is able to explain comprehensively the phenomenon of learning in design that occurs in design practice. Hence the main contribution of this thesis has been the development of a model of learning in design.

The weaknesses of the development of the model lie in the subjectivity that may exist in the protocol analysis and the lack of extensive evaluation using more protocol recordings and analyses. While the evaluation of the evolved model showed that it has explanative capability and limited predictive capability, it nevertheless lacks an inference structure for it to have comprehensive predictive capability. This weakness is due to the lack of extensive evaluation by which it is envisaged that a more generic model or theory of learning can be derived.

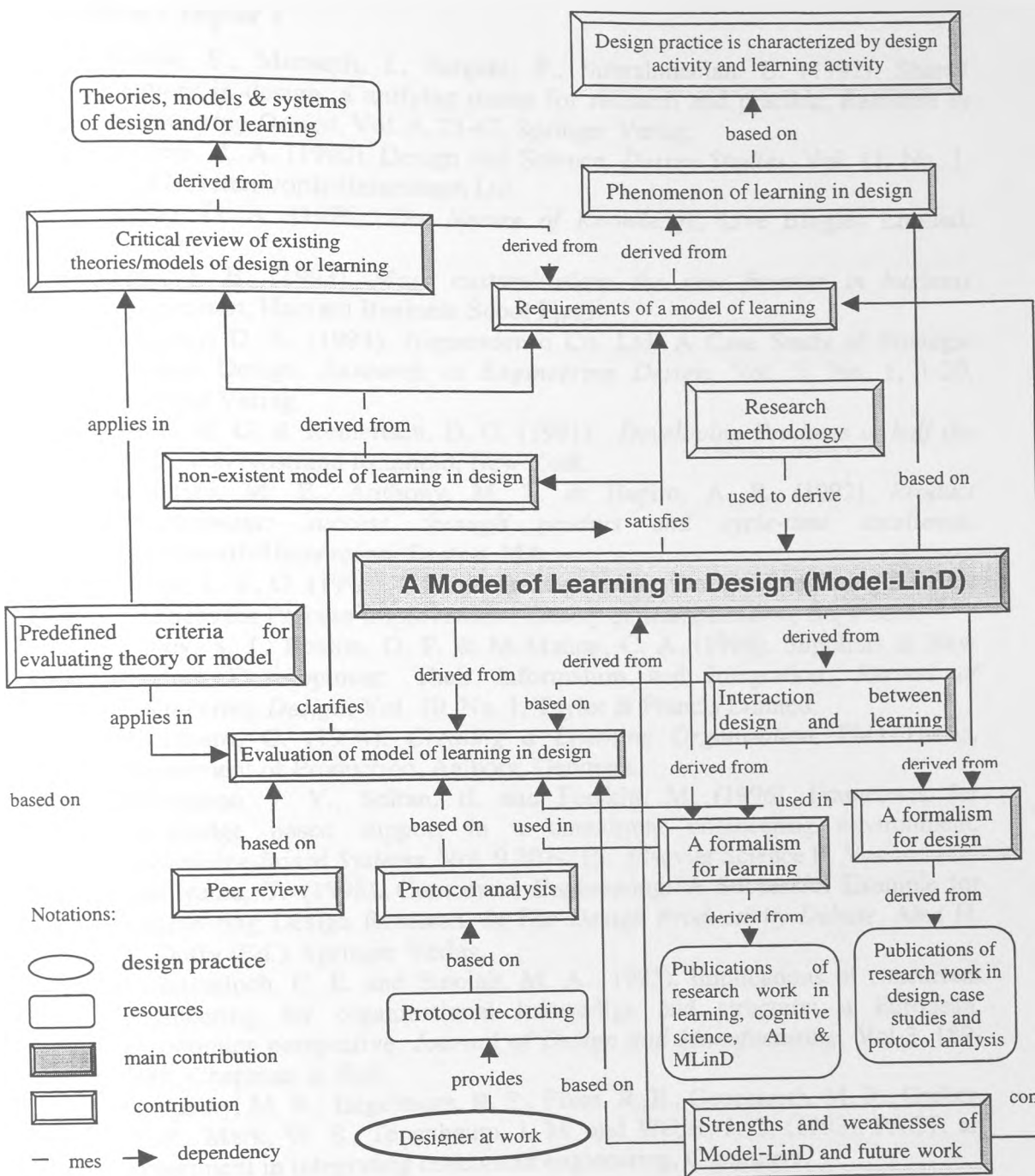
As the phenomenon of learning in design occurs both in the context of a single designer as well as a team of designers, the model lacks the comprehensibility in explaining the phenomenon in the context of a team of designers. Designers can learn through interactions among team members. Since the derivation and evolution of Model-LinD was based on the protocol recording of a single designer, it is capable of explaining the phenomenon of learning in design only in the context of a single designer.

Future work

The future work as recommended in Chapter 10 was based on enhancing the predictive capability of the evolved model and its comprehensiveness in explaining the phenomenon in the context of a team of designers. It is also based on the need to understand the difference between human learning and computational learning in design so that an

effective symbiosis between man and machine can be achieved. It is also based on an aspect of learning that was identified in the protocol analysis, that of visual learning that is not a feature in the model. It is envisaged that the future work would lead to a generic model or theory of learning in design on which future effective and efficient design support systems or Intelligent Design Assistants (IDA) can be developed to support designers.

Figure 11.1 provides a summary of the work presented in this thesis by highlighting the contributions and their dependencies.



References Chapter 1

- [1] Konda, S., Monarch, I., Sargant, P., Subrahmanian, E. (1992). Shared memory in design: a unifying theme for research and practice, *Research in Engineering Design*, Vol. 4, 23-42, Springer Verlag.
- [2] Willem, R. A. (1990). Design and Science, *Design Studies*, Vol. 11, No. 1. 43-47, Butterworth-Heinemann Ltd.
- [3] Kemp, D. A. (1976). *The Nature of Knowledge*, Live Bingley Limited, London.
- [4] Pine, J. B. (1993). *Mass customisation: the new frontier in business competition*, Harvard Business School Press.
- [5] Whitney, D. E. (1993). Nippondenso Co. Ltd: A Case Study of Strategic Product Design, *Research in Engineering Design*, Vol. 5, No. 1, 1-20, Springer Verlag.
- [6] Smith, P. G. & Reinertsen, D. G. (1991). *Developing Products in half the Time*, Van Nostrand Reinhold, New York.
- [7] McGrath, M. E., Anthony, M. T. & Hapiro, A. R. (1992). *Product Development: success through product and cycle-time excellence*, Butterworth-Heinemann, Boston, MA.
- [8] Miller, L. C. G. (1993). Concurrent Engineering Design: Integrating the Best Practices for Process Improvement, *Society of Manufacturing Engineers*.
- [9] Culley, S. J., Boston, O. P. & McMahon, C. A. (1999). Suppliers in New Product Development: Their Information and Integration, *Journal of Engineering Design*, Vol. 10, No. 1, Taylor & Francis Limited.
- [10] Neergaard, C. (1994). *Creating a Learning Organisation*, PhD Thesis, Department of Production, Aalborg, Denmark.
- [11] Harrington, J. V., Soltan, H. and Forskitt, M. (1996). Framework for knowledge based support in a concurrent engineering environment. *Knowledge-Based Systems*. Vol. 9 207-215. Elsevier Science B. V.
- [12] Tomiyama, T. (1998). Concurrent Engineering: A Successful Example for Engineering Design Research. In *The Design Productivity Debate*, Alex H. B. Duffy (Ed.). Springer Verlag.
- [13] Siemiemiuch, C. E. and Sinclair, M. A. (1993). Implications of concurrent engineering for organizational knowledge and structure: a European, ergonomics perspective. *Journal of Design and Manufacturing*. Vol 3. 189-200. Chapman & Hall.
- [14] Cutkosky, M. R., Englemore, R. S., Fikes, R. E., Genesereth, M. R., Gruber, T. R., Mark, W. S., Tenenbaum, J. M. and Weber, J. C. (1993). PACT: an experiment in integrating concurrent engineering, *Computers*.
- [15] Levitt, R. E., Jin, Y. Dym, C. L. (1991). Knowledge-based support for management of concurrent, multidisciplinary design, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol 5, 77-95, Academic Press Limited.
- [16] Douglas Jr, R. E. and Brown, D. C. (1993). Concurrent accumulation of knowledge: a view of concurrent engineering. In *Concurrent Engineering: Contemporary Issues and Modern Design Tools*, Hamid R Parsaei and William G. Sullivan (Eds.). 402-412. Chapman & Hall.

- [17] Busby, J. S. (1999). The Problem with Design Reuse: An investigation into Outcomes and Antecedents, *Journal of Engineering Design*, Vol. 10, No. 3, Taylor & Francis Limited.
- [18] Busby, J. S. (1998). Effective Practices in Design Transfer, *Research In Engineering Design*, 178-188, Springer-Verlag London Limited.
- [19] Duffy, A. H. B. & Ferns, A. F. (1999). An Analysis of Design Reuse Benefits, In *Proceedings of International Conference on Engineering Design*, ICED 99 Munich.
- [20] Mital, A., Anand, S. and Kopardekar, P. (1993). An ergonomic evaluation of three popular CAD packages, *Journal of Design and Manufacturing*, Vol. 3, 201-213, Chapman & Hall.
- [21] Thomas P. J. (Ed.). (1996). *CSCW requirements and evaluation*. Springer Verlag, New York.
- [22] Chaffey, D. (1998). *Group Ware, Workflow and Intranets: Reengineering the Enterprise with Colloborative Support*. Digital Press.
- [23] Stevenson, S. and Dooley, K. J. (1994). The use of Best Design Practices: An Analysis of US Navy Contractors, *Research in Engineering Design*, Vol. 6, 14-24, Springer Verlag London Limited.
- [24] Duffy, S. M. & Duffy, A. H. B., (1996). Sharing the learning activity using intelligent CAD, *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, 1996, Vol. 10, 83-100, Cambridge University Press.
- [25] MacCallum, K. J., Does Intelligent CAD exist? *Artificial Intelligence In Engineering*, 1990.
- [26] Duffy, A. H. B. (1986). *Computer modelling of early stage numerical ship design knowledge and expertise*. Ph. D Thesis, Department of Shipbuilding and Marine Technology, University of Strathclyde, Glasgow, Scotland.
- [27] Watson, I. And Perera, S. (1997). Case-based design: A review and analysis of building design applications, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol 11, 59-87, Cambridge University Press.
- [28] Maher, M. L., Brown, D. C., & Duffy, A. H. B. (1994). Special Issue: Machine learning in Design, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 8, No. 2.
- [29] Duffy, A. H. B., Brown, D. C., & Maher, M. L. (1996). Special Issue: Machine learning in Design, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 10, No. 2.
- [30] Duffy, A. H. B., Brown, D. C., & Goel, A. K. (1998). Special Issue: Machine learning in Design, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 12, No. 2.
- [31] Gruber, T., Baudin, C., Boose, J. and Weber, J. (1991). Design Rationale Capture as Knowledge Acquisition: Tradeoffs in the Design of Interactive Tools. In *Proceedings of the Eighth International Workshop on Machine Learning*, Evanston, IL, June 1991.
- [32] Sim, S. K. & Duffy, A. H. B. (1998). A foundation for machine learning in design, *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, Vol. 12, 193-209.
- [33] Zuboff, S. (1988). *In the Age of Smart Machine: the future of work and power*. Basic Books, New York.

- [34] Rabin, M. J., (ed), *Goals and priorities for research in engineering design*, An NSF Workshop, ASME.
- [35] Persidis, A, & Duffy, A. (1991). Learning in engineering design. In *Intelligent CAD III* (Yoshikawa, H., Arbab, F. and Tomiyama, I., Eds), 251-272. Elsevier Science Publishers B. V. (North Holland), Amsterdam.
- [36] Argyris, C. & Schon, D. A. (1978). *Organisational Learning: A Theory of Action Perspective*, Addison Weesley Publishing Company.
- [37] Anderson, J. R. (1995). *Learning and Memory*, John Wiley and Sons.
- [38] Duffy, A. H. B. (1997). The "What" and "How" of Learning in Design, *IEEE Expert*, 71-76.
- [39] Reich, Y. (1998). Learning in design: From characterising dimensions to working systems, *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, Vol. 12, 161-172, Cambridge University Press.
- [40] Grecu, D. L. & Brown, D. C. (1998). Dimensions of machine learning in design, *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, Vol. 12, 117-121, Cambridge University Press.
- [41] Hergenhann, B. R. (1988). *An Introduction to Theories of Learning*, Prentice Hall.
- [42] Marton, F & Booth, S. (1997). *Learning and Awareness*, Lawrence Erlbaum Associates, Inc., Publishers, Mahwah, New Jersey.
- [43] Kimble, G. A. (1961). *Higard and Marquis' conditioning and learning* (2nd edition). Englewood Cliffs, NJ: Prentice Hall.
- [44] Klein, S. B. (1987). *Learning: Principles and Applications*, McGraw Hill Book Company.
- [45] Simon, H. A. (1983). Why should machines learn? In *Machine Learning: An artificial Intelligence Approach*, Michalski et. Al. (eds.), Morgan Kaufmann.
- [46] Mitchell, T. M. (1997). *Machine Learning*, The Mc-Graw Hill Companies, Inc.
- [47] Atkinson, R. C., & Shiffrin, R. M. (1968). Human memory: A proposed system and its control processes. In K.W. Spence & J. T. Spence (Eds.), *The psychology of learning and motivation* (Vol 2.). New York: Academic Press.
- [48] Gillund, G., & Shrifffin, R. M. (1984). A retrieval model for both recognition and recall. *Psychological Review*, 91 1-67.
- [49] Adelman, G. (Ed.). *Readings from Encyclopedia of Neuroscience: Learning and Memory*. (1987). Birkhauser Boston Inc.
- [50] Cooper, L. N. (Ed.). (1995). *How we learn; How we remember: Toward an understanding of brain and neural systems*, World Scientific Publishing Co. Pte. Ltd. Singapore.
- [51] Goker, M. H., (1997). The effects of experience during design problem solving, *Design Studies*. 405-426. Elsevier Science Ltd.
- [52] Raichie, M. (1994). *Visualising the Mind*, Scientific American. Vol. 270, No. 4. 36-42.
- [53] Schank, R. (1982). *Dynamic Memory: A theory of reminders and learning in computers and people*. Cambridge University Press, Cambridge, U. K.
- [54] Kerr, S. M. (1992). Human Memory Research Developments and its Influence on Intelligent Computer-Aided Design. CADC/R/92-04, CAD Centre, University of Strathclyde, Glasgow, UK.

References Chapter 2

- [55] Smithers, T. (1996). On Knowledge level Theory of Design Process, in J. S. Gero and F. Sudweeks, *Artificial Intelligence in Design '96*, Kluwer Academic Publishers.
- [56] Braithwaite, R. B. (1970). Models in Empirical Science, in Baruch A. Brody (ed.) *Readings in the Philosophy of Science*, 268-293, Prentice Hall.
- [57] Solso, R. L. (1991). *Cognitive Psychology*, Ally and Bacon.
- [58] Dixon, J. R. (1987). On Research methodology towards a scientific theory of engineering in design, *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, Vol. 1, 145-157, Cambridge University Press.
- [59] Smithers, T. and Troxell, W. (1990). Design is intelligent behaviour, but what's the formalism. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, Vol. 4, 89-98.
- [60] Mason, J., Mathias, P. and Wescott, J. H.(Eds). (1986), *Predictability in Science and Society*, Proceedings of a Joint symposium of the Royal Society and the British Academy, Univeristy Press, Cambridge.
- [61] Ziman, J. (1984). *An introduction of science studies: the philosophical and social aspects of science and technology*. Cambridge University Press.
- [62] Korn, J. (1996). Domain-independent Design Theory, *Journal of Engineering Design*, Vol. 7, No. 3. Journals Oxford Limited.
- [63] Michalski, R. S. (1993). Inferential theory of learning as a conceptual basis for multistrategy learning, *Machine Learning*, Vol. 11 3-43.
- [64] Shanahan, M. Prediction is deduction but explanation is adduction. In Commonsense Reasoning.
- [65] Chang, R. (1998). Chemistry (6th edition). McGraw Hill.
- [66] Lighthill, J. (1986). The recently recognised failure of predictability in Newtonian dynamics. In Mason, J., Mathias, P. and Wescott, J. H.(eds), *Predictability in Science and Society*, Proceedings of a Joint symposium of the Royal Society and the British Academy, Univeristy Press, Cambridge.
- [67] Bridgman, P. W. (1938). Operational analysis. *Philosophical Science*.
- [68] Hempel, C. G. (1965). *Aspects of Scientific Explanation*, The Free Press, 1965.
- [69] Miller, G. G. (1995). *Living Systems*, Univeristy of Colorado Press, 1995.
- [70] Dasgupta, S. (1991). *Design Theory and Computer Science*, Cambridge University Press.
- [71] Yoshikawa, H. (1981). General Design Theory and a CAD System, *Man-machine Communication in CAD/CAM*, Sata, T., Warman, E. A., (eds.), North-Holland, Amsterdam.
- [72] Tomiyama, T. and Yoshikawa, H. (1985). Extended General Design Theory. In *Design Theory for CAD*, Proceedings of the IFIP Working group 5.2 Working conference, Tokyo, Yoshikawa, H. and Warman, E. A. (eds.), North-Holland, Amsterdam, 95-130, 1987.
- [73] Tomiyama, T., Kiryama, T., Takeda, H., Xue, D., Yoshikawa, H. (1989). Metamodel: A Key to Intelligent CAD Systems, *Research in Engineering Design*, Vol. 1, 19-34, Springer Verlag.
- [74] Takeda, H., Hamada, S., Tomiyama, T., & Yoshikawa, H. (1990). A cognitive approach to the analysis of design process. In *Design Theory and*

- Methodology – DTM '90*, DE-Vol. 27, (Rinderle, J. R., Ed.) 153-160, ASME, New York.
- [75] Takeda, H., Tomiyama, T., and Yoshikawa, H. (1990). Logical formalisation of design processes for intelligent CAD systems. In *Intelligent CAD II*, (Yoshikawa, H., & Holden, T., Eds), 325-336. North Holland, Amsterdam.
 - [76] Takeda, H., Tomiyama, T., and Yoshikawa, H. (1992). A logical and computable framework for reasoning in design. In *Design Theory and Methodology – DTM '92*, DE-Vol. 42, (Taylor, D. L. & Stauffer, L. A., Eds.) 167-174, ASME, New York.
 - [77] Suh, N. P. (1990), *The Principle of Design*, New York: Oxford Univeristy Press.
 - [78] Suh, N. P. (1998). Axiomatic Design Theory for Systems, *Research in Engineering Design*, Vol. 10, 189-209, Springer-Verlag London Limited.
 - [79] Babic, B. (1999). Axiomatic design of flexible manufacturing systems, *International Journal of Production Research*, Vol. 37, 1159-1173.
 - [80] Hubka, V. & Eder, W. E. (1987), A scientific approach to engineering design, *Design Studies*, Vol. 8 No. 3, 123-137.
 - [81] Hubka, V. & Eder, W. E. (1988). *Theory of Technical Systems*, Springer Verlag.
 - [82] Ehrlenspiel, K. (1994). Theory of Technical Systems, *Journal of Engineering Design*, Vol 5, 117-128.
 - [83] Kogge, P. (1991). *The Architecure of Symbolic Computers*, McGraw Hill, Inc.
 - [84] Arciszewski, T. & Michalski, R. S. (1994). Inferential Design Theory: a conceptual outline. In, *Artificial Intelligence in Design '94*, (J. S. Gero and F. Sudweeks, Eds.), 295-308, Kluwer Academic Publishers.
 - [85] Karni, R. & Arciszewski, T. (1997). A tool for the conceptual design of production and operations systems, *Research in Engineering Design*, Vol. 9, 146-167.
 - [86] Tomiyama, T. (1994). From general design theory to knowledge intensive engineering, *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, Vol. 8, 319-333.
 - [87] Smithers, T. (1998). Towards a Knowledge level Theory of Design Process. In J. S. Gero and F. Sudweeks, *Artificial Intelligence in Design '98*, Kluwer Academic Publishers.
 - [88] Gebala, D.A. & Suh, N. P. (1992). An Application of Axiomatic Design, *Research in Engineering Design*, Vol. 3, 149-162.
 - [89] Albano, L. D. & Suh, N. P. (1992). Axiomatic Approach to Structural Design, *Research in Engineering Design*, Vol. 4, 171-183.
 - [90] Suh, N. P. (1984), Development of the science base for manufacturng field through axiomatic approach, *Robotics and Computer Integrated Manufacturing*, Vol. 1, 399-415.
 - [91] Suh, N. P. Bell, A. C. & Gossard, D. (1978a). On an axiomatic approach to manufacturing systems, *Journal of Engineering for Industry*, Transactions of ASME, Vol. 100, 127-130.
 - [92] Suh, N. P., Kim, S. H., Bell, A. C., Wilson, D. R., Cook, N. H., Lapidot, N. (1978b). Optimisation of manufacturing systems through axiomatics, *Annals of CIRP*, Vol. 27.

- [93] Suh, N. P. (1982). Orthonormal processing of metals, Part 1: concept and theory, *Journal of Engineering for Industry, Transactions of ASME*, Vol. 104, 327-331.
- [94] Sekimoto, S. & Ukai, M. (1994). A study of creative design based on the axiomatic design theory, In *Design Theory and Methodology – DTM '94*, DE-Vol. 68, (Eds.), 71-77, ASME, New York.
- [95] Agüero, U., *A Theory of Plausibility for Computer Architecture Designs*, Ph. D. Thesis, University of Southwestern Louisiana, Lafayette, LA.
- [96] Finger, S. & Dixon, J. R. (1989). A review of research in mechanical engineering design. Part1: descriptive, prescriptive, and computer-based models of design processes, *Research in Engineering Design*, Vol. 1, 51-67.
- [97] Birmingham, R., Cleland, G., Driver, R. and Maffin, D. (1997). *Understanding Engineering Design – context, theory and practice*. Prentice Hall.
- [98] Maffin, D. (1998). Engineering Design Models: context, theory and practice. *Journal of Engineering Design*, Vol. 9, No. 4. Carfax Publishing Limited.
- [99] Pahl, G. and Beitz, W. (1st edition 1984, 2nd edition 1996). *Engineering Design*, The Design Council, Springer-Verlag, London.
- [100] Andreasen, M. M. (1991). Design Methodology, *Journal of Engineering Design*, Vol. 2, No. 4, 321-335.
- [101] Pugh, S. (1991) *Total Design: Integrated methods for successful product engineering*, Addison Wesley Publishing Company, Wokingham, England, UK.
- [102] Ullman, D. G. (1992). *The Mechanical Design Process*, McGraw Hill Inc., 1992.
- [103] Ulrich, K.T. and Eppinger, S. D. (1995). *Product Design and Development*, McGraw Hill International.
- [104] Smithers, T., Conkie, A., Doheny, J., Logan, B., Millington, K. & Tang, M. X. (1990). Design as intelligent behaviour: an AI in design research programme, *Artificial Intelligence in Engineering*, Vol. 5, No. 2, 78-109, Computational Mechanics Publication.
- [105] Smithers, T., Tang, M. X., Tones, N., Buck, P., Clarke, B., Lloyd, G., Poulter, K., Floyd, C., Hodgkin, E. (1992). Development of a knowledge-based design support system, *Knowledge-Based Systems*, Vol. 5, No. 1, 31-40, Butterworth-Heinemann Ltd.
- [106] Hayes-Roth, B. (1985). A blackboard architecture for control, *Artificial Intelligence*, Vol. 26, 251-321.
- [107] Duffy, S. M., Duffy, A. H. B. & MacCallum, K. J. (1995). A design reuse model, *International Conference of Engineering Design (ICED)*, Vol. 2, 490-495.
- [108] Duffy, A. H. B., Persidis, A., & MacCallum, K. J. (1995a). NODES: A numerical and object based modelling system for conceptual engineering design, In *Knowledge Based Systems*, (Edmonds, E. A., Ed.). Elsevier Science Publishers Ltd., Amsterdam.
- [109] Kerr, S. M. (1993). *Customised viewpoint support for the utilisation of experiential knowledge in design*, Ph.D. thesis, CAD Centre, Department of Design, Manufacturing and Engineering Management, University of Strathclyde, Glasgow, U.K.

- [110] Duffy, A. H. B. & Duffy, S. M. (1996). Learning for design reuse, *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, 1996, Vol. 10, 139-142.
- [111] Manfaat, D., Duffy, A. H. B. & Lee, B. S. (1998). SPIDA: Abstracting and generating layout design case, *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, 1998, Vol. 12, 141-159.
- [112] Duffy, A. H. B. (1997). The "What" and "How" of learning in design. In *IEEE Expert, Intelligent Systems and Their Applications*, 71-76. IEEE Computer Society, New York.

References Chapter 3

- [113] Smith, N. C. & Dainty, P. (1991). *The Management Research Handbook*, Routledge, London.
- [114] Reich, Y. (1994). Layered models of research methodologies, *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, Vol. 12, 193-209.
- [115] Guba, E. G. (1990). The alternative paradigm dialog. In *The Paradigm Dialog*, (Guba, E. G., Ed.), 17-27. Sage Publications, Newsbury Park, California.
- [116] Beach, D. (1996). *The Responsible Conduct of research*, VCH Publishers Inc., New York.
- [117] Morrow, R. A. (1996). *Critical Theory and Methodology*, Sage Publications.
- [118] Kuhn, T. S. (1962). *The Structure of Scientific Revolution*, The University of Chicago Press, Chicago, Illinois.
- [119] Lakatos, I. (1970). Falsification and the methodology of scientific research programmes. In *Criticism and the Growth of Knowledge* (Lakatos and Musgrave Eds). 91-196, Cambridge University Press.
- [120] Chalmers, A. F. (1999) 3rd edition. *What is this thing called Science?* Hackett Publishing Company.
- [121] Fulcher, A. J. and Hills, P. (1996). Towards a Strategic Framework for Design Research, *Journal of Engineering Design*, Vol. 7, No. 2, Journals Oxford Ltd.
- [122] Simon, H. A. (1981). *The science of the artificial*, 2nd Ed, MIT Press, Cambridge, MA, USA.
- [123] Lloyd, P. and Scott, P. (1994). Discovering the design problem, *Design Studies*, Vol. 15, No. 2, Butterworth-Heinemann Ltd.
- [124] Alberts, L. K. (1994). YMIR: A sharable ontology for the formal representation of engineering-design knowledge. In *Formal Design Methods for CAD* (J. S. Gero & E Tyugu Eds.), 3-32.
- [125] Duffy, A. H. B. & O'Donnell. (1998). A Design Research Approach, presented at the *AID98 Workshop on Research Methods in AI in Design*, AID'98, Lisbon, Portugal, 19 July 1998.
- [126] Sim, S. K. & Duffy, A. H. B. (1998). A formalism for coupled design/learning activities, presented at the *AID98 Machine Learning in Design Workshop*, AID'98, Lisbon, Portugal, 19 July 1998.

References Chapter 4

- [127] Reffat, R. M. and Gero, J. S. (2000). Computational Situated Learning in Design, In J. S. Gero and A. Christian (eds.), *Artificial Intelligence in Design*, '00, Kluwer Academic Publishers, The Netherlands.
- [128] Schon, D. A. (1987). *Educating the Reflective Practitioner*, JosseyBass Publishers.
- [129] Duffy, A. H. B., & Kerr, S. M. (1993). Customised perspectives of past designs from automated group rationalisation, *Artificial Intelligence in Engineering*, Vol. 8, Computational Mechanics Publication.
- [130] Kolb, D. A. (1984). *Experiential Learning*, Prentice Hall Pte. Ltd.
- [131] Polanyi, M. (1958). *Personal Knowledge: Towards a Post-Critical Philosophy*, London: Routledge and Kegan Paul.
- [132] Nonaka, I. and Takeuchi, H. (1995). *The knowledge creating company: how Japanese companies create the dynamics of innovation*, Oxford University Press.
- [133] Johnson-Laird, P. N. (1983). *Mental Models*, Cambridge University Press.
- [134] Anderson, J. R. (1976). *Language, memory and thought*. Hillsdale, NJ: Erlbaum.
- [135] Cohen, N. J., and Squire, L. R. (1980). Preserved learning and retention of pattern analysing skills in amnesia: Dissociation of knowing how and knowing what. *Science*, 210, 207-210.
- [136] Schacter, D. L. (1987). Implicit memory: History and current status. *Journal of Experimental Psychology: Learning, Memory and Cognition*, Vol. 13, 501-518.
- [137] Yan, M. (1993). Representing design knowledge as a network of function, behaviour and structure, *Design Studies*, Vol 14, No. 3, 314-329, Butterworth-Heinemann Ltd.

References Chapter 5

- [138] Simon, H. A. (1970). The structure of ill-structured problems, *Artificial Intelligence*, Vol. 4: 181-201
- [139] Logan, B. and Smithers, T. Creativity and Design as Exploration, in J. S. Gero & M. L. Maher, *Modeling Creativity and Knowledge-Based Creative Design*, Lawrence Erlbaum Associates, Hillsdale, New Jersey, pp 139-175.
- [140] Zienkiewicz, O. C. (1971). *The Finite Element Method in Engineering Science*, McGraw Hill, London.
- [141] Hubka, V. and Eder, W. E. (1996). *Design Science*, Springer.
- [142] Hubka, V. (1982). *Principles of Engineering Design*, Butterworth Scientific, London.
- [143] Chan, C. S. (1990). Cognitive processes in architectural design problem solving, *Design Studies*, Vol. 11, No. 2, 60-80, Butterworth & Co (Publishers) Ltd.
- [144] Stauffer, L., Ullman, D. G., Dietterich. (1987). T. G., Protocol Analysis of Mechanical Engineering Design. In the *Proceedings of the 1987 International Conference on Engineering Design*, Boston, Mass.
- [145] Stauffer, L. A. and Ullman, D. G. (1988). A comparison of the results of

- empirical studies into the mechanical design process, *Design Studies*, Vol. 9, No. 2, Butterworth & Co (Publishers) Ltd.
- [146] Ullman, D. G., Dietterich, T. G. and Stauffer, L. A. (1988). A Model of the Mechanical Design Process Based on Empirical Data, *AI EDAM (Artificial Intelligence for Engineering Design, Analysis and Manufacturing)*, Vol. 2, No. 1, 33-52, Academic Press Limited.
- [147] Takeda, H., Hamada, S., Tomiyama, T. and Yoshikawa, H., A. (1990). Cognitive Approach to the Analysis of Design Process, In J. R. Rinderle (Ed.), *Design Theory and Methodology*, 153-160, Chicago, September, ASME
- [148] Visser, W. (1992). Designers' activities examined at three levels: organisation, strategies and problem-solving processes, *Knowledge-Based Systems*, Vol 5, No. 1, March, 92-104, Butterworth-Heinemann Ltd.
- [149] Walters, R. J. (1986). Informed, well-ordered and reflective: design inquiry as action research, *Design Studies*, Vol 7, No. 1, January.
- [150] Crabtree, R. A., Fox, M. S., Baid, N. K. (1997). Case Studies of Coordination Activities and Problems in Collaborative Design, *Research in Engineering Design*, No: 9, 70-84, Springer Verlag.
- [151] Rittel, H. W. J. and Webber, M. M. (1973). Planning problems are wicked problems, *Policy Sciences*, Vol.4 155-169.
- [152] Duffy, S. M. (1995). The Design Complexity Map and the Design Coordination Framework. In the proceedings of the 10th Integrated Production Systems Research Seminar, Fuglso, Denmark, 3-5 April.
- [153] Gero, J. S. (1990). Design prototypes: a knowledge representation schema for design, *AI Magazine*, Vol 11. No. 4. 27-36.
- [154] Dasgupta, S. (1994). *Creativity in design and invention: Computational and Cognitive Explorations of Technological Originality*, Cambridge University Press.
- [155] Newell, A. (1990). *Unified Theories of Cognition*, Chapter 3, Harvard University Press, Cambridge, England.
- [156] Miles, J. and Moore, C. (1994). *Practical Knowledge-based Systems in Conceptual Design*, Springer Verlag.
- [157] Balkany, A., Birmingham, W. P. and Tommelein, I. D. (1993). An Analysis of Several Configuration Design Systems, *Artificial Intelligence for Engineering Design, Analysis and manufacturing*, Vol. 7, 1-17. Cambridge University Press.
- [158] Boothroyd, G., Dewhurst, P. and Knight, W. (1994). *Product Design for Manufacture and Assembly*, Marcel Dekker, Inc.
- [159] Bralla, J. G. (1996). *Design for Excellence*, McGraw Hill, Inc.
- [160] Roozenburg, N. F. M. & Eekels, J., *Product Design: Fundamentals and Methods*, John Wiley & Sons, 1995.
- [161] Fricke, G. (1996). Successful Individual Approaches in Engineering Design, *Research in Engineering Design*, Vol. 8, 151-165, Springer Verlag London Limited.
- [162] Eppinger, S. D., Whitney, D. E., Smith, R. P. and Gebala, D. A. (1994). A model-based method for organising tasks in product development. *Research in Engineering Design*, Vol. 6. No: 1, 1-13, Springer Verlag.
- [163] Hoover, S. P., Rinderle, J. R. and Finger, S. (1991). Models and Abstractions

- in design, *Design Studies*, Vol. 12, No. 4, 237-245, Butterworth-Heinemann Ltd.
- [164] Goel, V. and Pirolli, P. (1989). Motivating the Notion of Generic Design with Information Processing Theory: The Design Problem Space, *AI Magazine*, Spring, Vol. 10, No. 1, 18-36.
 - [165] Barzel, R. (1992). *Physically-Based Modelling for Computer Graphics - A Structured Approach*, Academic Press, Inc.
 - [166] Hetz, K. (1990). A coherent description of the process of design, *Design Studies*, Vol. 13, No. 4, 394-411.
 - [167] Ash, S. E., (1949). A reformulation of the problem of association, *American Psychology*, Vol. 24, 92-102.
 - [168] Osborn, A. F. (1963). *Applied imagination; Principles and Procedures of Creative Problem Solving*, 3rd edition. New York: Scribner.
 - [169] Pimmler, T. U., & Eppinger, S. D. (1994). Integration Analysis of Product Decompositions, DE – Vol. 68, *Design Theory and Methodology – DTM '94*, 343- 351, ASME.
 - [170] Challa, V. and Fadel, G. (1995). Design Data Decomposition, CED-Vol. 7, *Engineering Data Management and Emerging Technologies*, 53-65, ASME.
 - [171] Liu, J. and Brown, D. C. (1992). The generation of Decomposition Knowledge for Near Routine Design, in D. E. Grierson, G. Rzevski and R. A. Adey, (Eds.), *Applications of Artificial Intelligence in Engineering VII*, Computational Mechanics Publication.
 - [172] Liu, J. and Brown, D. C. (1994). Generating Design Decomposition Knowledge for Parametric Design Problems, in J.S. Gero and F. Sudweeks (Eds.), *Artificial Intelligence in Design '94*, 661- 678, Kluwer Academic Publishers.
 - [173] Hashim, F. M., Juster, N. P. and de Pennington, A. (1994). A Functional Approach to Redesign, *Engineering with Computers*, Vol. 10, 125-139, Springer-Verlag London Limited.
 - [174] Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y. and Tomiyama, T. (1996). Supporting conceptual design based on function-behaviour-state modeler, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 10, 275-288. Cambridge University Press.
 - [175] Kusiak, A. and Wang, J. (1993). Efficient organizing of design activities, *International Journal of Production Research*, Vol. 33, No. 4, 753-769.
 - [176] Kusiak, A. and Wang, J. (1993). Decomposition in Concurrent Design, in *Concurrent Engineering: Automation, Tools and Techniques*, Andrew Kusiak (Ed.), 481-507, John Wiley & Sons, Inc.
 - [177] Freeman, P. and Newell, A. (1971). A Model for Functional Reasoning in Design, *Proceedings, 2nd International Joint conference on Artificial Intelligence*, London, 621-633.
 - [178] Kota, S. and Lee, C. (1990). A functional framework for hydraulic systems design using abstraction/decomposition hierarchies, *Proceedings of the 1990 ASME International Computers in Engineering Conference*, 327-340.
 - [179] Welch, R. V. and Dixon, J. R. (1994). Guiding Conceptual Design Through Behavioural Reasoning, *Research in Engineering Design*, No: 6, 169-188, Springer Verlag.
 - [180] Bozzo, L. M. and Fenves, G. L. (1994). Qualitative Reasoning and the

- Representation of Fundamental Principles in Structural Engineering, *Research in Engineering Design*, No: 6, 61-72, Springer Verlag London Limited.
- [181] Qian, L. and Gero, J. S. (1996). Function-behaviour-structure paths and their role in analogy-based design, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 10, 282-312. Cambridge University Press.
 - [182] DeKleer, J. and Brown, J. S. (1984). A qualitative physics based on confluences. *Artificial Intelligence*, Vol. 24, 7-83.
 - [183] Thomson, S. (1996). *Design Activities*, MSc Thesis, University of Strathclyde.
 - [184] Ivezic, N. and Garrett, J. H. (1994)., A neural network-based machine learning approach for supporting synthesis, *Artificial Intelligence for Engineering Design, Analysis and manufacturing*, Vol. 8, 143-161. Cambridge University Press.
 - [185] Flemming, U., Adams, J., Carlson, C., Coyne, R., Fenves, S., Finger, S., Ganeshan, R., Garrett, J., Gupta, A., Reich, Y., Siewiorek, S., Sturges, R., Thomas, D & Woodbury, R. (1992). Computational Models for Form-Function Synthesis in Engineering Design. Technical Report EDRC 48-25-92, Engineering Design Research Centre, Carnegie Mellon University, Pittsburgh, PA.
 - [186] Alberts, L. K. (1993). *YMIR: an ontology for engineering design*, PhD thesis.
 - [187] Kannapan, S. M. & Marshek, K. M. (1996). A Comparative Analysis of Techniques in Engineering Design, in Manjula B. Waldron & Kenneth J. Waldron (Eds), *Mechanical Design: Theory & Methodology*. Springer-Verlag, New York.
 - [188] Hall, C. W. (1995). *The Age of Synthesis*, Peter Lang Publishing Inc.
 - [189] Ulrich, K. T. and Seering, W. P. (1989). Synthesis of Schematic Descriptions in Mechanical Design, *Research in Engineering Design*, Vol. 1, 3-18, Springer Verlag
 - [190] Chakrabarti, A. and Bligh, T. P. (1994). An Approach to Functional Synthesis of Solutions in Mechanical Conceptual Design. Part I: Introduction and Knowledge Representation, *Research in Engineering Design*, Vol. 1, 127-141, Springer Verlag London Limited.
 - [191] Chakrabarti, A. and Bligh, T. P. (1996). An Approach to Functional Synthesis of Solutions in Mechanical Conceptual Design. Part II: Kind Synthesis, *Research in Engineering Design*, Vol. 8, 52-62, Springer Verlag London Limited.
 - [192] Hansen, C. T. (1995). An Approach to Simultaneous Synthesis and Optimization of Composite Mechanical Systems, *Journal of Engineering Design*, Vol. 6, No. 3.
 - [193] Finn, D. P. (1993). Introduction: Preliminary Stages of Engineering Analysis and Modeling, *AI EDAM*, Vol. 7, No. 4, 231-237.
 - [194] Brooke, D. M., de Pennington, A., and Bloor, M. S. (1995). An Ontology for Engineering Analyses, *Engineering with Computers*, Vol. 11, 36-45.
 - [195] Finn, D. P. (1993). A Physical Modelling Assistant for the Preliminary Stages of Finite Element Analysis, *AI EDAM*, Vol. 7, No. 4, 275-286.
 - [196] Fruchter, R. Helmut, K. and Kincho, H. L. (1993). Qualitative Modeling and

- Analysis of Lateral Load Resistance in Frames, *Artificial Intelligence for Engineering Design, Analysis and manufacturing*, Vol. 7, 239-256. Cambridge University Press.
- [197] Starkey, C. V. (1992). *Engineering Design Decisions*. Edward Arnold.
 - [198] Eekels, J. (1990). An introduction to evaluation and decision in design, Evaluation and Decision in Design. In EVAD Reading, N. F. M. Roozenburg & J. Eekels (eds), *Schriftenreihe WDK 17*. Heurista, Zurich, 5-12.
 - [199] Akin, O. & Lin, Chengtah. (1996). Design Protocol Data and Novel Design Decisions, in Nigel Cross, Henri Christiaans, Kees Dorst (Eds), *Analysing Design Activity*, 33-63, John Wiley & Sons.
 - [200] Dwarakanath, S. & Wallace, K. M. (1995). Decision-making in Engineering Design: Observations from Design Experiments, *Journal of Engineering Design*, Vol. 6, No. 3, 191-206.
 - [201] McGinnis, B. and D. G. Ullman, (1992). The evolution of commitments in the design of a component, *Journal of Mechanical Design*, 1-7.
 - [202] Chan, W T. and Paulson, B. C. (1987). Exploratory Design using Constraints, *Artificial Intelligence for Engineering Design, Analysis and manufacturing*, Vol. 1, 59-71. Cambridge University Press.
 - [203] Cross, N. & Cross, A. C. (1996a). Observations of Teamwork and social processes in Design, in Nigel Cross, Henri Christiaans, Kees Dorst (Eds), *Analysing Design Activity*, 291-317, John Wiley & Sons.
 - [204] Baya, V. & Leifer, L. J. (1996). Understanding Information management in Conceptual Design, in Nigel Cross, Henri Christiaans, Kees Dorst (Eds), *Analysing Design Activity*, 151-168, John Wiley & Sons.
 - [205] Reitmann, W. R. (1994). Heuristic decision procedures, open constraints, and the structure of ill-defined problems. In *Human Judgements and Optimality*, eds. M. W. Shelly & G. L. Bryan. John Wiley, New York.
 - [206] Goel, V., A. (1994). comparison of design and non-design problem solving spaces, *Artificial Intelligence in Engineering*, Vol. 9, 53-72.
 - [207] Logan and Smithers, T. (1993). Modelling Creativity and Knowledge-Based Creative Design, in J. S. Gero & M. L. Maher (Eds.), Lawrence Erlbaum Associates Publishers.
 - [208] Brazier, F. M. T., van Langen, P. H. G. and Treur J. (1995). Modelling conflict management in design: An explicit approach, *Artificial Intelligence for Engineering Design, Analysis and manufacturing*, Vol. 9, 353-366. Cambridge University Press.
 - [209] Oh, V. and Sharpe, J. (1995). Conflict management in an interdisciplinary design environment, *Artificial Intelligence for Engineering Design, Analysis and manufacturing*, Vol. 9, 247-258. Cambridge University Press.

References Chapter 6

- [210] Leith, P. (1990). Formalism in AI and Computer Science, Ellis Horwood Limited.
- [211] Grecu, D. L. And Brown, D. C. (1996). Dimensions of Learning in agent-based Design, in A. H. B. Duffy, M. L. Maher, and D. C. Brown, eds. *Proc. Machine Learning in Design, Workshop, (AID '96: Fourth International*

- Conference of Artificial Intelligence in Design), Kluwer Academic Publishers.
- [212] Vancza, J. (1991). Improving Design Knowledge by Learning, in *Intelligent CAD III* (Eds.) H. Yoshikawa, F. Arbab and T. Tomiyama, IFIP, Elsevier Science Publishers, 289-272-305.
 - [213] Reich, Y. (1993). The development of BRIDGER: A methodological study of research in the use of machine learning in design, *Artificial Intelligence in Engineering*, Vol. 8, No. 3, 165-181.
 - [214] Persidis, A. (1989). *Modelling of abstractions for computer-aided design*, Ph. D. Thesis, CAD Centre, University of Strathclyde, Glasgow, Scotland.
 - [215] Li, H. (1994). *Machine Learning in Design Concepts*, Computational Mechanics Publication.
 - [216] Chabot, R. and Brown, D. C. (1994). Knowledge Compilation using constraint inheritance, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 8, 125-142, Cambridge University Press.
 - [217] Brown, D. C. (1991). Compilation: The hidden dimension of design systems. *Proc. 3rd IFIP WG 5.2 Workshop on Intelligent CAD*, Osaka, Japan, *Intelligent CAD III* (Yoshikawa, H., Arbab, F., and Tomiyama, T., Eds.), 99-108, North Holland, Amsterdam.
 - [218] Liu, J. and Brown, D. C. (1992). The generation of Decomposition Knowledge for Near Routine Design, in D. E. Grierson, G. Rzevski and R. A. Adey, (Eds.), *Applications of Artificial Intelligence in Engineering VII*, Computational Mechanics Publication.
 - [219] Liu, J. and Brown, D. C., Generating Design Decomposition Knowledge for Parametric Design Problems, in J.S. Gero and F. Sudweeks (Eds.), *Artificial Intelligence in Design '94*, 661- 678, Kluwer Academic Publishers.
 - [220] Murdoch, T. and Ball, N. (1996). Machine Learning in configuration Design, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 10, 101-113, Cambridge University Press.
 - [221] McLaughlin, S. & Gero, J. S. (1987). Acquiring Expert Knowledge from Characterised Designs, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 1, 73-87, Cambridge University Press.
 - [222] Bhatta, S. R. and Goel, A. K. (1996). From design experiences to generic mechanisms: Model-based learning in analogical design, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 10, 131-136, Cambridge University Press.
 - [223] Erman, L. E., Hayes-Roth, F., Lesser, V. R. and Reddy, D. R. (1980). The Hearsay-II Speech-understanding System: integrating Knowledge to Resolve Uncertainty, *Computing Surveys* 12, 213-253.
 - [224] Knapp, D. and Parker, A. (1991). The ADAM Design Planning Engine, *IEEE Transactions on the Computer-Aided of Integrated Circuits and Systems*, Vol. 10. No 7.
 - [225] Mostow, J. (1989). Design by Derivational Analogy: Issues in the Automated Replay of Design Plans, *Artificial Intelligence*, Vol. 40, 119-184.
 - [226] Huhns, M. N. & Acosta, R. D. (1992). An Analogical Reasoning System for Solving Design Problems, In Chris Tong (Ed.) *Artificial Intelligence in Engineering Design*, Vol. 2, 105-143, Academic Press.

- [227] Tong, C. (1992). Using Exploratory Design to cope with Design Problem Complexity in Tong (Ed.), *Artificial Intelligence in Engineering Design*, Vol. 2, Academic Press, 287-332.
- [228] Wang, J. & Howard, H. C. (1994). Recording and reuse of design strategies in an integrated Case-based Design System, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 8, 219-238, Cambridge University Press.
- [229] Gluck, M. & Corter, J. (1985). Information, uncertainty and the utility of categories. Proceedings of the Seventh Annual Conference of the Cognitive science Society, 288-287, Irvine, CA: Lawrence Erlbaum.
- [230] Mostow, J., Barley, M. and Weinrich, T. (1992). Automated Reuse of Design Plans in BOGART, in Tong(ed.) *Artificial Intelligence in Engineering Design*, Vol. 2, Academic Press, 287-332.
- [231] Carbonell, J. G. (1983). Derivational Analogy and its role in Problem Solving, In: *Proceedings AAAI-83*, Washington, DC. 64-69.
- [232] Carbonell, J. G. (1986). Derivational analogy: A theory of reconstructive problem solving and expertise acquisition. In: R. S. Michalski, J. G. Carbonell and T. M. Mitchell (Eds), *Machine Learning: An Artificial Intelligence Approach 2* (Morgan Kaufmann, Los Altos, CA), 371-392.
- [233] Dietterich, T. G. and Michalski, R. S. (1983). A comparative review of selected methods for learning from examples, in R. S. Michalski et. Al. (eds.) *Machine Learning: An Artificial Intelligence Approach*, 41-81, Morgan Kaufmann, 1983.
- [234] Steinberg, L. I. (1992). Design as top-down refinement plus Constraint propagation: In Chris Tong (Ed.) *Artificial Intelligence in Engineering Design*, Vol 1, Academic Press, 251-272.
- [235] Mitchell, T., Mabadevan, S. and Steinberg, L. I. (1990). LEAP: A Learning Apprentice for VSLI Design. In: Yves Kodratoff and Ryszard Michalski (Eds.). *Machine Learning, An Artificial Intelligence Approach*, Volume III, 271-301, Morgan Kaufmann.
- [236] Mitchell, T., Keller, R. and Kedar-Cabelli, S. (1986). Explanation-based generalisation: A unifying view. *Machine Learning*, Vol. 1, No. 1.
- [237] DeJong, G. and Mooney, R. (1986). Explanation Based Learning: An Alternative View, *Machine Learning*, Vol. 1, No. 2, 145-176.
- [238] Britt, B. D. and Glagowski, T. (1996). Reconstructive Derivational Analogy: A machine learning approach to automate redesign, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 10, 115-126, Cambridge University Press.
- [239] Bhatta, S. R. and Goel, A. K. (1994). Discovery of physical principles from design experiences, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 8, 113-123, Cambridge University Press.
- [240] Ashok, A. K. and Stroulia, E. (1996). Functional device models and model-based diagnosis, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 10, 355-370, Cambridge University Press.

References Chapter 7

- [241] Dasgupta, S. (1994). Creativity in Invention and Design: Computational and

- Cognitive Explorations of Technological Originality. Cambridge University Press.
- [242] Newell, A. (1982). The Knowledge Level. *Artificial Intelligence*. Vol. 18(1). 87-127.
 - [243] Dietterich, T., Learning at the knowledge level, *Machine Learning*, Vol. 1, 121-138, 1986.
 - [244] Kocabas, S., A review of learning, *The Knowledge Engineering Review*, Vol. 6, No. 3, 195-222, 1991.
 - [245] Wilkes, M. V., The Best Way to Design an Automatic Calculating Machine, Report of the Manchester University Computer Inaugural Conference, Manchester, 118-121, July 1951. Reprinted in Wilkes (1986) [246].
 - [246] Wilkes, M. V., The Genesis of Microprogramming, *Annals of the History of Computing*, Vol. 8, No. 2, 116-126, 1986.
 - [247] McMahon, C. A. (1994). Observation on Modes of Incremental Change in Design, *Journal of Engineering Design*, Vol. 5 No. 3. 195- 209.
 - [248] Grecu, D. L. & Brown, D. C. (1996). Design agents that learn, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 10, 149-150, Cambridge University Press.
 - [249] Busby, J. A. and Lloyd, P. A. (1999). Influences on Solution Search Processes in Design Organisation, *Research in Engineering Design*, Vol. 11, 158-171, Springer-Verlag London Limited.
 - [250] Darke, J. (1979). The primary generator and the design process. *Design Studies*, Vol. 1. 191-206.
 - [251] Boston, O. P., Court, A. W., Culley, S. J. & McMahon, C. A. (1997). Design information issues in new product development, in: Duffy, A. H. B. (Ed.) *The Design Productivity Debate*. 231-254. Springer Verlag, London.
 - [252] Allen, T. J. (1969). The Differential Performance of Information Channels in the Transfer of Technology. In: Grubber, W. H. and Marquis, D. G. (Eds). *Factors in the Transfer of technology*, MIT Press, Cambridge, MA, USA.
 - [253] Bond, A. H. and Ricci, R. J. (1992). Co-operation in Aircraft Design. *Research in Engineering Design*, Vol. 4. 115-130.
 - [254] Court, A. W., Culley, S. J. & McMahon, C. A. (1994). Information Sources and Storage Methods for Engineering Data. *Proceedings of the 2nd ASME Biennial European Joint Conference on Engineering Systems Design and Analysis*, PD-Vol. 64-5, 9-16.

References Chapter 8

- [255] Cross, N. Christiaans, H and Dorst K. (Eds.). (1996). *Analysing Design Activity* John Wiley & Sons.
- [256] Titchener, E. B. (1912). The schema of introspection. *American Journal of Psychology*, Vol 23, 485-508.
- [257] Ericsson, K. A. and Simon, H. A. (1993). *Protocol Analysis, Verbal Reports as Data*, The MIT Press.
- [258] Mc Neill, T., Gero, J. S., and Warren, J. (1998). Understanding Conceptual Electronic Design using Protocol Analyses *Research in Engineering Design*, Vol. 10, 129-140.

- [259] Frankenberger, E. and Auer, P. (1997). Standardised Observation of teamwork-work in Design, *Research in Engineering Design*. Vol. 9, 1-9. Springer-Verlag London Limited.
- [260] Valkenburg, R. and Dorst, K. (1998). The reflective practice of design teams, *Design Studies*, Vol. 19. 249-271. Elsevier Sciences Ltd.
- [261] Gero, J. S. and Mc Neill, T. (1998). An approach to the analysis of design protocols, *Design Studies*, Vol. 19 No. 1, 21- 61.
- [262] Purcell, T., Gero, J., Edwards, H., and McNeill, T. (1996). The Data in Design Protocols: The Issue of Data Coding, Data Analysis in the development of Models of Design Process, in Cross, Christiaans & Dorst (eds), *Analysing Design Activity*, 35-63, John Wiley & Sons.
- [263] Van Someren, M. W., Barnard, Y. F. and Sandberth, J. A.C. (1994). The think aloud method: a practical guide to modelling cognitive processes, Academic Press, London.
- [264] Goldschmidt, G. (1991). The dialectics of sketching, *Creativity Research Journal*, Vol. 4, No. 2, 123-143.
- [265] Coyne, R. D., Rosenman, M.A., Radford, A. D., Balachandran, M. and Gero, J. S. (1990). *Knowledge-based Design systems*, Addison-Wesley Publishing Company.
- [266] Baykan, C. (1996). Design Strategies. In Cross, N. Christiaans, H and Dorst K. (Eds.), *Analysing Design Activity*, John Wiley & Sons.

References Chapter 9

- [267] Duffy, A., (1998a). AID'98 Machine Learning in Design, Workshop: Saturday 18 July 1998: Outcome.
<http://www.cad.strath.ac.uk/~alex/AID/AID98/outcome.html>.

References Chapter 10

- [268] Birkmire, J. and Pluempitiwiriawej, C. (1996). The Scientific Method in Research, In McCuen, R. H. *The Elements of Academic Research*, 24-36, ASCE Press.
- [269] Duffy, A. H. B. and Andreasen, M. M. (1995b). Enhancing the Evolution of Design Science, Proceedings of the International Conference of engineering Design, ICED95.
- [270] Reddy, J. M., Chan, B. & Finger, S. (1998). Patterns in design discourse: A case study, in *Knowledge Intensive Design*.
- [271] Perry, M and Sanderson, D. (1998). Coordinating joint design work: the role of communication and artefacts, *Design Studies*, 273-288, Elsevier Science Limited.
- [272] Klein, M and Lu, Stephen C. Y. (1989). Conflict resolution in cooperative design, *Artificial Intelligence in Design*, Vol 4, 168- 180, Computational Mechanics Publication.
- [273] Spoehr, K and Lehmkuhle, S. W. (1982). Visual information processing, W. H. Freeman and Company.

Time/Seg No		Design Elements	Learning Elements or Knowledge Change
00:00:00/1	The length corresponds to the length and breadth characteristics of the vessel currently under construction in America for the Royal Bahamas Defence Force. These vessels are 60 m long with 9m beam. They are presently propelled by three 1600 kW 16 cylinder diesel engines, Caterpillar engines. These together with the loading of the vessel, the requirement of the vessel, small guns, 60 crew propel this vessel at round about 24 knots. The vessel has a range of 2,000 nm at 14 knots. This is chosen as the most suitable for their sea-keeping conditions in the area. They want long range off shore patrol boats with good sea-keeping characteristics. As this boat is under construction, experience has been gained in building this boat. This seems a logical choice as a starting point in developing a new design performance requirements which defers to some extent in requirements are now: • Maximum speed of some 38 to 41 knots. • A requirement for highly sophisticated control system, a need to have good minimum man management the of systems, so that a lot of the function of the weapon system to be incorporated in the vessel.	Introductory comments by the ship designer	

	The vessel is also, as it's being built in US, with US money, there is a desire to fit as much as practical USA manufactured equipment rather than purchased it from elsewhere. This poses some difficulty because the marine market in the States has not been geared to the manufacture of small patrol craft or small warship. Mainly the American fleet is a blue water deep ocean going fleet. Their systems, weapons systems are geared for much larger crafts. There's a lot of home-made systems that suit the particular requirements. Therefore we've to look to overseas for different suppliers maybe some with USA connection which will make selection with some situation more acceptable to the USA government.
00:05:00/ 2	So, we've done some preliminary powering requirements for this vessel and have established there is a need for something in the order (I don't know the figure at hand which is X number of megawatts. This falls into two major areas of selection either 6 engines or 4 engines of a larger capacity. We're advised that the \$/kW price of the 4 engines applications is prohibitive and that we should concentrate our effort on 6 engines application. So this in itself is the driver to the design. This adds to the complication of having to accommodate 6 propulsors in a relatively small envelope.

• D_g = preliminary powering requirement.
 D_a = **Analysis**.
 I_k = details of vessel.
 O_k = X megawatts.
 • D_g = type of engines.
 D_a = **Exploring**.
 I_k = X megawatts.
 O_k = 6 engines or 4 engines.
 • D_g = prices of engine.
 D_a = **Information gathering**.
 I_k = 4 engine or 6 engine option.
 O_k = \$/kW price of 4 engines >> \$/kW price for 6 engines.
 • D_g = engine option.
 D_a = **Decision Making**.
 I_k = Price/kW of engine options.
 O_k = accommodate 6 engines.

• $K_r = Dr$
 $T_r = \text{In- situ } (L_{ri})$
 ΔK = Knowledge of power requirements from attributes of vessel.
 • $K_r = Di$
 $T_r = \text{In- situ } (L_{ri})$
 ΔK = Knowledge of options of engine type for given power requirements.
 • $K_r = Ac$
 $T_r = \text{In- situ } (L_{rp})$
 ΔK = Knowledge of prices for engine options.
 • $K_r = Dr$
 $T_r = \text{In- situ } (L_{ti})$
 ΔK = Knowledge of power requirements from attributes of vessel.

00:06:00/ 3	We can see straightaway this compromises certain aspects of arrangement. By the time we accommodate propulsors, the auxiliary and maybe weapons systems what little space is left we've to squeeze in logistics requirements of the vessel in terms of the stores that it need to carry to survive at sea for tens days, the number of people.
00:07:00/ 4	Notwithstanding these requirement we've to satisfy sensible routes for the instalment and ready removal of equipment that is likely to be replaced or serviced. This adds another driver to where these equipments may or may not be sited.
/5	First things first. We must look at how we will put the 6 engines in the vessel. Now, 6 engines of MTU 4,000 series, eh, the engines which are being proposed. There are perhaps one or two ways to put into the vessel sensibly.
/6	But there are maintenance considerations that we'll have to consider - the ease of the operatives to move about the engine spaces to conduct maintenance in and out of vessel. The ability to get them in and out of the engine space.

• D_g = overall arrangement. D_a = Decomposing. I_k = Ship space envelope. O_k = compromise allocation of space for engine, auxiliary, weapons systems, logistics.	• K_r = Decomposition T_r = In- situ (L_{ii}) ΔK = Knowledge that the ship space envelope has to be divided into functional spaces (e.g. propulsion, weapons systems etc.)
• D_g = Sensible routes for installation and removal of equipment. D_a = Identifying key issue. I_k = engine room configuration, weapons systems etc. O_k = Locations of equipment.	• K_r = Association T_r = In- situ ΔK = Knowledge of factors in configuring equipment.
• D_g = Location of engines. D_a = Prioritising I_k = engine room configuration, weapons systems etc. O_k = Locations of 6 engines in vessel.	
• D_g = Locations of 6 engines in vessel. D_a = Exploring. I_k = 6 engines of MTU 4,000 series. O_k = 2 ways of locating engines in vessel.	• K_r = Discovery T_r = In- situ/Retrospect ΔK = Knowledge of the options for engine room configuration.
• D_g = Locations of 6 engines in vessel. D_a = Identifying evaluation criteria. I_k = Maintenance consideration. O_k = a) Ease of operatives to move about engine room b) ability to get engines in and out of engine space.	• K_r = Association T_r = In- situ ΔK = Knowledge of criteria for location of engines

/7	How to get 6 engines, 6 propellers in a row? We could have 6 lines of shafting, 6 separate shafts. But the practicality is that the vessel could not accommodate 6 propellers in a row, let alone six engines.
00:09:00/ 8	Can we put 4 engines abreast? A pair of engines driving a single propeller and arrange engines in staggered format - in one area 4 engines abreast and in another area, 2 engines. Inspection of these prospects is shown on screen and as can be seen the ability to put 4 engines in a row is just grossly impractical. There is no room, there is no room for the ship structure, engines have to go in at heights which made them difficult to maintain and access, would raise the CG and the impact of the overall stability of vessel. So, at a glance we can see that this is not a sensible approach.
/9	So we'll look at another consideration. <i>We know from past installations</i> it is possible to put engine in U-drive gearbox which enable the engine to be placed in reverse arrangement with a stub shaft coming from the engine to the gearbox and then the main propulsion shaft coming out underneath the engine and running aft to the propellers. But this in itself may be a solution but as previously mentioned we can't accommodate 6 propellers.

• D_g = 6 engines and 6 propellers in a row. D_a = Generating concept. I_k = 6 engines & 6 propellers. O_k = Layout showing 6 engines connected to 6 propellers in a row. • D_g = Is concept feasible. D_a = Evaluating concept. I_k = Layout showing 6 engines connected to 6 propellers in a row. O_k = concept is not feasible as it is not possible to arrange 6 propellers in a row.	• K_r = Derivation T_r = In- situ ΔK = Knowledge of constraints on the design space (i.e. space constraint -not feasible to arrange 6 engines in a row).
• D_g = 4 engines abreast. D_a = Generating concept. I_k = 6 engines & 6 propellers. O_k = Layout showing two pairs of engines connected to a propellers each and 1 pair of engines to a propeller in another area. • D_g = Is concept feasible. D_a = Evaluating concept. I_k = Layout showing two pairs of engines connected to a propellers each and 1 pair of engines to a propeller in another area. O_k = concept not feasible because of space constraint.	• K_r = Derivation T_r = In- situ ΔK = Knowledge of constraints on the design space (i.e. space constraint -not feasible to arrange 4 engines in a row). • K_r = Derivation T_r = In- situ ΔK = Knowledge of reasons for infeasible concept
• D_g = 4 engines abreast. D_a = Generating concept. I_k = 6 engine connected to 6 U drive gearbox (GB) with stub shaft running under the engine. O_k = Layout showing engine and U- drive GB. • D_g = Is concept feasible. D_a = Evaluating concept. I_k = Layout showing 6 U drive to drive gearbox through stub shaft. O_k = concept is not feasible as it is not possible to arrange 6 propellers in a row.	L_g = engine layout. K_r = Similarity comparison (Sc) I_k = Similar past installation of engines using U-drive gearbox. O_k = engine placed in reverse connected to U drive gearbox with stub shaft under running underneath the engine. T_r = Retrospective. • K_r = Explanation (Ex) T_r = In- situ ΔK = Reason for infeasible design concept.

/10	The most we can fit in the width of the ship appears to be three at any one time.
/11	We need to fit some sort of gearbox that enables 2 engines to be connected to the one gearbox with a propulsion shaft. So, we <i>know</i> from commercial catamaran construction in the market place that very narrow hulls eh, have to accommodate a lot of engine installation and we <i>know</i> that there are combined gearboxes that eh, enable so many engines to be joined into one drive train.
/12	What we must do is communicate with gearbox manufacturers to find if there is anything suitable for this particular application and for this particular power.
00:13:00/ 13	We know that these are typically available for water-jet application. What needs to be done is to establish if reverse reduction capability for conventional propeller arrangements. The gearbox manufacturer has come back to us and has made a recommendation but it's only a tentative one as it happen to be a gearbox suitable for water-jet which doesn't have a reverse requirement. eh, because for reversing in water-jets, manipulation of the reversing bucket over the jet nozzle gives the reversing capability.

• D_g = 4 engines abreast. D_a = Constraining. I_k = Configuration of 4 engines or 6 engines abreast. O_k = At most 3 engines abreast.	• K_i = Generalisation (Ge) T_r = In- situ ΔK = A general rule for this specific vessel that at most three engines abreast.
• D_g = A gearbox which connects to 2 engines and output to propulsion shaft. D_a = Generating concept. I_k = 6 engines and 6 propellers O_k = combined gearbox.	• K_i = Similarity comparison (Sc) T_r = Retrospect ΔK = Application of past design concept using combined gearbox to generate a similar concept. L_{nv} = <i>Novelty of concept</i> from catamaran application that can be applied to current application.
• D_g = A suitable combined gearbox D_a = Information gathering. I_k = type of application and power O_k = response from gearbox manufacturer.	• K_i = Acquisition T_r = Provisional ΔK = Knowledge of a combined gearbox with suitable power
• D_g = A suitable combined gearbox with reverse reduction capability D_a = Information gathering from gearbox manufacturer. I_k = combined gearbox with reverse reduction capability for conventional propeller application O_k = Tentative response from gearbox manufacturer of a suitable gearbox but without reversing capability.	L_g = engine layout using combined gearbox (GB). K_i = Similarity comparison (Sc) I_k = Similar past installation of engine in waterjet application. O_k = engine connected to combined GB. T_r = Retrospective.

14	So, for a propeller version, you actually have to have a reverse reduction gearbox. The gearbox manufacturer is looking into a situation to see what impact the size of the gearbox to feature this reverse capability.
15	So, where do we start to put the engines? Put gearbox and forward set of engines in a forward machinery space the engines will be close coupled to the gearbox minimising any space take up. And then from a redundancy consideration we would like to think how to put after engine in a separate compartment dividing the two compartments with water-tight bulkhead.
16	One of the major design parameters for the vessel of this size is its damaged stability characteristics. There is a requirement in naval vessels for vessels in excess of 30 m and up to 90 m waterline length, it should survive damage of any two adjacent compartments. On the dividing bulkheads, there are sets of stability, intact stability requirements.
17	Where else can we consider putting the engines? The engine should be put in such a manner that where the propulsion shafting angled with respect to the buttock lines of the vessel in which the shaft line is arranged 6 to 12 degrees maximum in this category. This basically means that the angle that exists to another arbitrary line. This is a line which is representative of the buttock lines of the ship in way of the line of the shaft.

• D_g = A suitable combined gearbox (GB) with reverse reduction capability D_a = Information gathering from gearbox manufacturer. I_k = combined gearbox with reverse reduction capability for propeller application O_k = Size of GB to feature reverse capability.	L_g = engine connected to propeller K_r = Association (As). I_k = Similar past installation of engine connected to propeller. O_k = Gearbox requires reverse reduction. T_r = Retrospective.
• D_g = Arrangement of 6 engines. D_a = Configuring engine room layout. I_k = 6 engines & 6 propellers. O_k = Layout showing 4 engines in one watertight compartment and two in another compartment.	• K_r = Abstraction of engine room layout T_r = In- situ ΔK = Knowledge of feasible engine room layout.
D_g = Damage stability characteristics. D_a = Identifying key issue. I_k = Vessel in excess of 30 m and up to 90 m waterline length. O_k = Survive damage of any two adjacent compartments.	• K_r = Association of engine room layout with damage stability criteria T_r = In- situ ΔK = Knowledge of applying damaged stability criteria to 2 adjacent engine compartments.
α: Angle between shaft line and buttock lines • D_g = Location of engines. D_a = Constraining. I_k = Elevation of engine and shaft angle in relation to the buttock lines. O_k = $6 < \alpha < 12$	• K_r = Specialisation (Sp) T_r = In- situ ΔK = Application of a general rule of thumb: $6 < \alpha < 12$

18	And what we've to do is check the angle between the two. But the included angle between buttock and the propeller shaft is 8 degrees. In this present configuration that seems very reasonable.
00:18:00 19	This would tell me that it may well be possible to move engines further aft. This would have the impact of increasing the included angle, but then I've to actively consider where these engines in the overall context are in the overall position of the boat. These engines can't move much further aft because we may get out of balance of the boat.
/20	What is the other consideration? We <i>know</i> that from separate consideration that the propeller of the order of 1.8 m. This is typically derived by calculation to absorb power of the combined engines. {by calculation}. This is taking a typical 3 bladed propeller. We <i>know</i> that there are greater advances and more efficiency to be obtained from a 5-bladed propeller or more. But as an initial design consideration if we work on <i>what we know</i> for ease of quick calculation we can conclude that 1,800 mm diameter is what is necessary to absorb the power and we're assuming propeller speed propeller shaft speed of something in the order of 700 rpm.

• D_g = Check is α acceptable. D_a = Evaluating α. I_k = α is 8 degrees. O_k = Present location of engine is acceptable.	• K_r = Specialisation T_r = In- situ ΔK = Knowledge of satisfying a design constraint (i.e. $\alpha = 8$).
• D_g = Locations of engines in vessel. D_a = Exploring. I_k = Engine at current location. O_k = Move engine aft. • D_g = Locations of engines in vessel. D_a = Decision Making. I_k = Engine at aft location. O_k = Engine remain previous location because of trim problem.	• K_r = Derivation T_r = In- situ ΔK = Knowledge of relationship between engine and shaft angle. • K_r = Explanation T_r = In- situ ΔK = Knowledge of rationale of maintaining current location
• D_g = Type of propeller: 3 or 5 blade . D_a = Exploring. I_k = 5 bladed more efficient than 3 bladed. O_k = Choose 3 bladed for quick estimation. • D_g = Quick estimation of propeller diameter. D_a = Qualitative analysis. I_k = Similar past design of 3-bladed propeller and shaft speed of 700 rpm. O_k = Propeller diameter = 1.8 m. <i>Application of past design knowledge on propeller design to current design.</i>	L_g = Size of propeller for a given power K_r = Similarity comparison (Sc). I_k = Similar past design calculations of 3 bladed and 5 bladed propellers. O_k = 1.8 m, 5 bladed more efficient than 3 bladed. T_r = Retrospective. L_{nw} = <i>expedience driven</i> as there was no need to go through length calculation in deriving a quick but approximate propeller diameter.

21	A particular development of the hulls, these Vosper hulls is the in-built wedge characteristics at the back end of the boat to provide some lift.
00:20:00/22	And more recently, another innovation which is the vibration attenuation caves which are fitted directly over the propellers. These enable us to fit propellers with clearances a lot less to the hull than what would ordinarily be, perhaps to avoid vibration being transmitted to the hull. This attenuation panel made from elastomeric membrane is suspended from watertight box directly in way of each propeller. With this facility or characteristic we assume a propeller tip clearance of 15% of propeller diameter to work from. What is the membrane face (i.e.the elastomeric membrane face) to what is the centreline of the propeller? This gives us a datum point to the aft end of the vessel.
00:21:00	

A statement of design practice but no design action implied. <i>Application of design rule on Vosper hull to current design.</i>	L_g = Design feature of Vosper hull K_r = Generalisation (Ge). I_k = In-built wedge at stern of hull provides some lift. Past Vosper hull fitted with wedge. O_k = A rule for fitting built-wedge for all Vosper hulls. T_r = Retrospective.
• D_g = Minimise propeller induced vibration D_a = Resolving conflicts. I_k = a) Decrease tip clearance increases hull vibration; b) VAC reduces hull vibration O_k = Use VAC to decrease clearance between propeller and hull. <i>Application of knowledge learnt on VAC to resolve conflict to current design.</i> • D_g = Minimise propeller induced vibration D_a = Defining design parameter. I_k = a) Use VAC reduces hull vibration. O_k = propeller tip clearance of 15% propeller diameter.	L_g = Reduce hull vibration. K_r = Sc. I_k = To reduce vibration requires high propeller tip clearance. O_k = Fit vibration attenuation cave fitted over propellers. T_r = Retrospective (L_{rr}). • ΔK = Using a vibration attenuation cave (VAC) reduce hull vibration with reduced propeller tip clearance. K_r = Detailing T_r = Retrospect (L_{rr}) L_{nv} = <i>conflict avoidance</i> as VAC can reduce vibration and yet achieved reduce propeller tip clearance.

00:22:00/
23

So we already decided from *previous experience* we might use the benefits and *experience gained from previous craft*.

We know the rudder maybe perhaps a similar size maybe slightly different.

We *know* from a clearance point of view between the rudder and propeller we want to set the propeller as far aft as possible and clear of the rudder.

We use *all the advantages* that we have *on the previous boat*. So what we have manage to do now is to dictate the position of the propeller centre.

/24

We have the envelope shape of the engine plus gearbox. And what we must check, now is whether the bottom of the gearbox is going to be within the hull envelope. And what we need to do is perhaps draw a section through the vessel in way of the gearbox just to confirm that the dead-rise angle at this position of the vessel won't cause the gearbox to penetrate the hull.

So nominally the centre of the gearbox is taken to the edge of the gearbox at Frame 37. So we need to look at round about section 37.

Imported is the section of the ship's body plan.

We need to identify which is which and just pull 37 out and just do a check here at 37. So I need to copy that. Just pull 37 out. So we just do a check. So this is the gearbox at Frame 37. Just above it.

Comment on applying past design experience to current design. • D_g = Dictate position of propeller centre. D_a = Defining. I_k = Similar past design case. O_k = Locate propeller as far aft from the rudder <i>Application of past design knowledge on rudder to current design.</i>	L_g = Size of rudder. K_r = Similarity comparison I_k = Similar past design of rudder. O_k = Size of rudder. T_r = Retrospective L_g = Maximum clearance between rudder and propeller. K_r = Detailing. I_k = Location of rudder in relation to propeller of a past design case. O_k = Distance between rudder and propeller. T_r = Retrospective. L_{nv} = <i>expedience driven</i> as location of the propeller can be defined easily with respect to the rudder.
• D_g = Clearance between gearbox and hull. D_a = Identify interference between gearbox and hull. I_k = Envelope shape of gearbox and hull. O_k = Draw section through the vessel in way of the gearbox. • D_g = Clearance between gearbox and hull. D_a = Identify section at Frame 37 I_k = Section 37 of Body plan. O_k = Copy section 37 to current drawing. • D_g = Clearance between gearbox and hull. D_a = Evaluating clearance I_k = Gearbox sump and Frame 37. O_k = No interference (Just above it).	• K_r = Association T_r = In- situ ΔK = Knowledge of spatial relationship between gearbox and structure. • K_r = Derivation T_r = In- situ ΔK = Knowledge of procedure to check clearance

/25	Now, going to plan view. The engine has been placed in the boat to provide good separation of propeller, good separation between engines to satisfy maintenance requirements. So from this we've determined the approximate distance of the shaft off the centreline of 2.55 m. So in the section we've got we represent that position. So 2550. Now I need to determine where the baseline of the ship is. Now, what I fail to do the last time. 44, 57, Copy this from there to there. I should have done it before. [Drawing section]
/26	Right. So what we have done is to put in a uniform angle of shafting as a starting point. And, eh put in a 5 degree. And it doesn't have to be a round figure. But it makes life easier in remembering figures when juggling about the screen. The 5 degree is borne out of the eh propeller centreline. And what on the face of it there is sufficient clearance of the gearbox sump from the bottom of the vessel at the given frame position. So in manipulating and moving between different views and calculating what the shaft line might be at a given position, it looks reasonably clear. 5 degree seems to fit in that quite nicely.
27	So take a dimension to check and it is 1.708 m. So offset the baseline. Drive shaft centre at Frame 37. It is 1.708. This is where the propeller centre or the shaft centre is situated.

• D_g = Provide good separation of propellers and engines to satisfy maintenance requirements. D_a = Determining distance of shaft from centreline, S I_k = Plan view. O_k = S is 2550 • D_g = Same as above D_a = Determining location of baseline of ship. I_k = Elevation view. O_k = From drawing	• K_r = Detailing T_r = In- situ ΔK = Knowledge of geometric relationship between design objects (shafts from centreline)
D_g = Optimal shaft angle. D_a = Defining. I_k = Value of shaft angle. O_k = Shaft angle = 5 degree. D_g = Is shaft angle acceptable. D_a = Evaluating. I_k = Criteria: sufficient clearance between gearbox sump and bottom of vessel. O_k = Clearance is acceptable, so shaft angle is acceptable.	• K_r = Specialisation. T_r: In- situ. ΔK = Knowledge of design object attribute in relation to design object • K_r: Derivation. T_r: In- situ. ΔK = Knowledge of influence of shaft angle on the clearance between gearbox sump and bottom structure
• D_g = Distance of shaft centre from centreline, X_s. D_a = Determining X_s. I_k = Section at Frame 37. O_k = X_s is 1708	• ΔK = Knowledge of design attribute in relation to another design object. K_r: Derivation T_r: In- situ

28	Alright. The gearbox has been drawn to a cell DVJ971F. Can't remember the name I've put in. I want this and I want that. That's a shame. I've only the plan view and the end view. That's a nuisance. I haven't drawn the view I want. I may just have to do that. So I might have to find my drawing. So I need to find my sub-contractor drawing. I need only to draw the bottom contours of the gearbox. It's only the bottom of the gearbox that is important in the checking of the clearance. Where is my sub-contractor drawing? Over there. 36m [Drawing DVL931-E] Select object 48m
	Insert drawing block The subcontractor has not been able to provide us with computerised drawing so it's necessary to draw this out manually. Ah, certainly there's more ways to provide the information that you need or scale the drawing provided, but you've to start from some where. Some of these I may have to do because having drawn the other view I remember what the dimensions are. Em, Well, DVLJ931-E. DVLJ931-E. Let's do that block again. Slight point. Change the character of this line. That's better. Come on. OK Bingo. *Give me a handle that I can remember. *Leave that at the moment

Comments by designer on his forgetfulness.

- D_g = Drawing of a specific gearbox
- D_a = **Searching** for subcontractor drawing.
- I_k = Library of subcontractor drawings.
- O_k = Drawing located

- K_r : Detailing
- T_r : In- situ

ΔK = Knowledge of key to access CAD database for gearbox drawing

Comments on sub-contractor drawings

29	Right what we've determined is eh that the gearbox is too close to the hull. So, that means that the shaft angle will have to increase. We can't be certain of the eh, the distribution of the bottom structure. But chances are that we have supporting members in the vicinity of the mounting blocks. The distance between the mountings, girth distance of the mounting blocks is 900 mm.
30	<i>Experience tells us</i> with light-weight high speed craft we need to have close spacing of the eh shell in the bottom structure. There is a good chance that the spacing is 300 mm even if it allows to go larger elsewhere.
31	This as it is shown is 900 mm. If we assume 300, so make a line perpendicular to the hull. Move that to a possible location. We measure the distance we have. It is something like 140 mm from the distance of the shell line to the under side of the gearbox. There is a possibility that this stiffener the bottom shell longitudinal might well be 100 mm deep.

• D_g = Sufficient clearance between gearbox and bottom structure. D_a = Determining. I_k = a) Location of gearbox sump and location of bottom structure. O_k = Insufficient clearance • D_g = Sufficient clearance between gearbox and bottom structure. D_a = Decision Making. I_k = Insufficient clearance between gearbox and bottom structure. O_k = Increase shaft angle • D_g = Distance between mounting blocks, X_m. D_a = Determining X_m. I_k = a) Location of gearbox sump and location of bottom structure. O_k = Insufficient clearance	• K_r: Derivation (visually) T_r: In- situ ΔK = Knowledge of the spatial relationship (i.e. clearance) between design objects. • K_r: Derivation T_r: In- situ. ΔK = Knowledge of dependency between design objects and their parameters. • K_r: Derivation T_r: In- situ ΔK = Knowledge of dependency between design objects
• D_g = Close spacing of bottom structure D_a = Defining. I_k = a) Past design: spacing of bottom structure is 300 mm. O_k = Adopt spacing as 300 mm <i>Application of past design knowledge of spacing of bottom structure to current design.</i>	L_g = Spacing of bottom structure, S_b. K_i = Similarity comparison (Sc) I_k = bottom structure of past design of O_k = S_b is 300 mm. T_r = Retrospective. L_{nv} = expedience driven
• D_g = Sufficient clearance between gearbox (GB) and bottom structure. D_a = Determining. I_k = a) Location of gearbox sump and location of bottom structure. O_k = clearance of GB is 140 mm from shell line and the bottom shell longitudinal is 100 mm deep, so clearance is 40 mm.	• K_r: Derivation T_r: In- situ ΔK = Knowledge of spatial relationship between design object (i.e. clearance).

32	And we could we wind up with a situation where based on stiffer position we might be lucky enough to get the gearbox to clear. But the reality is that ships are not built eh to the same high precision that engineering machine shop would work to. There bound to be fluctuation in the form and the nominal clearance of 40 mm won't be considered to be anywhere sufficient.
33	So we have been task to review the shaft angle installation. Increase the angle by sliding from the propeller will have the effect of raising the gearbox and engine.
34	But as can be seen in the profile view, the engine installation height is still quite acceptable Em, there is plenty of room above the engine for the ability to be lifted and withdraw and the gearbox for any withdrawal consideration, to allow for good air circulation and to be able to move services in over and round by it. So, maybe 0.1 degree will give us sufficient clearance. Alright, now. That's ensuring that the GB is clear of the hull.
35	Now we've to look at the engineering considerations of the engine attachments to the gearbox. For the forward machinery space the engine is flexibly coupled to the gearbox direct. The engine in the aft machinery space will require different consideration.

• D_g = Sufficient clearance between gearbox and bottom structure. D_a = Evaluation of clearance based on manufacturing precision of ship's structure. I_k = a) Clearance of 40 mm. O_k = Insufficient clearance. <i>Application of past design knowledge of manufacturing to current design.</i>	L_g = Manufacturing precision. K_r = Dissimilarity comparison (Dc) I_k = a) Ships' manufacturing precision . b) Machine shop precision. O_k = Ship's precision < machine shop precision. T_r = Retrospective. L_{nv} = <i>failure avoidance</i>
• D_g = Review shaft angle installation D_a = Exploring the increase in shaft angle. I_k = a) Increase shaft installation angle. O_k = Raise the gearbox and engine	• K_r: Derivation T_r: In- situ. ΔK = Hypothesis of dependency between design parameters and design objects (i.e. increase shaft angle increase clearance between gearbox and engine).
• D_g = Sufficient headroom for engine installation and removal and good air circulation D_a = Evaluation the effect of the increase in shaft angle. I_k = a) Current position of engine and gearbox based on increased shaft angle. O_k = Sufficient headroom.	• K_r: Derivation T_r: In- situ. ΔK = Knowledge of dependency between design parameter and design objects is confirmed.
• D_g = Engine attachments to the gearbox D_a = Identify the design problem. I_k = a) Engine in forward machinery space. b) Engine in aft machinery space O_k = In forward machinery space, engine is flexibly coupled to gearbox. - In aft machinery space, different type of consideration	• K_r: Association/Disassociation T_r: In-situ/provisional ΔK = Knowledge of interfaces between engine and gearbox.

36	<i>Experience of previous vessels built</i> has enabled us to design an arrangement whereby the propulsion engine is in one engine room and gearbox and the after engine in another watertight space, the centreline engine connected to the input side of the gearbox.
37	<i>Previous vessel we built has a similar configuration</i> , be it a U-drive configuration where the engine is arranged in aft space and the gearbox in the forward space and eh, a small stub shaft was run from the engine flexible coupling through the bulkhead and then stopped at the bulkhead.
38	And then a flexible Cardan shaft, universal drive shaft was installed at the stub end at the bulkhead and at the gearbox. The out of balance forces from the flexible drive shaft caused vibration resonance at the bulkhead which led to localised cracking of some tank structures that are connected to the bulkhead.
39	This eh could have been overcome at the initial design stage had more attention been paid to the stiffness of the structure. But eh, we are confronted with this problem.

• D_g = Engine room configuration D_a = Configuring. I_k = a) past engine configuration. O_k = Engine in forward engine room, gearbox & aft engine in aft engine room, and centreline engine connected to gearbox. <i>Application of past design knowledge on engine configuration to current design.</i>	L_g = engine configuration. K_r = Similarity comparison (Sc) I_k = Source: past engine configuration. O_k = Engine in forward engine room, gearbox & aft engine in aft engine room, and centreline engine connected to gearbox. T_r = Retrospective. L_{nv} = <i>expedience driven</i> in using past arrangement on current vessel.
• D_g = U-drive engine configuration D_a = Configuring. I_k = a) Past design U-drive configuration. O_k = engine in aft space and GB in the forward space and a small stub shaft from the engine flexible coupling through and stop at bulkhead. <i>Application of past design knowledge on engine configuration to current design.</i>	L_g = U-drive engine configuration. K_r = Association I_k = Source: past engine configuration. O_k = engine in aft space and GB in the forward space and a small stub shaft from the engine flexible coupling through and stop at bulkhead. T_r = Retrospective.
<i>Application of past design failure to manufacturing to current design.</i>	L_g = Reason for cracks at tank structure. K_r = Ex I_k = a) Out of balance forces drive shaft caused vibration at bulkhead. b) Tank structure connected to bulkhead. O_k = Resonance vibration at bulkhead transmitted to tank structure. T_r = Retrospective. L_{nv} = <i>failure avoidance</i> of localized cracking
• D_g = Avoid of localized cracking of tank structure connected to bulkhead D_a = Identifying the design problem. I_k = Vibration resonance at bulkhead. O_k = stiffness of bulkhead structure	K_r = Explanation of physical phenomena T_r = retrospective ΔK = Meta knowledge for solving a design problem.

40	But, this is overcome by doing away with the bulkhead bearing and the introduction of a second pedestal bearing on what would be the forward side of the bulkhead replacing the bearing and thus taking away from the flexible drive shaft, from the bulkhead any out of balance forces which is connected to the bottom hull foundation. A pedestal bearing which is originally connected is more able to cope with the forces. <i>This is a lesson learnt and so in this new application we'll be looking to doing the same thing.</i>
41	Now at this moment in time we are not to sure how to configure the structure above on the main deck. We know we have to accommodate some weapon systems. We have to accommodate the various combustible air, compartment air requirements in the ship's machinery spaces.
42	We also need to lift these engines in and out of the vessel so we are driven towards perhaps making this engine room a little bit longer than what is necessary in order to accommodate the items of equipment fitted on the main deck above. So, the shaft that will run from the engine's flexible coupling to the bulkhead is longer than the previous arrangement but compared to this length of the main drive shaft is still comparatively small.

<i>This is an example of learning from design failure and avoiding the design fault in the new design.</i> • D_g = Avoid localized cracking of tank structure connected to bulkhead D_a = Decision Making I_k = Lessons learnt from past design failure(i.e. causes of cracks at tank bulkhead) O_k = No bearing at bulkhead but add 2nd pedestal bearing on forward side of bulkhead.	L_g = Reduce vibration at bulkhead. K_r = Explanation (Ex) I_k = No bearing bulkhead. - Add 2nd bearing on forward side of bulkhead. O_k = Out of balance forces is not transmitted to the drive shaft or bulkhead T_r = Retrospective L_{nv} = <i>failure avoidance</i> of localized cracking.
• D_g = Layout of structure above the main deck. D_a = Identify the design requirement. I_k = Some weapon systems, various combustible air and compartment air requirements in the ship's machinery space. O_k = All the necessary items to consider on the main deck.	K_r: Association T_r: In-situ/provisional ΔK = Knowledge of design requirements to configure deck structure.
• D_g = Need to lift engines in and out of vessel. D_a = Identify the design requirement. I_k = Items of equipment on main deck and engine hatch. O_k = Increase length of engine room. • D_g = Increase length of engine room. D_a = Evaluation of shaft lengths. I_k = Shaft from engine to bulkhead and main drive shaft of past design and main shaft length of current design O_k = Length of shaft from engine to bulkhead < length of main shaft.	L_g = Dimensions of shafts. K_r = Sc I_k = Shafting arrangement of past design O_k = Lengths of main drive shaft and shaft to bulkhead T_r = Retrospective

43	There will be an advantage in mounting the engine aft in some respects. Because with the rise of the buttock towards the after end and the installation angle which happen to be parallel with respect to the baseline on this vessel there will be a converging effect which mean to say that the height of supporting seat will be less and as such proved to be a lighter assembly.
44	So we arbitrary suggest that the small drive shaft in between, again this is dependent on the engine power, we really have to have this engine power in order to make any other judgement. This is the right document. 16V 4000 series rating 2320 kW We will know this will be de-rated to suit ambient air conditions in the theatre of operation. Typically this would be of the order of 2,000 kW, 2000 kW as anticipated. Wow, this is 2720 kW maybe 2,500 kW, about 2,500 kW.
45	Eh, <i>more recent engine installation</i> with 3,000 kW, a drive shaft diameter is 160 mm. This is 2,500 so we just arbitrary put it as a starting point at 150 mm. More detail calculations can follow

• D_g = Lighter assembly of seats. D_a = Qualitative analysis of mounting engine aft. I_k = a) Engine at aft location where angle of shaft installation angle is parallel to base line. b) Rise of buttock lines towards aft end. O_k = Converging effect reduces height of supporting seat.	• K_r: Derivation T_r: In-situ ΔK = Knowledge of the hull form and location of engines and their influence on the height and weight of seats.
• D_g = Suitable size of drive shaft. D_a = Identify design requirement. I_k = Size of drive. O_k = Knowledge of engine power. • D_g = Engine power for sizing drive shaft. D_a = Approximate analysis of engine power I_k = a) Correct engine catalog 16V 4000 series 2320 kW. b) Ambient air conditions in theatre of operation O_k = De-rated power of 2500 kW.	• K_r: explanation based on transmission of power T_r: provisional ΔK = Knowledge of dependency of size of drive shaft and engine power. • K_r: Derivation T_r: In-situ ΔK = Knowledge of influence of ambient air condition on engine power.
• D_g = Sizing drive shaft diameter. D_a = Defining of shaft diameter. I_k = Past design of 3,000kW engine installation has shaft diameter of 160 mm. - Current engine power is 2,500 kW O_k = Drive shaft diameter is set at 150 mm. <i>Application of past design knowledge on design parameter of a design object to current design.</i>	L_g = Dimensions of shafts. K_r = Sc I_k = Past design of shaft installation for engine power of 3,000 kW O_k = Drive shaft diameter is 160 mm T_r = Retrospective L_{tw} = expedience driven by using past design parameter in order to progress the design.

1hr 03m/46	Now, I assume the cardan shaft to be total length of 900 mm. This is not unrealistic. The cardan shaft flexible drive the included angle on this is very shallow. At the moment, it's 2½ degrees. This is very slightly. So let's put something in there. 450. And, No. This going to be 900 [Drafting]
47	Right, so we've a situation where we have a gearbox. We have input coupling from the gearbox and we're going to fit a flexible drive Cardan shaft with the gearbox and a flange on the stub shaft and a pedestal roller bearing, a watertight seal to the shaft to maintain the integrity of the compartment in the engine.
48	Now, if we consider that we're going to have a shaft of about 150mm diameter a corresponding pedestal bearing for that shaft will have to be evaluated to see whether if there is sufficient room allowed for the pedestal bearing
49	Right. Let's just check the flange diameter and existing coupling. That's 200. That's not very promising.

• D_g = Length of Cardan shaft. D_a = Defining of shaft length. I_k = Included angle of Cardan shaft flexible drive is $2\frac{1}{2}$ degrees. O_k = Drawing Cardan shaft length of 900 mm.	• K_r: Detailing T_r: In-situ ΔK = Knowledge of attribute and value of design object
• D_g = Shafting arrangement. D_a = Configuring. I_k = Input coupling from gearbox, fit a flexible drive Cardan shaft with the gearbox, a flange on the stub shaft, a pedestal roller bearing, a watertight seal to the shaft O_k = Shafting arrangement.	• K_r: Association T_r: Retrospective/In-situ ΔK = Knowledge of composition of shafting system and its configuration.
• D_g = matching of pedestal bearing to shaft. D_a = Associating I_k = shaft diameter is 150 mm. O_k = corresponding pedestal bearing to suit and sufficient room for it	• K_r: Association T_r: In-situ/provisional ΔK = Knowledge that the shaft diameter influences the selection of suitable size of pedestal bearing.
• D_g = Compatibility between gearbox coupling and stub shaft. D_a = Evaluation I_k = Stub shaft diameter. O_k = Coupling dia.	• K_r: Association T_r: In-situ ΔK = Knowledge of interfacing design objects (flange diameter to coupling).

50	Typically what I am doing here will be I'll be going to the library and ch looking up the catalogue on Cardan shaft to match torque carrying capacity. So I'm going to grab that document since it is not computerised.
01:00:07/ 51	[Looking for the document] Just so happen in this particular catalogue it doesn't have what I'm looking for. I need to consult yet another catalogue which is in the library.
01:00:09/ 52	[Looking for another catalogue.] Right, what I got to do is to choose one which have the right torque range. Previous experience has driven us to use the 190 Series which have been renamed to 390 Series. This being a heavy duty application. So we go to the 190 Series.
53	And what we need to determine is the TN value which is the torque value. This requires the engine kW rating divided by shaft revolution times a constant factor to give us kNm. So take the torque figure of the engine which was 2,720 at 2,100. $2,720/2,100 \times 9.555$.

• D_g = Catalog on Cardan shaft. D_a = Searching for catalog on Cardan shaft in library. I_k = Catalog on Cardan. O_k = Catalog located.	• K_r: Detailing T_r: In-situ ΔK = Knowledge of specific catalog for Cardan shaft.
• D_g = Catalog on Cardan shaft. D_a = Searching for suitable Cardan shaft I_k = Catalog on Cardan. O_k = No suitable Cardan shaft, so need to consult another catalog.	• K_r: Detailing T_r: In-situ ΔK = Knowledge of specific catalog for Cardan shaft.
• D_g = Suitable Cardan shaft. D_a = Selecting suitable Cardan shaft I_k = Right torque range. O_k = Series 190 in catalog.	• K_r: Detailing T_r: In-situ ΔK = Knowledge of specific series in a catalog for Cardan shaft.
• D_g = Suitable Cardan shaft. D_a = Determining the torque value of Cardan shaft I_k = engine kW/shaft revolution $\times$ constant factor = $2,720/2,100 \times 9.555$ O_k = 12.376 kNm.	• K_r: Detailing T_r: In-situ ΔK = Knowledge of equation to calculate design parameter (i.e. torque).

54	So, we are looking for something which has something in the region of 12.5kNm torque capacity range. Again, <i>experience has shown that something in excess of that capability is typically chosen</i>. The reason eh I think you find for the life cycle of the Cardan shaft. Again we can draw some conclusion from previous installations eh to know where to start from. All these characteristics can be refined at a latter date.
55	All we're trying to do at this stage is to get some approximate idea of space envelope. How much room do we need to allow for the machinery and once we have decided on that we then can move on to other aspects of the arrangement to see from how we can best fit in those needs.
56	So I was looking at arbitrary 900 mm long and it appears quite satisfactory that I can get for the Cardan shaft application which range from 430 mm long up to 1150 mm if required. So my 900 sit comfortably in the middle. And in terms of the torque carrying capacity, it would suggest that eh 190-55 series is quite suitable.
57	So let's just list some basic dimensions for this Cardan shaft. It has certain mating flange sizes which we're looking and that's given in the table as dimension A. This appears to be 250 mm diameter flange. And the flange thickness is given as G which is 18 mm. So we just represent the flanges on there.

D_g = Suitable torque value for Cardan shaft. D_a = Defining a design parameter. I_k = Value of torque value. O_k = Torque = 12.5 kNm. <i>Application of past design knowledge on design parameter of a design object to current design.</i>	I_g = Choice of torque value. K_r = Similarity comparison I_k = Past design practice. O_k = Choose torque in excess of calculation. T_r = Retrospective. Ltw = expedience driven in using value of past design attribute.
D_g = Approximate idea of space envelope. D_a = Prioritising on use of space envelope. I_k = 1) Room needed for machinery. 2) Room for other aspects of arrangement O_k = Optimal allocation of space.	•K_r: Sorting (Not defined yet in original list of knowledge transformers). T_r: In-situ ΔK = Knowledge of problem solving strategy.
• D_g = Suitable Cardan shaft D_a = Searching for suitable Cardan shaft I_k = Catalog on Cardan range from 430 mm to 1150 mm. Required length is 900 mm and calculated torque O_k = 190-55 series in catalog.	•K_r: Detailing T_r: In-situ ΔK = Knowledge of the design attribute of the component(i.e. component number).
• D_g = Mating flange sizes. D_a = Defining. I_k = Table of Cardan shaft dimension. O_k = Flange diameter A = 250; Flange thickness G = 18.	L_g = Knowledge of the design attribute of the component. K_r = Detailing. I_k = Current design case: Flange diameter A = 250; Flange thickness G = 18. O_k = Details of Cardan shaft. T_r: In- situ.

57	So we've a corresponding flange on our stub shaft. It could be a forge stub shaft and if that's the case the flange could be that much thicker. Let's arbitrary put it as 25 mm. That should be comfortable. If it is a forging, the flange will have a root radius, and it could be up to 15 mm radius, maybe a bit more.[This rectangle and rotate 5 and move there].
58	Right, so between these we have the Cardan shaft. What are the principal dimensions and M is 130. This represents the knuckle positions. Let's put that as 130 mm and 130mm.
59	Aah. Hmm. I don't know what you want to say about this. {Just say it.) I've done it wrong. I have chosen the wrong end point on the stub shaft. In fact there would be, instead of basing on shrunk-on diameter it has to be the other way round.
60	I'll have to make a judgement based on the input end to get a clue as to its dimension. So I know that to be 120 mm if my memory from the previous one serves me correct. So at 130. I've that 130 alright which in fact pushes that out there. Now, should be the other way round. Now, there will be a corresponding flange of the Cardan Shaft again. So let's put that one.
61	What we have in terms of distance there? 770 mm but even 770 mm is well within the scope. So I'll need to make a change there. So what we need to do get this [Drawing] This then becomes the knuckle points. Right, that's on the Cardan shaft.

D_g = Dimension of forged stub shaft. D_a = Defining. I_k = Rule for root radius of forged shaft; Mating flange for forged stub shaft. O_k = Flange diameter A = 250; Flange thickness G = 18; Root radius = 25 mm.	L_g = Knowledge of the design attributes of the forged stub shaft. K_r = Detailing. I_k = Flange diameter A = 250; Flange thickness G = 18; Root radius = 25 mm. O_k = Details of forged stub shaft. T_r: In- situ.
D_g = Dimension of Cardan shaft. D_a = Defining. I_k = Table of dimension. O_k = Knuckle positions M = 130 mm	L_g = Knowledge of the design attribute of the Cardan shaft. K_r = Detailing. I_k = Knuckle positions M = 130. O_k = Details of Cardan shaft. T_r: In- situ.
Confession of a mistake made	
D_g = Dimension of Cardan shaft. D_a = Defining. I_k = Table of dimension. O_k = Knuckle positions M = 130 mm	K_r: Detailing T_r: In-situ ΔK = Knowledge of the design attribute of the component.
D_g = Length of Cardan shaft. D_a = Determining. I_k = Table of dimension. O_k = Distance between knuckle positions is 770 mm	K_r: Detailing T_r: In-situ ΔK = Knowledge of the design attribute of the component.

62	Right, what is the next thing I need to determine? Pedestal bearings. Wouldn't it be great if there is only one manufacturer? Dimension to pedestal bearings. [Looking for information.] Found. Cooper bearing. Alright. Cooper pedestal bearing.
63	And what we are considering is a shaft which is nominally 150 mm diameter Here it's in imperial. So we need to look at 150. So we need to look at 150. Cooper roller bearing. It seems to fit 150 diameter. These seem to fit a P10 casing. Fit in P10 casing
64	This pedestral bearing eh which is a fixed type. Don't know the dimension. So I need dimension L. It is 174. 174 is the width of cartridge that carries the roller bearing. So 174 mm.

• D_g = Shafting arrangement. D_a = Prioritising. I_k = List of previous items. O_k = Pedestal bearing. • D_g = Dimension to pedestal bearing. D_a = Searching for pedestal bearing I_k = Catalog on pedestal bearing O_k = Cooper pedestal bearing.	K_r: Detailing T_r: Retrospective/In-situ ΔK = Knowledge of specific catalog to search for Cooper bearing
• D_g = suitable pedestal bearing to match shaft. D_a = Associating I_k = shaft diameter is 150 mm. O_k = corresponding pedestal bearing to suit shaft • D_g = Compatibility between shaft and pedestal bearing. D_a = Searching for Cooper roller bearing I_k = a) Catalog on Cooper bearing. b) shaft diameter = 150 mm O_k = Fit P10 casing, Cooper bearing.	K_r: Association T_r: In-situ ΔK = Knowledge of interfacing pedestal bearing to shaft diameter. ΔK = Knowledge of specific casing for pedestal bearing. K_r: Detailing T_r: In-situ
• D_g = Dimension of Cooper bearing. D_a = Determining dimension L (width of cartridge that carries the roller bearing). I_k = Table of dimension. O_k = L is 174 mm.	K_r: Detailing T_r: In-situ ΔK = Knowledge of design attribute of component (i.e. cartridge of roller bearing).

65	Have a look to see what we have at the bulkhead. Oops. And, presently we've 203 mm into which we'll be looking to put this cartridge. Well, for practical consideration of maintenance accessibility then we must allow the stub shaft to protrude beyond the bulkhead so that there is clearance all round this pedestal bearing for access for maintenance and inspection.
66	We'll probably allow at least 140. So, on that basis with cartridge width 174, half of which, I can't think, 174 divide by 2 is 87. So 87. So, all this time my cardan shaft is going to get shorter and shorter in length. So 87 is the cartridge width. [I've a nasty suspicion, Sim this is a hand fisted way of doing things.]
67	Where is G? 295 mm it is about 147.5 So that's on the cartridge. The pedestal foundation according to the catalogue, its dimension is 89 mm. Hmm interesting.
68	L, L is 174, 178, no, it is correct. H dimension. Dimension H. It's 181. Another dimension is T. T is 415. Put a right angle here. Just drawing in the boundaries of the pedestal. [Drawing]

• D_g = Space around the bulkhead. D_a = Determining clearance to put cartridge). I_k = Measure space available. O_k = 203 mm. • D_g = Clearance for pedestal bearing. D_a = Decision Making I_k = a) Cartridge position near bulkhead. b) Maintenance accessibility. O_k = Allow stub shaft to protrude around the bulkhead.	K_r: Association T_r: In-situ ΔK = Knowledge of rationale for allowing the stub shaft to protrude beyond the bulkhead
• D_g = Clearance for pedestal bearing. D_a = Evaluation I_k = a) Allow 140 mm as distance from bulkhead. b) half cartridge width is 87 mm. O_k = acceptable. Comment by designer	K_r: Association T_r: In-situ ΔK = Knowledge of evaluation criteria for design object (i.e. clearance allowance for pedestal bearing).
• D_g = Dimension of Cooper bearing. D_a = Determining dimension of pedestal bearing. I_k = Table of dimension: dimension G, dimension of pedestal foundation, P. O_k = G is 147.5 mm, P is 89 mm.	K_r: Detailing T_r: In-situ ΔK = Knowledge of attributes of design object.
• D_g = Dimension of Cooper bearing. D_a = Determining dimension of pedestal bearing. I_k = Table of dimension: Dimensions L, H and T. O_k = L is 174 mm, H is 181 mm and T is 415 mm.	K_r: Detailing T_r: In-situ ΔK = Knowledge of attributes of design object.

69	Right. This is pedestal bearing. And now we should legislate for some clearance on the other side of the pedestal bearing.
70	Go back to Cardan Shaft. Just confirm again what those prospects were here for dimension. In Series 740, 780 710 for dimensions. Em, within series 713. So 710. Emm, is the dimension we want to maintain 710. 710.
71	Let's choose the clearance in this cartridge. 25 mm beyond the cartridge. Let's go for 87. $87 + 25 = 112$ say 115. Move that item that point from there to there. That should be sufficient.
72	Right. Now, what've we got left? All this effort is to determine whether we have to move the GB up the shaft line and likewise the engine.
73	So we need to find out what the dimension are from the knuckle joints of flexible Cardan shaft. So it is 411, $411 + 260 = 671$.
74	So the dimension I don't want to fall below is 710.

• D_g = Clearance for pedestal bearing. D_a = Planning the next design action. I_k = Clearance for one side is acceptable. O_k = allow for some clearance on the other side of the pedestal bearing.	K_r: Detailing T_r: In-situ ΔK = Knowledge of design attribute of component.
• D_g = Dimension of knuckle joints of Cardan shaft, K. D_a = Determining K I_k = Series 713 O_k = K is 710 mm	K_r: Detailing T_r: In-situ ΔK = Knowledge of design attribute of component.
• D_g = Clearance around cartridge. D_a = Determining clearance to put cartridge. I_k = a) Cartridge width is 87 mm. b) Clearance is set at 25 mm. O_k = Total clearance is $87 + 25 = 112$ say 115 mm.	K_r: Derivation T_r: In-situ ΔK = Knowledge of spatial attribute associated with design object.
Restating a design goal	
• D_g = Dimension of knuckle joints of Cardan shaft, K. D_a = Determining K I_k = $411 + 260$ O_k = K is 671 mm by measurement.	K_r: Detailing T_r: In-situ ΔK = Knowledge of design attribute of component.
• D_g = Dimension of knuckle joints of Cardan shaft, K. D_a = Defining K I_k = Measured 671 mm but must not be below 710 mm O_k = K is 710 mm	K_r: Detailing T_r: In-situ ΔK = Knowledge of design attribute of component.

74	So I need to move the engine and the gearbox at least 40 mm up the shaft line.
75	Now it may just suit me to move it more than that figure. So I look at the gearbox sump to the frame line. The forward end of gearbox sump is some 92 mm, further aft from the frame line. In order to be able to fit a reasonably deep frame, in order not to cut away the gearbox sump, it may sense to move engine and the gearbox up the shaft line so that it will clear this frame.
76	We've to be careful that in the process of moving we do not move it through the bulkhead at the other end. That is some 92 mm in the figure. Move engine and gearbox 120 mm.
77	So that will add 120 mm to the length of the Cardan shaft. So let's then do that. So we need to extend this line. So 120. Extend there to there. Move gearbox and engine. Move from there to there.

• D_g = Optimal location of the engines. D_a = Decision making I_k = Measured K is 671 mm but desired is 710 O_k = Move engine and gearbox 40 mm up the shaft line.	K_r: Derivation T_r: In-situ ΔK = Knowledge of the design rationale for moving engine and gearbox up the shaft line.
• D_g = Clearance of gearbox (GB) of gearbox sump from frame. D_a = Exploring I_k = Current location of gearbox sump. O_k = Need distance of gearbox from frame line. • D_g = Clearance of GB sump from frame. D_a = Determining I_k = Distance of gearbox from frame line? O_k = Distance of gearbox from frame line is 92 mm. • D_g = Clearance of gearbox of gearbox sump from frame. D_a = Exploring I_k = Current location of gearbox sump. O_k = Need to move engine & gearbox up the shaft line so as to clear the deep frame.	K_r: Derivation T_r: In-situ ΔK = Knowledge of the spatial relationship between design objects.
• D_g = Move engine and gearbox (GB) up the shaft line. D_a = Exploring I_k = a) Distance of GB from frame is 92 mm. b) Need to move engine & GB up the shaft line so as to clear the deep frame. O_k = Move engine & GB 120 mm up the shaft line.	K_r: Derivation T_r: In-situ ΔK = Knowledge of the design rationale for moving engine and gearbox up the shaft line.
• D_g = Dimension of knuckle joint of Cardan shaft. D_a = Defining I_k = Increase length of Cardan shaft by 120 mm O_k = Let $K = 710 + 120$ mm.	K_r: Derivation/association T_r: In-situ ΔK = Knowledge of influence of design decision on associated design object.

01:38:00 /78	Now, what haven't we done. I'll do it again. Only this time, we've put the elements of the Cardan shaft. Oops, why isn't that happening? Right the assumed bulkhead position, the various connection can come off the engine which make clearance quite tight. Check what clearance we got available. 347 which not accessible by a man.
79	So what shall we do? Instead of moving the bulkhead. Maybe necessary to move the whole configuration further aft rather than further forward. So just leave that for the time being.
80	I think we can feel reasonably confident with the arrangement as it stands now. Just show the other pedestal bearing. Em, using the frame structure to back it up. Whenever possible trying to position supporting, shaft supporting equipment close to frame structure knowing that down the line we have good, good routing for individual equipment.
81 01:43:00/ 82	Alright! Now, one thing that's happens in moving the shaft or the gearbox and the engine further up the existing shaft line has been to increase clearance between the gearbox sump and the bottom of the boat and we should go back to check to see if it's an the acceptable height. Still the same point there. We've gain a little but not a lot. So in reality we still have to make some minor changes to the angle of the shaft to ensure clearance at the gearbox sump. So no great gains there.

• D_g = Clearance around the engine. D_a = Determining I_k = Clearance of engine from bulkhead? O_k = Clearance is 347 mm.	K_r: Derivation T_r: In-situ ΔK = Knowledge of the spatial relationship between engine and bulkhead.
Exploring but without any action made.	
• D_g = Integrating shafting system to structural system. D_a = Synthesis. I_k = Support of pedestal bearing. O_k = Layout showing integration of shaft supporting equipment to frame structure.	K_r: Association T_r: In-situ ΔK = Knowledge of integration between physical building blocks (i.e. shafting system and structural system).
• D_g = Sufficient clearance between GB & bottom. D_a = Evaluating I_k = a) Location of gearbox sump b) Location of bottom structure. O_k = Clearance is acceptable.	ΔK = Knowledge of the spatial relationship between gearbox sump and bottom structure K_r: Derivation T_r: In-situ

83	Now what do we do? Em, we still have the opportunity to put generators in the aft position rather than the forward position.
84	It's a noise issue here. Generators are likely to be kept running. The operating profile of this vessel is such that it has limited time at sea. It has to spend a lot of time in the harbour. Chances are the harbour facilities may not have the shore supplies. It will be necessary to run generators. The crew is likely to live on board this vessel. This means to say that the generators will be run all the time which put heavy burden on noise control.
85	And to have the generators directly next to the accommodation space there is a high risk of not meeting eh, acoustic performance profiles. So subject to the weight balance of the vessel, we're might consider putting the generators aft of the main propulsors keeping all the noise source further aft.
86	Notwithstanding that means that if the accommodation move further aft that will come up against the engine which will be a greater noise source but perhaps of intermittent nature. Maybe something we can live with. We put these to see how the rest of the layout turns out.

• D_g = Locations of generators. D_a = Exploring I_k = Forward or aft position in engine room. O_k = Aft position.	K_r: Derivation T_r: In-situ ΔK = Knowledge of problem structuring (i.e. generators in forward or aft locations).
• D_g = Optimal position of generators. D_a = Identifying design issue – noise issue I_k = a) Generators are kept running b) Vessel & crew spend more time in the harbour than at sea. c) Less likelihood of shore supplies at the harbour facilities. O_k = Necessary to run generators and the need for noise control.	K_r: explanation T_r: retrospective/In-situ ΔK = Knowledge of design factors influencing location of design objects.
• D_g = To satisfy acoustic performance profile. D_a = Exploring I_k = a) Generators next to accommodation. b) Generators aft of main engines. c) Weight balance of the vessel O_k = Consider putting generators aft of main engines.	K_r: Explanation T_r: retrospective/In-situ ΔK = Knowledge of location of design objects on the performance of the design.
• D_g = Optimal locations for accommodation. D_a = Exploring I_k = a) Accommodation move further aft near main engine, a greater noise source but of intermittent nature. O_k = Evaluate layout to decide.	K_r: Explanation (from the knowledge of noise propagation) T_r: retrospective/In-situ ΔK = Knowledge of the influence of noise source on accommodation K_r: Explanation T_r: retrospective/In-situ ΔK = Knowledge of qualitative criteria

87	So, taking advantage of <i>previous vessel arrangement</i>, we just get some idea of the crew numbers we may need to accommodate. Take a look at what we've got at the moment. 2, 4, 6, 8, 10 within bulkhead space. It's quite crowded.
88	We've to move to accommodate seven officers. This could be more reasonably arranged, more comfortably arranged for 7 officers. It's in the mid-body of ship which from a comfort point of view, it'll be more acceptable. There is a tendency to structure comfort with rank of the enlisted personnel.
89	What else we need to accommodate? We got 28 crew possibly a split balance between senior rating and junior ratings. So 28, maybe 12 seniors rates and 16 junior rates.
90	Ah, other documents will have to be look at to see whether there's a more definitive requirement. Crew numbers. Crew makeup. Right, we got 36 crew. 8 officers including Commanding Officer, 1 Chief Officer and 6 others, 10 Chief Petty Officers 18 listings. Lots of these documents.

• D_g = Space for crew accommodation. D_a = Determining crew numbers I_k = a) crew accommodation arrangement of past design O_k = 10 crew berthing spaces between bulkheads. • D_g = Space for crew accommodation. D_a = Evaluation I_k = a) 10 crew berthing spaces between bulkheads. O_k = Quite crowded.	K_r: Similar comparison T_r: retrospective ΔK = Knowledge of space requirement for acceptable accommodation. L_{tw} = <i>expedience driven</i> by using value of past design attribute. K_r: Derivation T_r: In-situ
• D_g = Space for officer accommodation. D_a = Identifying need for officer accommodation. I_k = a) Past arrangement of officer accommodation. b) Greater comfort for officer O_k = Officer accommodation in mid-body of ship. • D_g = Space for officer accommodation. D_a = Evaluation I_k = a) Officer accommodation in mid-body of ship. O_k = More acceptable.	K_r: Similar comparison T_r: retrospective ΔK = Knowledge of relating comfort with rank of personnel. K_r: Derivation T_r: In-situ
• D_g = Space for crew accommodation. D_a = Identifying need for crew accommodation. I_k = a) 28 crew. O_k = 12 senior rates and 16 junior rates.	K_r: Detailing T_r: In-situ ΔK = Knowledge of specific accommodation for various ranking of crew.
• D_g = Definitive requirements for crew accommodation. D_a = Identifying definitive number of crew. I_k = a) Documents. O_k = 36 crew: 1 Commanding Officer, 1 Chief Officer, 6 Officers, 10 Chief Petty Officers, 18 listings.	ΔK = Knowledge of crew make-up. K_r: Detailing T_r: In-situ

91	What do we have within this block. We say we got 8, 10, 12 16, 20. Emm, show some promise. [Block shows officer accommodation].
1:51:00/ 92	I think we should determine where roughly we're going to start to put the propulsion engine. We should now look to see what the other limitations are for the internal arrangement of vessel.
93	So we need say for a nominal say displacement of 550 tonnes, we <i>know from the past vessels</i> that we've built, we've a set of hydrostatic we can determine the design waterline would be.
94	As the vessel has been designed to ABS, the requirements of high-speed rules, Lloyd's Register Special Service Craft. Eh, there are regulations within those documents to dictate positions of collision bulkhead. Once we've look at those rules and determine where these bulkheads should be. That would give the other extreme of the envelope that we have to use for this vessel.

• D_g = Space for officer accommodation. D_a = Evaluation I_k = a) Space for 18 officers O_k = Space shows some promise.	K_r: Detailing T_r: In-situ ΔK = Knowledge of space required to accommodate officers
• D_g = General arrangement of vessel. D_a = Identifying design issue I_k = Location of propulsion engine. O_k = Investigate the limitations for the internal arrangement of vessel.	K_r: Association T_r: In-situ ΔK = Knowledge of the interaction between propulsion engine and internal arrangement of vessel.
• D_g = Design waterline. D_a = Determining design parameter I_k = a) A set of hydrostatics b) displacement of 550 tonnes O_k = Design waterline?	T_r: retrospective K_r: Similar comparison (past design knowledge used in activity of determining). ΔK = Knowledge learned from past design (i.e. hydrostatic information). L_{nv} = <i>expedience driven</i> based on knowledge of design procedure to derive design parameter.
D_g = Positions of collision bulkhead. D_a = Defining. I_k = Lloyd's Register Special Service Craft. O_k = position of collision bulkheads. D_g = Envelope of space for use. D_a = Determining. I_k = Position of collision bulkheads O_k = Envelope of space to use for this vessel.	L_g = Knowledge of influence of regulation on collision bulkhead and on space envelope. K_r: Derivation. I_k = position of collision bulkheads based on Lloyd's Register Special Service Craft. O_k = a design case of positions of collision bulkhead for a particular vessel. T_r: In-situ. ΔK = Knowledge of appropriate requirement (regulation) to satisfy.

95	Consultation of hydrostatics of previous vessel which I happen to have at hand. Right, nominally, a draft of 2.7 m will correspond to a displacement of approximately 550 tonnes. That will give us an indication. A draft, a design draft, something of a design draft of 2.7 m is appropriate.
96	Right now the requirement of the rules with regard to collision bulkhead is that the bulkhead be placed not less than 94 % of ship's length and not greater than 97 %. Something of that nature.
	I need to check, as I can't remember. So more document consultation. Well something you remember. But the likes of this work is such that it doesn't come sufficient enough to stick all in the head.
97	Collision bulkhead in table 2.42. Consideration will be given to the proposal for positioning the bulkhead slightly further aft of the arrangement. The craft, blur, blur,.....The minimum is 0.05 length on waterline. The maximum is 0.08. And as we're going as much as practical in this boat, always squeezing for more space. So we'll kick off at 0.05 and see where that sits in relation to transverse framing eh as arranged in the previous vessel.
98	Right, what we need to determine now is based on 2.7 m draft. We need to see what this represents in terms of waterline length. This corresponds to $55.4 \text{ m} \times 0.05$. This means to say the collision bulkhead from the stem needs to be arranged a minimum of 2.77m aft of the intersection of waterline with the stem.

D_g = Design draft. D_a = Determining. I_k = a) A set of hydrostatics b) displacement of 550 tonnes O_k = Design draft is 2.7 m.	K_r: Derivation T_r: In-situ ΔK = Knowledge of procedure to determine the design attribute of vessel.
D_g = Positions of collision bulkhead. D_a = Information gathering. I_k = Requirements of Lloyd's rule for Special Service Craft. O_k = Not less than 94% ship's length or not greater than 97%.	K_r: Detailing T_r: provisional ΔK = Knowledge of the requirement of rule for the location of collision bulkheads.
Comments by designer for not remembering certain things	
D_g = Positions of collision bulkheads. D_a = Information gathering. I_k = Collision bulkhead in Table 2.42. O_k = Minimum is 0.05 length on waterline (LWL), Maximum is 0.08 LWL D_g = Positions of collision bulkheads. D_a = Decision-making. I_k = Minimum is 0.05 LWL, maximum is 0.08 LWL O_k = Set at 0.05 LWL to get more space.	K_r: Detailing T_r: in-situ ΔK = Knowledge of the requirement of rule for the location of collision bulkheads. K_r: Detailing T_r: retrospective/in-situ/provisional ΔK = Knowledge of reason for choosing the lower value is to allow for more space.
D_g = Position of forward collision bulkhead from stem. D_a = Determining. I_k = At, design draft is 2.7 m and LWL is 55.4 m O_k = Distance from stem is $0.05 \times 55.4 = 2.77\text{m}$.	ΔK = Knowledge of the requirement of rule for the location of collision bulkheads. K_r: Detailing T_r: In-situ

99	So just see where 2.77m comes to. 2.77 and it seems to sit quite reasonably close to what we've before. This is 31. So let's take 32 and the distance between is 1,000 mm. So that gives us the clue where the frame is. So got 28. It's true. Isn't it. That's 15. So 1, 2, 4, 5, 6, 7, 8, 9, 10.
99	Let's put in the <i>previous craft</i> bulkhead at the moment so that I know where I am. And that's 16, 17, 18, 19, 20, 21, 22. That's correct. So that's where the <i>previous craft</i> had the bulkhead. Obviously, there is some advantage if we can retain some of these positions. That'll help out in thinking as we <i>know</i> what space we got. So, that'll be eh, the FP bulkhead position of the <i>previous craft</i> . That looks like 0, 1, 2, 3,...
100	That's forepeak bulkhead. That's the demand of the rule requirement.
101	At best this FP bulkhead could move $\frac{1}{2}$ frame as we're likely to have half frames in the fore end of the boat. Because it's getting so fine and we can't practically run longitudinals and won't be able to add any more structure than what we have there in any case because of the shape of the vessel. So there is a possibility that the FP bulkhead could go forward by half frame.
102	Now part of the requirement is what we can do is we can draw in from <i>previous vessel</i> the arrangement and see how best that suits this new configuration. We can always edit out what we don't need.

D_g = forward collision bulkhead with respect to frame line. D_a = Determining. I_k = 2.77 m from the stem of vessel O_k = Reasonably close to Frame 32.	
See next segment <i>Application of knowledge learnt of location of past design objects on current design.</i>	L_g = Knowledge learned from past design of location of FP bulkhead. K_r: Derivation. I_k = position of collision bulkheads of previous craft O_k = a design case of positions of FP bulkhead for current vessel. T_r: retrospective. Ltw = expedience driven in applying location of past design object.
D_g = Positions of forepeak (FP) bulkhead. D_a = Defining. I_k = location of FP bulkhead of previous craft. O_k = position of FP bulkhead.	
D_g = Optimal position of FP bulkhead. D_a = Exploring. I_k = Half frame transverse framing due to shape of vessel. O_k = possibility of moving FP bulkhead by half frame space.	K_r: Explanation T_r: In-situ ΔK = Knowledge of rationale for moving FP bulkhead half a frame space as the vessel is stiffened at half frame interval.
D_g = Positions of forepeak (FP) bulkhead. D_a = Evaluation. I_k = Location of FP at half frame. O_k = Position of FP bulkhead evaluated against previous vessel.	K_r: Explanation T_r: In-situ ΔK = Knowledge of rationale for moving FP bulkhead half a frame space.

	OK. So what we've done is to import the accommodation characteristics from previous vessel. And, now we've to consider the features required for the new vessel.
103	Now, we know we've a sophisticated gun to the main deck. Dear, oh dear. This vessel is a warship. It's got to be capable to deliver the weapon systems. [Extract some info.] What is known as hard kill factors? {Check document}
104	Alright. Point defence missile system. Nominated is RAM, R A M, Rolling Airframe Short range surface to air missile system. In short, RAM comprising of Mark 49 Guided Missile Launching System, Mark 42. Block One Guided Missile RAM Pack., 11 RAM version.
105	Surface to surface missile system. Harpoon. 8 Harpoon Missiles. Block 1 G Missile with Grade B shock resilient canister.
106	Medium calibre gun system chosen. Otto Melara 76/62 super rapid CIWS (Close in Weapon System). Phalanx. Block 1 Bay.

Comment on the milestone reached thus far.	
D_g = Vessel is warship. D_a = Identifying design requirement. I_k = Design specification. O_k = Must be capable to deliver weapon systems, hard kill factors.	K_r: specialisation to hard kill factors T_r: retrospective/in-situ ΔK = Knowledge of the weapons requirement for warship.
D_g = Vessel is warship. D_a = Identifying design requirement. I_k = Design specification - Point defense missile system. O_k = RAM comprising of Mark 49 Guided Missile Launching System, Mark 42.Block One Guided Missile RAM Pack.	K_r: decomposition of missile system T_r: retrospective/in-situ ΔK = Knowledge of the missile system requirement for warship.
D_g = Vessel is warship. D_a = Identifying design requirement. I_k = Design specification - Surface to surface missile system. O_k = 8 harpoons missiles & grade B shock resistant canister.	K_r: decomposition of SS missile system T_r: retrospective/in-situ/provisional ΔK = Knowledge of the surface to surface missile system requirement for warship.
D_g = Vessel is warship. D_a = Identifying design requirement. I_k = Design specification – Medium calibre gun system O_k = Otto Melara 76/62. CIWS, Phalanx, Block 1 Bay.	K_r: specialisation to Otto Melara & Phalanx T_r: in-situ ΔK = Knowledge of the medium calibre guns requirement for warship.

107	Minor calibre guns - 2 off M60, GPMG M60. M60, M60, M60. Not to sure is that a 0.5" or .62" calibre. And that comes as requirement for small arms which includes those 2 machine guns and 16 assault rifles and 8 automatic pistols. They are not so much of a worry space wise. So those are the large items that have got to be considered.
108	Soft kill consideration which will take up a bit of space. And that's decoy launcher and anti-missile decoy launcher. And, ah, Super barricade 4 launchers.
	All these need their space envelope both for installation and for their effective use.
109	The initial problem seems the point defence system and Close-in Weapons system chosen are very similar in their roles. And each requires a large area of target acquisition. And as one of the other requirements of the vessel, it should be, its visual should be kept small so that its optical reflection are limited. One does not want to build superstructure and tiers of superstructure in an effort to mount all these system at different levels so that they don't interfere with each other. So somewhere along the line there's got to be a compromise. So, what gonna give. I don't know.
110	First of all let's see about putting Otto Melara Gun.

D_g = Vessel is warship. D_a = Identifying design requirement. I_k = Design specification – Minor calibre guns. O_k = GPMG M60, 16 assault rifles, 8 automatic pistols.	ΔK = Knowledge of the minor calibre guns requirement for warship. K_r: specialisation to GPMG, assault rifles and pistols T_r: in-situ
D_g = Vessel is warship – soft kill factors. D_a = Identifying design requirement. I_k = Design specification – soft kill factors. O_k = Decoy launcher & anti-missile decoy.	ΔK = Knowledge of the soft kill consideration requirement for warship. K_r: specialisation T_r: in-situ.
Comment	
D_g = Vessel is warship – effective defense role. D_a = Identifying design requirement. I_k = Point defense system & CIWS. O_k = Each requires a large area of target acquisition. D_g = Vessel is warship – effective defense role. D_a = Identifying design requirement. I_k = Point defense system & CIWS. O_k = a) Low optical reflection by keeping its visual small and the need to restricting size of superstructure.	K_r: Explanation T_r: In-situ/ retrospective ΔK = Knowledge of influence of specific design requirements of the weapon systems on the overall superstructure design of the vessel. (small superstructure is desirable for low optical signature and large target acquisition area).
D_g = Vessel is warship. D_a = Prioritising design tasks. I_k = Lists of weapons requirements. O_k = Otto Melara Gun.	

111	So search for library cells for that Gun Otto 76 [Searching the database]
112	One thing that is <i>known about</i> this particular gun is that the under-deck carousel which carries the ammunition is such that there's a need for clearance in order to load ammunition on to it. And a minimum of 4 m in terms of compartment would be required as we know from the <i>experience of previous installation</i>. Right. Based on existing bulkhead positions, the forward most compartment was 5m long. So that would appear to be more than sufficient.
113	Unfortunately that would place the gun quite a long way forward which from a ship-motion point of view would not be quite clever. Not that its' a man operated gun but it still does not want to be in a severe environment. These guns are gyro stabilised and they do not want to be constantly engaged in correcting for trim variation just to maintain a target trajectory.
114	So if the gun is placed a little bit further aft it might improve matters. But the further aft it comes it comes to the superstructure. The superstructure will have the effect of restricting the training arc of the gun. Thus limiting the target acquisition area. So again we have to seek a compromise, we got to look at everything that is going on in the boat and see what's good at the end of the day to move things around.

D_g = Vessel is warship. D_a = Searching design tasks. I_k = Databases of library cells of guns. O_k = Otto Melara Gun.	K_i: Detailing T_r: In-situ/ retrospective/ provisional ΔK = Knowledge of details of the gun from catalog.
•D_g = Sufficient clearance to load the ammunition. D_a = Evaluation. I_k = Past design: Minimum of 4 m compartment. Forward compartment is 5m. O_k = Sufficient clearance. <i>Application of past knowledge of space requirement of under-deck carousel on current design.</i>	•L_g = Design requirement for loading of ammunition. K_i: Association. I_k = Otto Melara Gun O_k = Under-deck carousel with good clearance. T_r: retrospective L_{tw} = conflict avoidance. •L_g = Under-deck carousel with good clearance. K_i: Sc. I_k = Past design of under-deck carousel of Otto Melara Gun O_k = Minimum of 4 m compartment. T_r: retrospective
•D_g = Optimal position of gun. D_a = Evaluation. I_k = Forward location of gun. O_k = High ship motion at the forward end. Constant correction of gun gyro to allow for trim variation.	K_i: Explanation T_r: In-situ ΔK = Knowledge of reason why gun cannot be to placed to far forward).
•D_g = Optimal position of gun. D_a = Exploring location of gun. I_k = Move gun further aft. O_k = Too near the superstructure restricts the training arc of the gun and the target acquisition area.	K_i: Explanation T_r: In-situ ΔK = Knowledge of problem structuring in terms of the influence of the superstructure on the training arc of the gun

115	It's may be better for us to consider what we do now is going to have an effect on accommodation and how best to organise this accommodation. Let's just see what else have we got. What we'll do is introduce another layout to this arrangement to include the another deck. [Key strokes].
02:14:00/ 116	Right, now Harpoon. Let's get the missile, the Harpoon Missile Plan. We know that the height of the missile installation is such that it's possible to walk under the level of the lowest canisters. So we can arrange to maintain a ship's side walkway to get around the missile installation. So we just nominally barricade 900 mm on the ship's side. [Key strokes] Oh, why am I not getting the offset? Why I'm not getting the offset? We'll try the copy command then. Just a nominal ...Oops. Just the nominal starting position.
117	So now we got to include.... I have to have 8 missiles. This is 4. So, it is customary to have missiles pointing in opposite directions, so the other bank of four facing the opposite direction. At least when they don't have to when they take off, go round the houses and come all the way back again. Half will go one way, the other half will go the other way.

D_g = Optimal position of gun. D_a = Exploring location of gun. I_k = Location of gun. O_k = Effect on how best to organise accommodation.	K_r: Explanation T_r: In-situ ΔK = Knowledge of problem structuring in terms location of the gun and the below deck accommodation.
D_g = Location of Harpoon. D_a = Evaluation. I_k = Harpoon Missile Plan. O_k = Possible to walk under the lowest canister. D_g = Location of Harpoon. D_a = Decision Making. I_k = Possible to walk under the lowest canister. O_k = Arrange ship's side walkway around installation. D_g = Ship's side walkway around missile installation. D_a = Defining. I_k = Location of missile and surrounding deck area. O_k = ship's side walkway of 900 mm.	K_r: Explanation T_r: In-situ. ΔK = Knowledge of the design of attribute on the operational requirement of the vessel. K_r: Association T_r: In-situ. ΔK = Knowledge of the relationship between missile and the walkway. L_g = Knowledge of size of walkway. K_r: Derivation. I_k = Current design case: location of missile and deck space around missile O_k = Design case: location of missile and location of walk way. T_r: In- situ.
D_g = Ship's side walkway around missile installation. D_a = Defining. I_k = Location of missile and surrounding deck area. O_k = ship's side walkway of 900 mm.	L_g = Knowledge of the arrangement of missiles. K_r: Derivation. I_k = Current design case: locations of bank of missile O_k = Design case: location of one bank of missile in one direction and the other in opposite direction. T_r: In-situ.

118	Oops. Em. Let do a rotate and copy. I did, come on. Rotate. Em, because I've got the object snap on it went to the other end of the line to rotate. It didn't accept mirror. OK that'll just get some where on the boat. I've to accommodate them somewhere.
119	This is not the best position to site eh the missile <i>because</i> there is the requirement for the engine removal hatches which means to say the alignment that maybe arranged for the missiles will be disturbed every time an engine was withdrawn. What's more <i>because</i> the engine removal hatch is likely to be bolted, the rigidity of the connection to the main deck is in question.
120	As the engine removal hatch saddles the full width of the ship it doesn't seem any clear way that we can avoid not putting these missiles on the bulkhead section unless they are placed somewhere else on the ship which may well have to be the consideration.
121	Let's move one bank of missiles forward to avoid the generators just out of curiosity. It will be alright there. It would be a lot smaller to remove the generators.

Drawing commands	
•D_g = Location of Harpoon. D_a = Evaluation. I_k = a) missiles in way of engine hatch. b) bolted hatch less rigid. O_k = Not the best position to site missile.	K_r: Derivation/explanation T_r: In-situ. ΔK = Knowledge of factors in evaluating location of missiles.
•D_g = Location of missiles. D_a = Constraining location of missiles. I_k = a) engine hatch saddles full width of ship. O_k = Missiles on bulkhead section	K_r: Derivation/explanation T_r: In-situ. ΔK = Knowledge of constraint on siting of missiles.
•D_g = Location of missiles. D_a = Decision Making on location of missiles. I_k = a) Generators in the way. b) Generators are smaller to remove. O_k = Move one bank of missiles forward to avoid the generators.	ΔK = Knowledge of alternative location for missile in a less restrictive location. K_r: Explanation T_r: In-situ.

122	We could find ourselves with keeping slightly more positive foundations for missiles in way of the smaller bolted hatches than what we do have for the machinery hatches. But as it stands at the moment because of the extent of the generator space we can only accommodate one bank of missiles.
123	There is another thought that we can organise the missiles to be arranged at the forward end of the superstructure. But, this will have the effect of raising or the need to raise the height of the superstructure which again will compromise the optical signature of the vessel. Taking hold of the missiles, let's find out what we can do here.
124	See whether we got them on file. If we haven't got a problem. CIWS – CIWS Goalkeeper, CIWS Phalanx.
125	That engine got to go somewhere and that's where the bulkhead is. I don't want that bulkhead any more. That could be where the superstructure is. Doesn't look as though it can go. Let's try and move this engine and see what comes out of it. Let's grab hold of it down there. It looks as if the hatch is around the bulkhead. We could get pass some fixed structure up to that point. OK.

•D_g = Location of Harpoon. D_a = Evaluation. I_k = a) Missile in way of machinery hatch or smaller hatch. O_k = Better foundation with smaller hatch. •D_g = Location of missiles. D_a = Decision Making on location of missiles. I_k = Extent of generator space/hatch. O_k = Arrange one bank of missiles on smaller hatch.	K_r: Derivation/explanation T_r: In-situ. ΔK = Knowledge of factors in evaluating location of missiles. K_r: Explanation T_r: In-situ. ΔK = Knowledge of rationale for locating one bank of missiles over generator hatch.
•D_g = Location of missiles. D_a = Exploring. I_k = Forward end of superstructure. O_k = Raise height of superstructure and optical signature.	K_r: Derivation T_r: In-situ. ΔK = Knowledge of factors in locating the missiles at the forward end of superstructure.
Retrieving information on CIWS	
•D_g = Location of engines. D_a = Exploring. I_k = Possible locations of engine if bulkhead is moved. O_k = Move engine and evaluate.	K_r: Derivation T_r: In-situ. ΔK = Knowledge of constraints in locating the engine.

Hopefully I got it here. RAM RAM RAM. I don't know what it's been called. [Locate rolling airframe frame missile.]

Oops it may not be in my library. I got it in my library. Decoy, Harpoon, Exocet, RAM Rolling Air Frame. Em Cancel, cancel. What's else we've got. See whether we can find this other. Looks as though it has been done but not committed to this as yet. What's that gonna be? Oh it's not there! Interesting! Where did it go? 325, 327 F1 or F2 E1 or E2. Let's try E1.

128

So relying on *previous vessel's eh design* where same equipment have been used. So just going into that vessel and I am going to extract a drawing, also I can see on this particular arrangement as to whether there are other considerations towards other associated electronic equipment that may be installed below the mounting. So, see to do that.

129

Let's draw a line from let's take it from em. Let's take a line from there to there. Copy that cell. Object snap. Intersection, intersection there to that intersection there.

130

From. It looks as though there are some space allocation below the mounting for associated electronics. So just need to make a note of the likely area. It's nominally of 3m square envelope required below the mounting. Let's just check in the schedule to see whether compartment 91 CWS equipment room. It's the same. This is an optional fit. It's either CWS or Rolling Air Frame.

ΔK = Knowledge of the database to search for design object (i.e. RAM). I_1 = Library on missiles: Decoy, Harpoon, Brauet, RAM Rolling Air Frame O_1 = RAM found	
• D_1 = Location of Rolling Air Frame (RAM). D_1 = Identifying design requirement for RAM. I_1 = RAM O_1 = Design consideration of associated electronic equipment	L_1 = Knowledge of the arrangement of missiles. K_1: Sc. I_1 = Past design installation of RAM O_1 = Location applied to current design T_1 In-situ. Knowledge of the spatial relationship between RAM and other electronic equipment K_1 Association T_1 In-situ I_{1m} = conflict avoidance
Drawing commands:	
D_1 = size of the space for associated electronics D_1 = Associating I_1 = RAM or CTWS and the space below mounting. O_1 = 3 m square envelope required Application of past design knowledge of design object recorded in database	K_1: Association/detailing T_1: retrospective by storing information in system. ΔK = Knowledge of the space envelope required for associated electronics for RAM missile. I_{1m} = experience driven in using information on design object stored in database.

131	That's where our experience comes in. From other clients they either select CIWS or Rolling Air Missile System and not necessary both. This is where we may have trouble.
	[It is very interesting in this arrangement that we got a few design features that may well benefit us in the new vessel proposal. So what we can do is actually copy a fair proportion of these details into the other drawing so that we'll have them available to us. No point in re-inventing the wheel.
02:31:00/ 132	It's there. It's a different chaff system. That's the layout of lower deck carousale. Hmm. Drag. {Keystrokes}. I just pull in from another ship. Just go back to that ship again. Just check.. Em. What's that are being pulled out. Layer CIWS. It's a LCMK9P, LCMK9P. LCMK9P. Don't know. Don't know where the name comes from. Let's get into my library and find out. LCMK9. Well, there it is. After all that.
133	Now, that's all the bits and pieces with it. Loading working space. So in order to be able to load missile into vessel you need this space as well.

Example of knowledge of past designs not sufficient to apply to current design	
Copy details of other drawings. Documenting	
•D_g = Location of CIWS D_a = Searching. I_k = From another ship, layout of lower deck carousale. O_k = Layer CIWS – LCMK9P.	K_r: Detailing T_r: In-situ. ΔK = Knowledge of the database to search for design object (i.e. LCMK9P).
D_g = Space for loading missile. D_a = Associating I_k = Layer CIWS – LCMK9P. O_k = Working space of CIWS.	K_r: Association T_r: In-situ. ΔK = Knowledge of the spatial relationship between missile and the missile loading bay.

134	Ops. I better go look at the plan view as well and see what that says. CIWS. CIWS. CIWS. Phalanx. [Keystrokes]. But we <i>know</i> that the magazine can be arranged under-deck. So we could pull that down at that level where it's slightly there.. Hmm, maybe this will go higher to see if we can accommodate it.
135	So what else have we got? Minor calibre guns that can be organised in the sides. Small guns. Medium calibre guns. Surface to surface we got. Point defence system we got. Decoy launchers. Probably not too bad.
136	Do we have enough accommodation? Chances are they don't. We need a very large operation room. Do we have something in the making? We know the existing vessels in the client's fleet and the way that that vessel is configured could help us. The superstructure is extremely far forward which we wouldn't like from a comfort point of view.
137	So let's just mirror the line here and see what we got. Oh! Why I do that?
138	Whether we could gain if this was the generator's space from a maintenance point of view if we could get generators closer together. How could we get these generators closer together? At the moment, the shaft line is about 1200. If we could get these generators together, we could limit space between them to 800 mm. That will do for us And where is the centre of the generators? That's the distance of centreline. That is 1770.

D_g = Space for magazine. D_a = Configuring I_k = Layer CIWS – LCMK9P. Past design case CIWS installation. O_k = Magazine arranged under-deck. <i>Application of design knowledge of under-deck location of magazine on current design.</i>	L_g = Knowledge of the arrangement of magazine to CIWS. K_r: Sc. I_k = Past design installation of CIWS O_k = Magazine arranged underdeck. T_r: Retrospect. Ltw = <i>expedience driven.</i>
Summary of task done	
D_g = Space for enough accommodation. D_a = Evaluation I_k = a) Accommodation layout in superstructure of past design. b) Superstructure is far forward O_k = Do not like from a comfort point of view.	L_g = Accommodation layout in superstructure. K_r: Sc. I_k = Similar past design on accommodation in superstructure O_k = Accommodation layout in superstructure. Superstructure is far forward. T_r: Retrospect.
Drawing comments	
D_g = Arrange generators closer. D_a = Exploring I_k = a) Shaft line is 1200 mm. b) Sufficient room for maintenance O_k = Move generators to 800 mm spacing between them. D_g = Arrange generators closer. D_a = Determining I_k = Centre of the generators. O_k = Distance between centres of generators is 1770 mm.	K_r: Derivation T_r: In-situ. ΔK = Knowledge of constraints on the generator space envelope

139	Right, we <i>know</i> that for our vessel of this type that the structure distribution at the bottom of the vessel will probably be of the order of 300 to 350 mm pitch. And chances are we could have a probably a much broader spacing on the main deck. We probably stick to 350 in order to maintain some alignment characteristics. We <i>know</i> in that deckhouse construction, the aluminium could limit the thickness to 350 mm distribution.
140	With that in mind we want to organise where the shipping operations based on that longitudinal distribution. So we can start to design some sort of likely areas of the boat where we want either some soft patches or engine casing to be situated. With those positions in mind it influences the way we organise our equipment.
141	So we got a distance of 800 mm. Ooh! Centre to centre is 1770 mm. 350 mm distribution is 1750. Ah! What is the next one? 2,100. Let's see what that means.
142	This would enable the centre of the generators to be lifted. What can we do? 2,100. That's still leaves us a certain amount of ship's side. 2.3 m Due to the tumble-home effect that may leave us with 2.0 m. What can we do with 2 m?

D_g = Distribution and alignment of structure. D_a = Standardising I_k = Similar past design – distribution of structure for bottom, deck and deckhouse. O_k = spacing of structure is 350 to maintain alignment characteristics. <i>Application of past design attribute on current design.</i>	L_g = bottom structure distribution. K_r: Sc. I_k = Similar past design on bottom structure. O_k = bottom structure distribution: 300 to 350 mm pitch. T_r: Retrospect. L_{nv} = expedience driven or conflict avoidance by using standard pitch of stiffeners of past design.
D_g = Locations for shipping operations. D_a = Associating I_k = Longitudinal distribution. O_k = Likely locations for soft patches or engine casing. D_g = Organisation of deck equipment. D_a = Associating I_k = Locations for soft patches or engine casing. O_k = Locations of equipment.	K_r: Association T_r: In-situ ΔK = Knowledge of size, location of soft patches and engine casing (i.e. deck opening in relating the overall deck structure) is influenced by deck longitudinal distribution. ΔK = Knowledge of deck openings influence locations where equipment will be sited.
D_g = Location of opening in relation to generators. D_a = Exploring I_k = a) Centre to centre of generators is 1770 mm. O_k = 350 mm distribution is 1750 or 2,100 mm between them.	K_r: Derivation T_r: In-situ. ΔK = Knowledge of location of generators in relation to the distribution of deck structure.
D_g = Lifting of generators in relation to openings. D_a = Evaluation I_k = a) Centre to centre of generators is 2100 mm. O_k = Ship's side is 2.3 m with tumble-home effect, 2.0 m.	K_r: Derivation T_r: In-situ. ΔK = Knowledge of lifting of generators in relation to deck openings.

02:44:00 /144	What I'm trying to decide is whether I can build on the main deck superstructure. I'm not sure if I can. If I box that in, what am I going to achieve? Subject to what's the arrangement in the deckhouse you need some sort of superstructure thorough-fare. Then you've to come round the casing and then go out by the side of the ship. That's a possibility. Don't' want to do that.	D_g = Build a main deck superstructure. D_a = Exploring I_k = a) Route for superstructure thorough-fare, round the casing and on to side of ship. O_k = Superstructure is a possibility but may not want to do that.	K_r: Derivation T_r: In-situ ΔK = Knowledge of the problem structure (i.e. relationship between superstructure and thorough-fare).
145	Move that for a moment And 1, 2, 3, 4. Emm. Right 4. But it could go on the side Let's work on the premise of At 3m, 42. I assume this is to be the engine casing	Comment on drawing instruction	
	So we could have the superstructure up to this point. No, don't trim this. Please don't do this. Please don't do this. Oh No! Because this is an autoship project file there could be countless lines on top of each other hundreds of lines and this is still busy working away.	Comment on CAD system	
	Well it's fast approaching the time I'm going home.	End of day Comment	

The details of the coding of segments of the protocol are given in Appendix 1. Appendix consists of four columns. Column 1 indicates the segment number of the protocol segment shown in column 2. In column 3, the segment has been analysed into the elements of the identified design activity (i.e. the design goal, the input knowledge and the output knowledge). Column 4 shows the knowledge change inferred from column 3 and the corresponding knowledge transformer. Where evidence of retrospective learning is identified, the formalism representing the learning activity is used to depict the elements of the activity.

Appendix 2

1 Coding Schemes

Protocol data is very rich in cognitive information but highly unstructured. A video/audio protocol of a design session represents a particular form of qualitative data. It can provide a deep insight into the design process from a perspective of interest leading to a better understanding of the process. However, the transcripts are not the actual data on which analysis is performed. Since design involved the application of declarative and procedural knowledge, the purpose of protocol analyses in design has been to elicit the knowledge structure present in the protocols through some coding schemes. Hence a number of coding schemes have been evaluated so that either a suitable coding can be used or a variant of it devised.

1.1 Review of Coding Schemes

The purpose of the review of coding schemes used by researchers in analysing protocols of design type experiments is to understand the nature of the coding scheme used and the nature of the results that can be derived through encoding the protocols.

According to Dorst and Dijkhuis [1], protocol analysis methods are divided into two categories:

- The process-oriented approach, in which design processes are described in terms of a general taxonomy of problem-solving, i.e. problem-states, operators, plans, goals, strategies, and so on.
- The content-oriented approach which aims at revealing the contents of what designers see, attend to, think of and retrieve from memory while designing.

Process based coding

Purcell et al's [2] coding scheme incorporated three broad classifications, two concerned with the designer's place in the problem space while the third was concerned with the designer's activity at any time. The first classification (called Problem Domain: Level of Abstraction) reflects the complexity of the design problem in terms of the level of abstraction the designer is working on. The second classification (Problem Domain: Function, Structure and Behaviour) monitors the reasoning about the design in terms of its requirements, purpose, behaviour and structure. The third classification (called Strategy) describes the moment-to-moment activities of the designer during the design episode.

Using these codes to analyse the Delft experiments, Purcell et al. were able to discern patterns of design activities across the design episode and made comparisons across the three classifications. For example, the pattern of design activities revealed in the Delft experiments were the activities of proposing a solution, analysing a proposed solution, analysing the problem and consulting information about the problem. Based on the first classification, the overall pattern of design activity focussed on progressively decomposing the problem from the highest level of abstraction to the lowest level of abstraction.

Analysing the experiments in terms of the second classification (i.e. Problem Domain), the results showed that:

- at the beginning of the design session, more time was devoted to functional reasoning at the highest level of abstraction,
- a period of rapid changes between reasoning about the behaviour and structure throughout the design session associated with the activities of synthesising and evaluating solutions at the lowest level of abstraction, and
- slower periods of changes coinciding with analysis of the problem.

The analysis based on the third classification revealed that the activity of analysis dominates the earlier part of the design session when the designer is reasoning about Function and Behaviour. The analysis activity is followed by activities of synthesis and evaluation. These cycles correspond closely to the reasoning about behaviour associated with analysis and evaluation while structural reasoning is associated with synthesis.

In Gero and Mc Neill's [3] coding scheme the first dimension describes the design object in terms of the *Level of Abstraction* and *Function, Behaviour and Structure* and *Requirements*. The other two dimensions are Micro Strategy and Macro Strategy. Aimed at depicting the evolutionary stages of the design process in arriving at design solutions, there are sixteen categories under Micro Strategy dimension classified into three sub-groups: analysing a solution, proposing a solution and making explicit references to knowledge used by the designer. Included in Micro Strategy is a sub-group that identifies the role of domain knowledge, application knowledge and the procedural knowledge to adopt in progressing the design showing the association of design knowledge with the design actions. Under the Macro Strategy dimension, five distinct categories include codes to denote the top down (goal-directed) or bottom up (data-directed) approach in a solving problem. The top down approach often involves the decomposition of the problem, giving rise to a code for the activity of decomposition. Problem solving may lead to a dead-end, or an external influence that makes a change of direction advantageous, necessitates in codes for backtracking actions and opportunistic strategy.

Gero and McNeill [3] and Purcell et al [2] used a multi-tier coding structure to analyse protocol data of routine electronic engineering design sessions and the Delft experiments [4]¹ respectively.

Content based coding

Lloyd and Scott [5] impose a rather restrictive set of only three categories of utterances (talk aloud phrases or sentences) in their protocol analysis of a design exercise. Their single-tier coding structure encodes utterances as generative, deductive and evaluative utterances. However by restricting to three categories, there is a severe loss of detail from the original protocol.

¹ In the Delft experiment, the designer was given a problem to which he/she had not previously been exposed.

Cognitive and Perceptual Content

Design protocol studies exhibit both complementary verbal and visual data that reflect the designer's cognitive and perceptual processes. This understanding of the dual nature of the protocol data has prompted Akin and Lin [6], Suwa et al [7] to devise coding schemes that elicit both the cognitive and perceptual processes of the designer.

Akin and Lin [6] propose a model that depicts the protocol as a dual mode process in which the verbal-conceptual and visual-graphic data are dealt with, alternatively. They map the designer's activities of drawing, thinking, examining, talking, writing and listening (observed from the protocols) into the dual mode model. The activities of drawing and writing correspond to the visual-graphic mode. Three (i.e. thinking, examining and listening) correspond to the verbal-conceptual mode. However, verbalisation (i.e. the talking activity) is seen as serving the function of reflecting the contents of the other five activities, and not as a direct outcome of the design process. Hence in the activity-based model, speech is shown as a separate category only as an echo of all that goes on in the other five activities. Hence the talking activity corresponds to the 'transcription' part of the dual mode model.

The difficulty of such a model is that neither the verbal (transcriptions) nor the visual (drawings) data alone can explain the design process adequately. To overcome these problems, they augment the dual mode model with an activity-based model of the design process.

Given these modes of activities from a protocol, Akin and Lin investigated how the activities influence the emergence of design decisions that influence important features of the design. They hypothesised that while these decisions are made vis-à-vis the activities, thinking, drawing, listening, writing and examining, the evidence that narrates these decisions is present in the speaking activity.

Akin and Lin conclude that the activity model of the design protocol provided interesting results in terms of its contribution of activity modes to the development of designs. One such result is the correlation between the conceptual cum visual modes (Examining-Drawing-Thinking) and novel design decisions (NDDs). This result suggests that designers explore their domain of ideas in a variety of activity modes (three out of three, in this case) when they go beyond routine decisions and achieve design breakthroughs. The other result is the dependency between the two data forms, verbalisations and drawings and the hypothesis is that transcriptions and drawings may be echoes of each other. Subjects verbalising in protocol experiments, in essence, are echoing what they are doing. Thus, in protocol analysis the primary evidence should be collected from data outside the verbalisations, and the transcriptions should only be used as a secondary source to the former.

Hierarchical Scheme

Suwa et al devised a hierarchical coding scheme to systematically code cognitive actions of designers from their video/audio protocols. The hierarchical coding scheme codifies cognitive actions into four major categories that correspond to the

cognitive levels of information processing []. These categories are physical, perceptual, functional and conceptual. Physical actions correspond to sensory level, perceptual actions to perceptual and both functional and conceptual to semantic level. Each major category, in turn, is classified into subcategories.

Three subcategories of physical actions refer to actions that have direct relevance to physical depictions on paper. The D-action is to make depictions (D) on paper, such as diagrams, figures, symbols, annotations, memos and even sentences while the M-action denotes the motion (M) of pencil or hands that does not end with physical depictions. The L-action looks (L) at existing depictions.

Known as P-action, the perceptual category refers to actions of attending to visuo-spatial features of depicted elements on sketches under three subclasses. The first subclass caters for visual features of elements (e.g. the shapes, sizes and textures). The second and third subclasses cater for the spatial relations among elements, (i.e. proximity, remoteness, alignment, intersection, connectivity) and the organisations and comparisons among the elements (e.g. grouping of elements, the similarity/uniformity and the difference/contrast of the visual features of the elements) respectively.

The functional action (known as F-action) refers to actions of conceiving non-visual information that depict elements and their visuo-spatial features that those elements are able to carry. This category consists of two subclasses. The first one concern the interactions between designed artefacts and surrounding natural resources, e.g. the circulation of people in a spatial configuration (the design artefact). The second subclass of functional actions refers to the psychological reactions of the people when they interact with the design artefacts (e.g. curvilinear pathway might give visitors a sense of 'fascination' or 'expectation').

The conceptual category has three types of cognitive actions that are not directly suggested by physical depictions or visuo-spatial features of elements; E, G and K actions. E-actions refer to the designer's preferential or aesthetic evaluation (E) of P-actions or F-actions. G-actions reflect the set up of goals in either the bottom-up or the top-down manner. Bottom-up goals are to trigger the depicted elements in terms of the visual spatial features (i.e. P-actions) or the non-visual information (i.e. F-actions). Top-down goals or sub-goals are initiated when a designer divides the current problem into sub-problems that in turn give rise to other actions in a top-down manner. K-actions refer to retrieval of knowledge (K) from memory in either the forward reasoning or backward reasoning mode. The forward reasoning involves applying knowledge to infer new information from existing information. In the backward reasoning mode, the information which is necessary to substantiate known, or existing information is determined.

Dependency relations and temporal relations described the relationships that may exist between actions in the same or different categories. The action categories coded in an upper level can be dependent on other actions in the lower levels. The dependency relations between different level of actions are *dependent on*, *suggested by* or *triggered by*. The temporal relations are 'new', 'continual' and 'revisited'.

The design action is indexed as 'new' when a designer makes a particular depiction, attends to a particular feature, thinks of a particular function, set up a particular goal, or retrieves a particular piece of knowledge for the first time in the design session. For the segment that follows immediately a previous segment, the index is 'continual'. A design action that was done earlier but subsequently worked on again is indexed as 'revisited'.

Suwa et al [7] applied the coding scheme on protocol (visual) recording design actions of a practicing architect with retrospective reporting on the actions taken when the video recording of their actions were replayed. They were able to conclude that design sketches serve not only as external memory or the provider of visual cues for association of non-visual information, but also as a physical setting in which design thoughts are constructed dynamically. They conclude that designers' cognitive actions are definable in a systematic manner using the hierarchical coding system and that a designer's cognitive behaviour can be represented as a structure composed of defined primitive actions.

The purpose of imposing structure on protocols through a coding system was therefore to test existing hypotheses and models of the design process. Wide variations in the categorical structures imposed may actually inhibit comparisons between protocol experiments and the development of theory. Hence Purcell et al use the dual approach of theory generated categories collaborated by data generated categories derived from protocol analysis. Ericsson and Simon posit that encoding and analysing empirical data – whether verbal reports or other kinds of data – always takes place in some kind of theoretical context. Ericsson and Simon [8] advocate the use of model based coding to identify relevant information heeded in a protocol. Based on this recommendation, it is proposed that a coding system based on the formalism of design activity and learning activity developed in Chapters 5 and 6 be formulated for the protocol analysis of experiment reported in this Chapter. No distinction is made here as to whether the coding should be process-based or content-based. This is because both aspects of design are relevant here in terms of the designing activities which are process based in nature and the knowledge change in these activities which is inferred from the information/knowledge content of each activity.

References

- [1] Dorst, K and Dijkhuis, J. (1995). Comparing paradigms for describing design activity, *Design Studies*, Vol. 16 No. 2, 261- 274.
- [2] Purcell, T., Gero, J., Edwards, H., and McNeill, T. (1996). The Data in Design Protocols: The Issue of Data Coding, Data Analysis in the development of Models of Design Process, in Cross, Christiaans & Dorst (eds), *Analysing Design Activity*, 35-63, John Wiley & Sons, 1996.
- [3] Gero, J. S., and McNeill, T. (1995). A method of protocol analysis of design sessions, *Research in Engineering Design*.
- [4] Cross, N. Christiaans, H and Dorst K.(Eds.). (1996). *Analysing Design Activity* John Wiley & Sons.

- [5] Lloyd, P. and Scott, P. (1994). Discovering the design problem, *Design Studies*, Vol. 15 No. 2, 1994, 125- 140.
- [6] Akin, O. and Lin, C. (1996). Design Protocol Data and Novel Design Decisions, in Cross, Christiaans & Dorst (eds), *Analysing Design Activity*, 35-63, John Wiley & Sons.
- [7] Suwa, M., Purcell, T., Gero, J. (1998). Macroscopic analysis of design processes based on a scheme for coding designers' cognitive actions, *Design Studies*, Vol. 19, 455-483, Elsevier Science Ltd.
- [8] Ericsson, K. A. and Simon, H. A. (1993). *Protocol Analysis, Verbal Reports as Data*, The MIT Press.

Appendix 3

Meaning of symbols used in Backus Nur Form

Variables in the BNF definition are enclosed in $\langle \rangle$.

A metasymbol is a special character from the BNF alphabet itself.

Here, the metasymbols include [$:=$, $<$, $>$, $|$, $\{$, $\}$, $\rightarrow$, $\parallel$].

The metasymbol $\{u\}$ denotes zero or more repetitions of the string inside the brackets $\{$ $\}$.

The meta-symbol $\rightarrow$ in the expression $a \rightarrow b$ denotes that b succeeds a .

The meta-symbol $a \parallel b$ denotes that a occurs concurrently as b .

The start symbol of the BNF definition of the model of learning in design is the variable $\langle \text{learning in design} \rangle$.