Customised Viewpoint Support for Utilising Experiential Knowledge in Design

by

Sandra MacKenzie Kerr

A thesis submitted for the degree of Doctor of Philosophy

CAD Centre
Department of Design, Manufacture and Engineering Management
University of Strathclyde,
Glasgow, Scotland, UK

December 1993

- Dedicated to my parents -

For giving me opportunities in life they never had and for their patience, love, support and understanding.

Abstract

This research aims to improve the effective utilisation of experiential design knowledge by supporting the extraction and subsequent use of knowledge from a store of design experiences. Current computational approaches that support the utilisation of experiential knowledge promote the *regurgitation* of knowledge from pre-defined viewpoints reflecting knowledge engineers' perspectives of designers' knowledge needs. However, from an investigation into the application of experiential knowledge, it is argued that designers can generate numerous viewpoints according to their own particular perspectives. Consequently, the perspectives imposed by current approaches may be of little use in design if they do not map onto those needed by a designer. A new approach, called 'customised viewpoint', is presented in this thesis as one that promotes the *application* of more relevant knowledge by generating appropriate viewpoints according to designers' perspectives. Numerical design is presented as a well-defined problem area within which this approach is developed, tested and evaluated. The PERSPECT system is the realisation of a 'customised viewpoint' tool developed by integrating and extending the functionality of three relevant existing systems: DESIGNER (a numerical design system), S-PLUS (an extensive data analysis package), and ECOBWEB (a concept formation system). PERSPECT provides valuable assistance; it supports a designer to (a) render new numerical domain models or check and update existing ones in the light of new design experiences, and (b) develop a design solution by (i) supporting the opportunistic utilisation of empirical equations and generalisations from generated customised viewpoints and (ii) reducing design complexity via the abstraction of an existing domain model. However, further work is required to improve PERSPECT's ability to support numerical design. The 'customised viewpoint' approach has been shown to compliment the CAD philosophy of "design assistance" but extensive work is still required to realise an ideal 'customised viewpoint' tool that fully supports the needs of practising designers.

Acknowledgements

At the early stages of writing up, I remember thinking my acknowledgements would be the easiest part to write compared to the main body of my thesis. I was naively optimistic! Finding the right words has been by no means straightforward. However, though difficult it has been the most enjoyable part; I now have the chance to express my appreciation to those who have technically and emotionally helped me through the last few years.

First of all I wish to thank Dr Alex Duffy, my supervisor. He has not only supervised me but trained me to write, listen, think and formulate answers better. With his ceaseless penetrating questions, dormant parts of my brain have been awakened. This has been a painful and exhausting process - for both of us - and the time and effort he has spent is greatly appreciated. I am also indebted to him for his sensitive supervision that he exercised, after my period of illness, to bring me back to 'full speed' working. His support, understanding and careful management was invaluable in helping me pick up the fragments of my work, and his ever constant encouragement helped to maintain my own enthusiasm and energy till completion. He has made my struggle that bit more bearable.

I would like to thank Professor Kenneth MacCallum, the Director of the CAD Centre, for providing a congenial working environment, and the staff and students of the CAD Centre for their friendship, encouragement and assistance.

I greatly appreciate the fruitful discussions with Mr Alan Gilfillan and Dr David Barry (both from YARD Ltd). They took time out from their own work commitments and, when this was not possible, used their own valuable personal time to assist in the evaluation of this work.

I would also like to thank Professor Kenneth MacCallum and Dr Alex Duffy for their work on building and developing DESIGNER and Dr Yoram Reich for his work on

ECOBWEB; two systems which have been used in this research.

I wish to thank the Science and Engineering Research Centre for their financial assistance during my PhD. In addition, many thanks are due to the Royal Pinner School Foundation who financed my final year, which allowed me to complete my thesis free of financial worry.

Many thanks are due to my family and friends for their concern and for listening patiently to the highs and lows of PhD life.

In particular, I'd like to thank my 'big' sister (Elaine) and my parents. Elaine has shown an impeccable sense of timing and astuteness in knowing when 'the coast has been clear' to talk about my PhD. My parents have put up with me throughout my undergraduate and postgraduate years - not an easy task. Both have cared for and supported me when I needed it most; then uncomplaining, they stood back while I devoted my time and energy to work. They have constantly believed and trusted in my abilities. Therefore, without them and their love I would never have achieved the goals I have set myself.

I would also like to thank my Mother for introducing me to the delights of a glass (or two) of Brandy and Babycham at the end of a bad day.

Last of all, a special 'thank you' to "TC", who provided my means of escapism after which I regained my sense of humour.

Contents

Appendices 184

List of Figures

List of Tables

1 Introduction

Early workers in design were craftsmen who typically lacked formal design education or training, and practised what is now called vernacular design. Their knowledge of their "craft" was predominantly *experientially* based (i.e. originating from practice), initially acquired during their apprenticeship under the guidance of a 'journeyman' and further expanded throughout their life as a craftsman. Craftsmen developed artefacts in parallel to manufacture. Examples of some artefacts of vernacular design are wooden clogs [1], wagons [2] and traditional tools such as hammers, saws and axes [3]. If the development was unsuccessful, the result, by then realised as an artefact, failed (sometimes with drastic consequences). Therefore, vernacular design became expensive with material, time and/or lives. In addition, this trial and error approach to design became increasingly infeasible as design complexity increased. To improve the likelihood of success and avoid costly mistakes, the application of science in design became more prevalent [3, 4]. This gradual introduction of science presented *theoretical* knowledge that supplemented the *experiential* knowledge of early craftsmen; however, for its effective application by designers, it required to be formally taught. Thus vernacular design evolved from a craft, previously carried out by people whose understanding of design (experiential knowledge) was acquired from practice, to technological design requiring designers with more formal training to acquire and utilise necessary (theoretical) knowledge [3].

In today's environment, it is almost impossible to escape the realisations (artefacts) of science in design; they permeate the world around us. For example: the application of the science of aerodynamics in aircraft design [3], the science of ecology in landscape design [5, 6], the science of electromagnetic induction in the design of electrical machines [7], the science of hydrodynamics in ship design [4], the science of material (e.g. metals, alloys, polymers, ceramics, glass and composites) properties in material design [8], the science of mechanics, materials, control and the environment in process design [9], the science of psychology in graphic design [10] and 'man/machine interface' design [11], the science of thermodynamics in steam engine design [7] and the science of metallurgy (metal properties) in lifting vehicle design [12]. Today, some of the most useful and/or abundant artefacts stem from technological design. These artefacts are useful because they fulfil some desire or need. Irrespective of size (e.g. from a chair to a multi-storey building), complexity (i.e. from relatively simple artefacts to complex ones, e.g. from a pin to a turbine) or design domain (e.g. from aircraft design to ship design), artefacts are the embodiment of ideas that are expected to fulfil successfully some need or desire. Therefore, science plays an important role in our everyday lives.

The development and utilisation of theoretical knowledge in design, and the desire to capitalise on this knowledge have, in the past, led to the development of Computer Aided Design (CAD), i.e. the application of computers in the design process to assist designers in the development of a design solution. Traditionally, CAD systems have supported detailed analysis of designs and have required the representation and utilisation of theoretical knowledge to help assess the performance of designs. However, there is more to design than analysis. Design encompasses exploration, synthesis and evaluation activities (see Chapter 3.3) and uses both *experiential* and *theoretical knowledge*, as depicted by the following:

> "... [design] an irrational search, conducted over ground prepared by experience, the study of principles and the analysis of site and purpose" [13]
> "... experience is equally important as theories in engineering design" [14]
> "... [engineering design] a creative application of engineering theory and experience to achieve a practical objective that satisfy a market need." [15]

Therefore, if CAD is to be realised throughout design then these activities, and the representation and utilisation of experiential knowledge should also be addressed. Such systems are beginning to appear and are directed to support early design stages, which are less analytical and more creative in nature. In other words, CAD systems now emerging help generate as well as analyse design solutions. However, in general, these systems do not utilise experiential knowledge effectively. They typically support the representation of pre-defined experiential knowledge, support limited generalisation of implicit knowledge, and do not support the abstraction and generalisation of knowledge to suit the particular needs of designers. In other words, current CAD systems are deficient in the following areas:

- *Richness of experiential knowledge*

  They do not fully integrate the richness of experiential knowledge originating from multiple sources (e.g. design processes, past designs and experiments) or opportunistically utilise multiple forms in which experiential knowledge can be expressed.

- *Learning experiential knowledge*

  They have limited learning capabilities to promote the effective utilisation of experiential design knowledge. Relatively little assistance is available to help designers develop their experiential knowledge of a design domain. In other words, the acquisition and evolution (i.e. learning) of represented experiential knowledge has been generally the sole responsibility of designers.

- *Viewpoints of experiential knowledge*

  They represent experiential knowledge that reflects pre-defined viewpoints of knowledge. In other words, the knowledge represented is dependent on knowledge engineers' perspectives of what knowledge is required by designers. Consequently, these systems do not support the utilisation of knowledge that reflects designers' knowledge requirements.

A goal of Intelligent CAD (Int.CAD) is to provide a more effective and efficient "design assistant" tool [16] by promoting the interaction between and effectiveness of a designer and computer through the application of artificial intelligence techniques. The scenario presented by Int.CAD can be described as two very different entities (a designer and a computer) working together to fulfil a design need. Both components depend not only on their own store and processing of knowledge but also on each other.

The applicability of experiential knowledge represented in a CAD system is instrumental to its success or failure. If a system represents knowledge that is not usable in subsequent design problems, then the system fails to assist the designer. This thesis focuses on improving "design assistance" by developing an approach that addresses the deficiencies of existing CAD systems and supports the generation and utilisation of customised, rather than pre-defined, viewpoints of experiential design knowledge. It presents, illustrates and evaluates an approach that supports the development (learning) and utilisation of experiential design knowledge according to the needs of designers. In other words, this thesis is motivated to increase the success of CAD systems.

The following outlines the structure of this thesis. **Chapter 2** defines the objective and aims of the research reported. **Chapter 3** investigates experiential knowledge utility in a number of design process models to ascertain *why* such knowledge is used in design. **Chapter 4** investigates the application of experiential knowledge, that is *how* knowledge is applied, to identify a list of general requirements for computer support that effectively utilise experiential knowledge. **Chapter 5** critically reviews existing computer support, according to these requirements, to assess existing approaches and techniques, and emphasise the need for improving support for the utilisation of experiential knowledge. **Chapter 6** presents a new approach proposed to compliment the "design assistance" philosophy of CAD. The **'customised viewpoint'** approach is proposed to support the update and evolution of experiential knowledge, the generation of appropriate viewpoints of experiential knowledge and the utilisation of applicable knowledge. **Chapter 7** describes a well-defined problem area, i.e. preliminary numerical design, within which the new approach is developed, tested and evaluated. This chapter then continues with an overview of three relevant existing systems that address some of the features of a 'customised viewpoint' tool, discusses their contribution and limitations to the development of this new approach in the chosen problem area and therefore identifies further required functionality to be supported. **Chapter 8** presents how this required functionality can be achieved. **Chapter** 9 presents the architecture and some of the main features of a system called PERSPECT, i.e. an implementation of the new approach. **Chapter 10** illustrates, highlights and evaluates the utility of a numerical 'customised viewpoint' tool by discussing an exemplary design session of PERSPECT. **Chapter 11** discusses more general aspects of the new approach that can advance its applicability in design: i.e. its ability to support designers' creativity, ways in which the approach can be improved and avenues of further research. **Chapter 12** closes the thesis with the main conclusions resulting from the work presented.

2 Aims

The overall aim of the research presented in this thesis is to improve the effective utilisation of experiential knowledge during the development of engineering design solutions. This requires supporting the extraction and use of experiential knowledge, based on designers' knowledge needs, from a source of past designs. To achieve this aim, the following objectives have been identified:

- To identify the roles experiential knowledge can play in design by developing a framework of design from which to explicate the utility of experiential knowledge in design, i.e. *why* experiential is utilised in design.

- To identify general requirements of computer support for the effective utilisation of experiential knowledge in design by investigating *how* experiential knowledge is applied in design.

- To identify the general deficiencies of existing computational approaches that support the utilisation of experiential design knowledge by critically reviewing them according to the identified general requirements.

- To propose a new approach that compliments "design assistance" by addressing the deficiencies of existing CAD systems' approaches that utilise experiential knowledge.

- To identify the contribution and limitations of existing technology suitable for the implementation of the new approach within a well-defined design problem area, i.e. preliminary numerical design.

- To implement the new approach such that computational support for the effective utilisation of experiential design knowledge can be demonstrated.

- To test and evaluate the utility of the developed implementation within the chosen problem area.
- To evaluate the utility of the new approach and propose avenues of further research.

3 Utility of Experiential Knowledge in Technological Design

The introduction of this thesis acknowledged the utilisation of experiential and theoretical knowledge in design. However, it criticised the limited assistance designers have in developing and utilising experiential knowledge, as opposed to the more formal approaches available for utilising theoretical knowledge. Few studies exist that address why and how experiential knowledge can be utilised in design. Consequently, the collective aim of this chapter and the following Chapter 4 is to formulate a basic understanding of the implications of utilising experiential design knowledge.

This chapter investigates experiential knowledge utility in a number of design process models to ascertain *why* designers use this type of knowledge. Previous discussions of design process models have failed to make explicit the roles experiential knowledge can play in design. For example: Coyne [17] discusses the contribution to design understanding by adopting science, logic and language paradigms of the design process with the explicit aim of developing design process models to be computationally implemented; and Finger & Dixon [18] present a classification (i.e. prescriptive, descriptive and computational) of design process models defined by the investigative basis with which models have been developed.

The following sections discuss the utility of experiential knowledge in three relevant design process models, which make up a design framework. Each section refers to a component of this framework, and each component relates to a design process model that reflects an increased focus on the design process. Two of the three models exist in design literature; the remaining model represents one that has been expanded for the purpose of this thesis. The three components used to identify the roles experiential knowledge can play in design are **types, phases and activities**.

Section 3.1 entitled *design types* discusses a model of design that refers to the various natures of the design process and represents the most global classification view of design in this chapter. Section 3.2 entitled *design phases* discusses another design model that proposes the general structure of the design process for all design types. Section 3.3 entitled *design activities* discusses necessary activities that designers execute for the effective completion of the global design phases. The *design activity* component differs from the previous two since it presents and discusses an extension to an existing design model. Section 3.4 summarises the presented framework and the identified roles of experiential knowledge in design.

The aim of this chapter is to identify the roles experiential knowledge can play in design. This is achieved by collating a number of applicable design process models into a framework of design from which to discuss the utility of experiential knowledge in design.

3.1 Design Types

This thesis defines design types as referring to the nature of the design process, which involves focusing on issues such as the existence and/or amount of 'innovation' employed during design.

There is a varied opinion on how the differences among design types are defined. For example, according to different degrees of design artefact 'newness' that causes a change in the potential customer's behaviour or would require customers learning to utilise the artefact [19]; the amount of new knowledge and change in manufacture required to realise a design [20]; the combined influence of design requirements, process and design solutions [21]; the change in a design's components and linkages [22]; the amount of information and problem solving strategies used [23, 24]; or a new design's location relative to a space of possible designs [25]. However, the classification that facilitates the most effective discussion on the utility of experiential design knowledge is that presented by Pahl & Beitz [21] as the **original, adaptive** and **variant** design type classification.

Pahl & Beitz classify technological design by focusing on the requirements, process strategies and design solutions. They classify requirements as being either *new* (i.e. previously unseen) or *old* (i.e. previously seen), process strategies as being either *elaboration, adaptation* or *variation*, and design solutions as being either *completely new, partially new* or *previously seen.* Original design involves the *elaboration* (i.e. development) of an idea that results in the definition of a design that is *completely new*; the motivating requirements behind original design can be either *new* or *old.* Adaptive design involves the *adaptation* (i.e. modification) of a *previously seen* design to serve a *new* requirement and consequently involves parts or assemblies of the previous design to undergo change. Consequently, adaptive design generates a *partially new* design. Variant design involves the *variation* (i.e. adjustment) of size, arrangement or attribute values of a *previously seen* design to satisfy *new* design requirements. Therefore, original, adaptive and variant design types are dependent on different combinations of requirements, process strategies and design solution types.

Pahl & Beitz illustrate that the degree of experiential knowledge generated from previous requirements, processes and designs influence the type of design designers carry out. Consequently, to carry out different types of design, designers require a sound understanding of satisfied design requirements, the applied processes and past designs.

Culverhouse [20] identifies somewhat similar design types to those discussed here, i.e. repeat order design, variant design, innovative design and strategic design. The precise definitions of these types are not of direct relevance here. What is important is Culverhouse's illustration of the effect a shift in design type has on the production (i.e. realisation) of designs. The effect is that the greater the degree of innovation in the design the greater the percentage change to production. Figure 1 illustrates an adaptation of his diagram. Each square, shaded differently, represents a different design type.

The size of the squares directly reflects the degree of originality of the design type, e.g. the larger the square the greater the degree of originality. This figure shows that moving from one design type to another, in a direction that requires more originality (i.e. moving away from the origin in the figure), requires a change in production. This change in production can have undesirable cost penalties to the design as a whole.

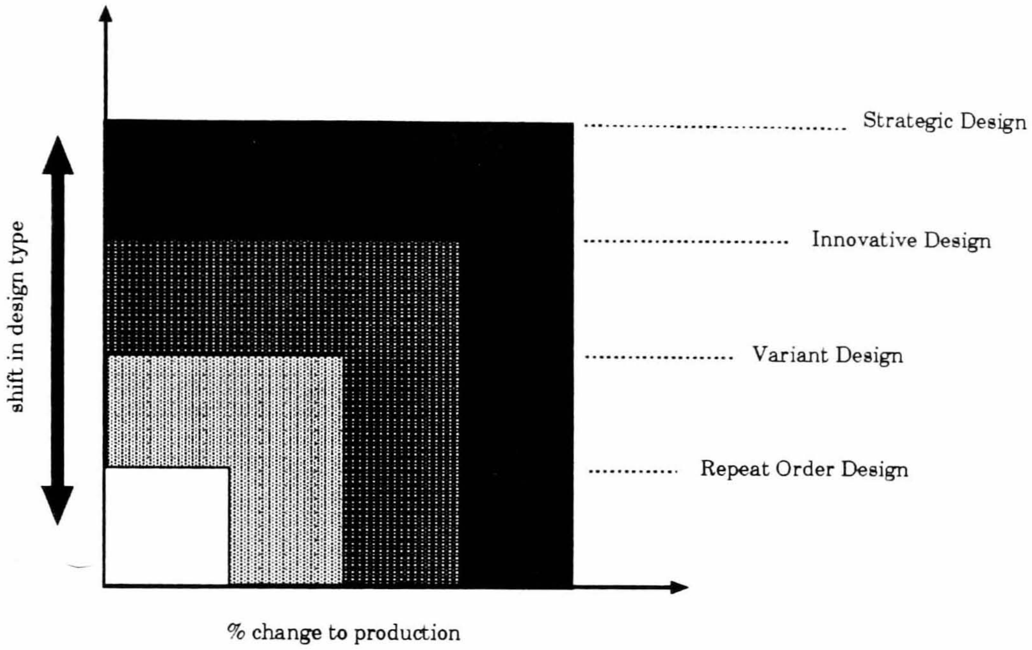


Figure 1: Effect of shift in design type to production, adapted from [20]

This finding by Culverhouse emphasises the need for designers to be aware of design types and to monitor their design type shifts so that they can accommodate appropriate production changes, or prevent unnecessary shifts and avoid changes to production.

One of the keys to identifying shifts in design type is experiential knowledge. Culverhouse has developed a *metric* (involving among other things experiential knowledge) for determining the design type of a new design or part of a design. Using this metric, designers can determine their new design's degree of originality and identify when their designs require a shift in design type. Consequently, acceptable production changes can be accommodated or undesirable changes can be prevented.

In summary, this section has proposed that designers can utilise experiential knowledge to carry out all types of design, to ascertain a degree of originality/innovativeness of their new design, and to identify shifts in the type of design they are carrying out.

3.2 Design Phases

Design phases represent stages of the design process, which progressively focus on more detailed levels of a design solution as it develops. The opinions [21, 26, 27] on the detail of these levels have varied little since Asimov presented the conceptual, preliminary and detailed levels [28]. However, perhaps one of the most noticeable extensions to these levels has been the acknowledgement of design totality [7, 12]. In Pugh's model of design totality [12], this concept encompasses marketing, through concept, preliminary and detailed, to manufacturing and selling phases. The importance of experiential knowledge throughout the design process manifests itself as a resource with which designers can generate initial design concepts or resolve specific manufacturing details, as acknowledged by Kamil, Vaish & Berke in their brief review of the aerospace design process [29]. Manufacturing and selling are important phases that impinge on the success of a design, and are recognised by such approaches as "design for manufacture" [30] or "design for assembly" [31] and effective selling strategies (e.g. advertising [32], packaging design [33]). However, this thesis does not consider these approaches as being predominantly influential in the effective use of experiential knowledge for generating and developing design solutions. Therefore, to help highlight the role of experiential knowledge in design, this section discusses the following design phases: **marketing, problem specification, conceptual, preliminary** and **detailed design** (see Figure 2).

Design is often initiated by the recognition of some need [12, 34]. This stage is called *marketing*. Designers carry out marketing to develop an understanding of the design climate from which they can identify, develop and/or assess needs. Hisrich & Peters

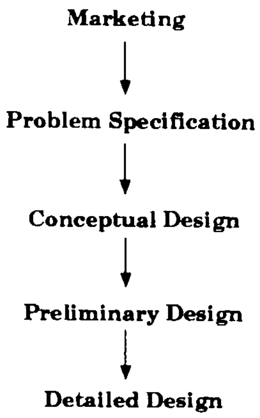


Figure 2: Design phases

[35] list a number of factors that designers should address when considering needs. For example: they should determine whether the type of need is well established, emerging or existing within a limited time span; they should check whether the timing of the human need coincides with the commercial need; and, if a measure of the market size is relevant, they should determine the potential market size that is to be satisfied. Satisfying a design need that is either timely, realisable or commercially viable can have penetrating effects on the success of a design solution. By attempting to minimise the potential number of inappropriate design needs, which designers could have developed into fully specified designs, designers can minimise the amount of design solution failures. Therefore, marketing is an important phase that requires the utilisation of results from market surveys, customer discussions and experiential knowledge of the design domain (e.g. past designs and current design trends) to help designers understand the design climate and desires of potential customers (i.e. users and/or owners) of a design.

After marketing, the design need is then formulated into a general statement of requirements during the *problem specification* phase. This involves further discussions with customers so that designers can clarify the true needs of their customers. It is important for designers to generate as broad a problem specification as possible, since the number and scope of alternative solutions that they can consider increases as the specification broadens [36]. Therefore, designers use experiential knowledge of the design domain (such as knowledge of past designs detailing their components, attributes and associated values) to generate a general statement of requirements and define the design space (i.e.

the space of possible design solutions).

Following on from the general statement of requirements, design progresses from the abstract to the specific in three basic phases [12, 21, 34, 37]: conceptual, preliminary and detailed.

Designers identify, develop and assess basic solutions to design requirements during the *conceptual design* phase. It is here that creativity is at its peak and the generation of ideas is dominant. Achieving design creativity requires a delicate balance of conservatism and adventurism by a designer. The bridge between these two extreme states of mind is design experience. This informs designers of what exists in a design domain and consequently can tend to inhibit their creativity to the bounds of the past [38]. However, by generalising and abstracting knowledge of the domain, and possibly from other domains, designers can 'jump' from the past to the present and thereby use experiential knowledge to contribute to adventurism [39]. The output of the conceptual phase is one or more design concept(s), which designers specify to such an abstract level that it (they) is (are) deemed acceptable. At this point, goals, constraints and criteria describe the design concept(s), and from this designers start the preliminary design phase.

The *preliminary design* phase further develops and assesses design concepts to facilitate the identification of the best or most acceptable design solution. To develop and assess possible design solutions, designers quantify design concepts to a more detailed level. This involves the identification [40, 41, 42], manipulation and management of complex interactions between design attributes [43, 44]. Designers can quantify these interactions, e.g. empirical equations generated and developed from past design information. Therefore, general experiential knowledge (e.g. empirical equations, correlation graphs, etc.) generated from specific past designs play an important role in the development of design concepts. For example, Fazio, Bedard & Gowri argue that designers (architects) rely on knowledge and guidelines, derived from specific previous designs, during this phase [45]. Consequently, this phase advances the conceptual design solution to a more detailed level of abstraction using experiential knowledge. However, it is the detailed design phase that refines the solution into a detailed design specification from which manufacturing can start.

The *detailed design* phase refines the design solution by detailing the components and sub-components that make up the design, and requires a comprehensive knowledge of component interactions. Designers have a number of design options open to them concerning possible refinements (e.g. material used, component parts, etc.) during this phase. Experiential knowledge can provide designers with the necessary understanding with which to base decisions on. For example, identified cost patterns help designers assess the viability of the new design compared to previous designs, or the components of previous designs can be used to help decide the components to use in new designs.

3.3 Design Activities

The design process progresses by expansion and contraction of solutions [46]. Designers carry out activities that facilitate the effective completion of design phases by expanding and contracting the description and/or number of design solutions.

Two commonly acknowledged design activities are synthesis ([12, 38, 46]) and evaluation [7, 14, 47, 48]. Synthesis typically results in the expansion of design solutions (i.e. the refinement of the description of possible design solutions or the generation of different solutions). To contract the space of possible solutions (i.e. constrain the degree of choice available to a designer) and converge a final acceptable design solution, designers evaluate these solutions. In addition to these activities, this thesis suggests the existence of an exploration activity. Designers can use this activity to generate suitable knowledge to assist in the expansion or contraction of design solutions. Consequently, three of the most significant design activities are **exploration, synthesis and evaluation**.

Smithers et al [38] take a more global view of the role exploration plays in the design process. In their discussion of design, they model the complete design process as that of an exploratory activity within which designers carry out a number of different kinds of activities, e.g. requirement description, decomposition, synthesis, analysis, optimisation. This thesis agrees to some extent with this exploration based process model of design in that exploration is carried out in design and that the result of this exploration is the acquisition and generation knowledge. However, this thesis emphasises the importance and execution of exploration on par with synthesis and evaluation activities, whereas Smithers et al consider synthesis and evaluation as ways in which exploration is achieved.

These three activities take into account Mayall's "principle of [design] iteration" [7] stated as the following:

> "...design requires processes of evaluation that begin with the first intentions to explore the need for a product or system. These processes continue throughout all design and development stages to the user himself, whose reactions will often cause the iterative process to continue with a new product or system."

An important aspect of this principle is the failure to mention the sequence in which designers execute design activities. In other words, do design activities start with exploration, synthesis or evaluation? Mayall professes that there "is no golden rule covering all areas of design work so far as this issue is concerned" [7]. Without a statement of an initial design need, originating from user evaluations, the exploration and synthesis of a design solution would be difficult to achieve. Alternatively, without exploration to identify a design need, the synthesis and evaluation of a solution would also be difficult. To promote the success of a design, designers explore its ability to satisfy design needs and remove any undesirable performances by further synthesis and evaluation. Hence, designers execute design activities iteratively and in no fixed sequence. However, this section presents these design activities in a sequence only to illustrate the utility of experiential knowledge in design.

Executing design activities involves *investigating* available knowledge to generate new knowledge, *utilising* this new knowledge and *assessing* the utility of that knowledge to determine its suitability. The following defines in greater detail these design activities.

The **exploration** activity *identifies* and *extracts* trends in existing knowledge, i.e. it promotes the creation of potentially useful knowledge. Oxman emphasises the execution of experiential knowledge exploration using past designs to identify and extract knowledge structures which are of relevance in a new design problem [39]. The **synthesis** activity *utilises* knowledge generated during exploration, thereby expanding the description of the required design to a more detailed level. Moneo's work acknowledges the exploration and synthesis activities via his acknowledgement of the utilisation of a knowledge abstraction called "types", which he regards as generated (by exploration) from past designs [49]. The result of designers exploring and synthesising can then be assessed during the **evaluation** activity. This activity *assesses* a value of 'fitness' for results using some criterion (e.g. quality, usefulness, cost, strength, etc.) or by comparison with a previous result (e.g. the performance of past designs). Therefore, evaluation provides a valuable means with which to determine knowledge quality or solution feasibility.

Designers execute these activities during each design phase. For example, during the marketing phase, experiential knowledge of "analogous and directly competitive products" [50] (i.e. past designs) are explored to identify and extract market trends, which are subsequently used to synthesis (i.e. develop or modify) the design need into a more comprehensive design need or a number of more detailed design needs. Alternatively, these needs could be defined by a customer or generated by discussions with a customer. However, in either case designers evaluate these needs. Designers evaluate needs to assess their value and/or determine those most appropriate, thereby facilitating the development of a general statement of requirements through to a complete design specification. Another example of executing these design activities can occur during the preliminary design phase. In this example, designers can explore experiential knowledge of past designs to identify trends (i.e. relationships) between specific design attributes and to extract descriptions of trends (e.g. as empirical equations). These trends can then be used to build (i.e. synthesise) a model of the domain, which can then be subsequently utilised to synthesise the design solution. In other words, designers can use generated empirical equation to estimate the value of design attributes. The resulting estimated attribute value can then be assessed according to its desired value.

Therefore, in summary, exploration and synthesis activities promote creativity by identifying, extracting and applying experiential knowledge to develop design solutions, and evaluation activities exercise judgement by assessing the feasibility of extracted knowledge and proposed design solutions.

3.4 The Role of Experiential Knowledge in Design

This chapter has presented three components, i.e. **types, phases and activities**, that make up a framework of design used to identify the roles experiential design knowledge. Table 1 summarises this framework by giving a brief definition of the three components and associated design process model.

Using this framework, this chapter has discussed the utility of experiential knowledge. In summary, it can be concluded that experiential knowledge plays a crucial role in all components of the framework, i.e.

> "This [experience] forms one of the most powerful and important sources of solutions to design problems."[51]

Component	Meaning	Design Model
Types	Nature of design process	original adaptive variant
Phases	Structure of design process	marketing problem specification conceptual design preliminary design detailed design
Activities	Necessary activities to effectively complete global phases.	exploration synthesis evaluation

Table 1: Technological Design Framework

The following list summarises the identified roles experiential knowledge can play in design.

Experiential knowledge is a resource utilised by designers to:

- carry out all types of design, determine the degree of originality/innovativeness of their new design, and identify design type shifts in the originality of the new design,
- develop an understanding of the design climate by analysing the domain from which the required new design belongs,
- generate a general statement of design requirements and define the design space,
- maintain a balance between design conservatism and adventurism by utilising knowledge of the past, which can be abstracted to compensate for the specificity of that knowledge,
- quantify design concepts to a detailed level for manufacture, and
- explore and evaluate knowledge to identify and extract more general knowledge, which can be used to synthesise a domain model and a new design, and evaluate the feasibility or performance of the resulting new design according to the performance of past designs.

Although this chapter has discussed the importance experiential knowledge plays in design, it says very little about how designers apply such knowledge. In other words, it does not address what 'components' of experiential knowledge designers can utilise, how designers manage the complexity of experiential knowledge and how they maintain and use experiential knowledge. The following chapter addresses these issues.

4 Requirements for Supporting Experiential Design Knowledge

This chapter continues the discussion on experiential knowledge to identify general requirements of computer support for the effective utilisation of experiential design knowledge by investigating *how* designers apply experiential knowledge in design. Section 4.1 presents the constituent parts (i.e. 'components') of experiential knowledge that are utilised by designers. Section 4.2 then explains how designers acquire and maintain experiential knowledge by their ability to learn, and proposes that designers' ability to coupling design and learning promotes the effective utilisation of experiential knowledge. Section 4.3 discusses how designers, by taking different viewpoints, manage the complexity and variety of experiential knowledge. Section 4.4 summaries the development of computer support for the application of experiential knowledge and presents a number of general requirements for the effective utilisation of experiential knowledge in design. The chapter closes in Section 4.5 with a brief summary.

4.1 Sources, Trends and Forms of Experiential Knowledge

Designers take advantage of lessons learned from design experiences by generating and using experiential knowledge. To understand more clearly the process of applying experiential knowledge in design, this section discusses in detail the 'components' of experiential knowledge.

This thesis considers experiential knowledge to consist of three main components:

- **Sources** - the origins of experiential knowledge, i.e. information related to specific past designs and design experiments.
- **Trends** - implicit experiential knowledge existing within information related to past designs and design experiments.
- **Forms** - defined abstractions of experiential knowledge that explicate implicit knowledge in abstract and general terms, e.g. heuristics, empirical equations, generalisations.

Sources

Past designs and design experiments are two basic sources of experiential knowledge. The information associated with past designs and design experiments is considered here to be specific and explicit, within which implicit knowledge concerning design trends exists.

Trends

Design trends refer to knowledge that is implicit in the available explicit information of past designs and experiments. They indicate design styles or how past designs (whole or in part) have behaved under experimentation. Designers can utilise this knowledge to maintain uniformity (of desirable styles) among the characteristics of a new design and that of the past. Conversely, they can use them to avoid generating a new design that similarly follows identified trends (non desirable styles), or to avoid undesirable behaviour in their new design without necessarily putting the new design under extensive testing. Consequently, trends are valuable implicit knowledge and, when utilised in a new design problem, can provide knowledge on 'what to' design.

Forms

General experiential knowledge is considered in this thesis as knowledge that is generated into abstract and general *forms* using specific experiences. Examples of some forms are *empirical equations* [40], *heuristics* ("rules of thumb" [52]), *correlation graphs* [40], *generalisations* [53, 54], or *conceptual chunks* ("types" [49]). Consequently, forms represent defined abstractions of experiential knowledge, which explicate implicit knowledge (i.e. trends) in abstract and general terms.

Abstractions of experience are considered here as the 'bones' of knowledge that can be organised into structures (i.e. 'skeletons' of knowledge) and contain only general characteristics of experiences. Empirical equations, generalisations (e.g. average values, ranges, list of possible values), heuristics (e.g. rules), etc., can be used to provide descriptions to these general characteristics and consequently represent the 'meat' of knowledge used to describe an abstraction.

In ship design, experiential knowledge in the form of empirical equations has been extracted using past designs [40, 42]. For example, from previous ship designs of type *bulker*, the relationship between ship attributes *Cb (block coefficient), Vs (speed) and L (length)* has been quantified by the equation $Cb = 0.968 - (0.269 \times Vs/(\sqrt{L/0.3048})$. Therefore, designers have abstracted the domain of ship design to a conceptual level using characteristics Cb,Vs and L (i.e. design attributes) and described this abstraction as an empirical equation (see Figure 3).

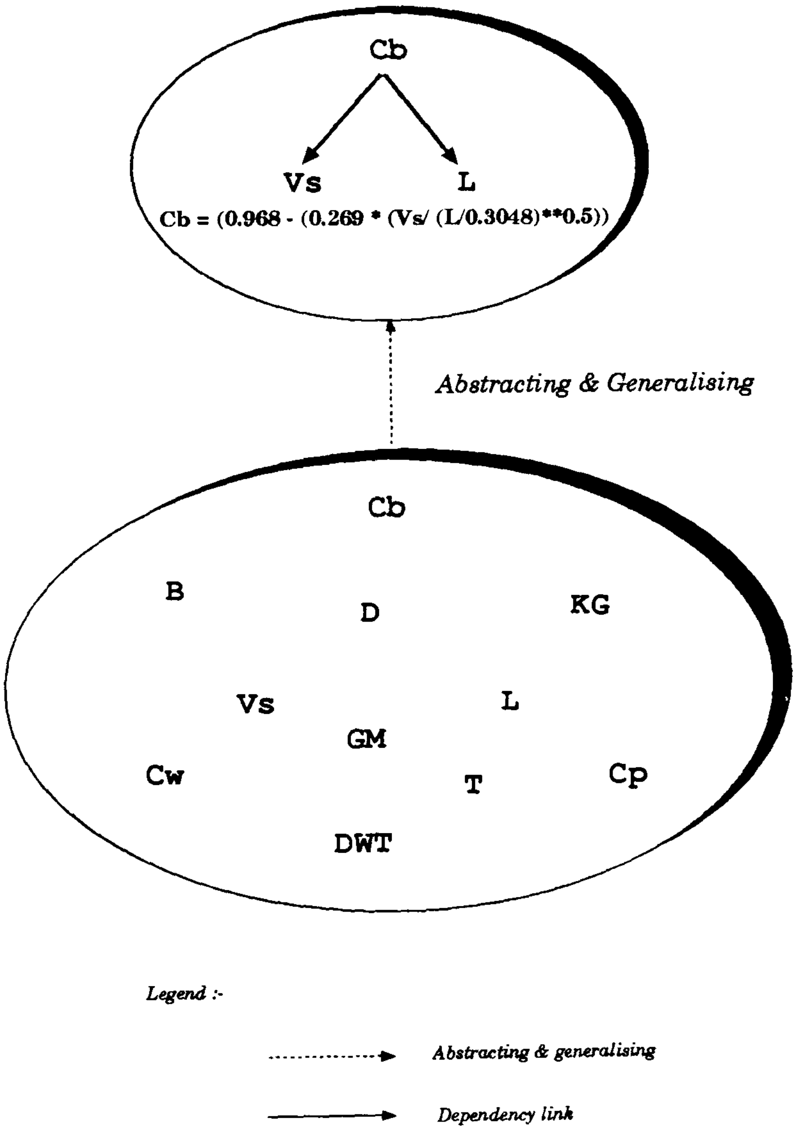


Figure 3: Illustration of an abstraction from the ship domain described as an empirical equation

In addition, relationships between ship attributes can also be represented as heuristics. For example, the heuristic 'if $Cp < 0.85$ then $Cw = 0.878 \times Cp + 0.1733$' represents the relationship between Cp *(prismatic coefficient)* and Cw *(waterplane coefficient)* ship design attributes.

Alternatively, empirical equations can be graphically presented in the form of correlation graphs, which pictorially describe the relationship between, at most, three attributes. For example, Taylor [55] generated a correlation graph that presented the relationship between *weight* and *span* attributes of previous bridge designs of various support methods (i.e. girder, cable-stayed and suspension) (see Figure 4). From this correlation graph it is possible to visualise why designers can generate experiential knowledge in the form of generalisations such as: girder supported bridge designs generally operate under spans of 1000ft; or cable-stayed supported designs operate within a span of 500 to 2000ft. This type of generalisation results in possible value ranges for design attributes (i.e. cable-stayed bridge designs have a design attribute *span* of value 500 to 2000ft).

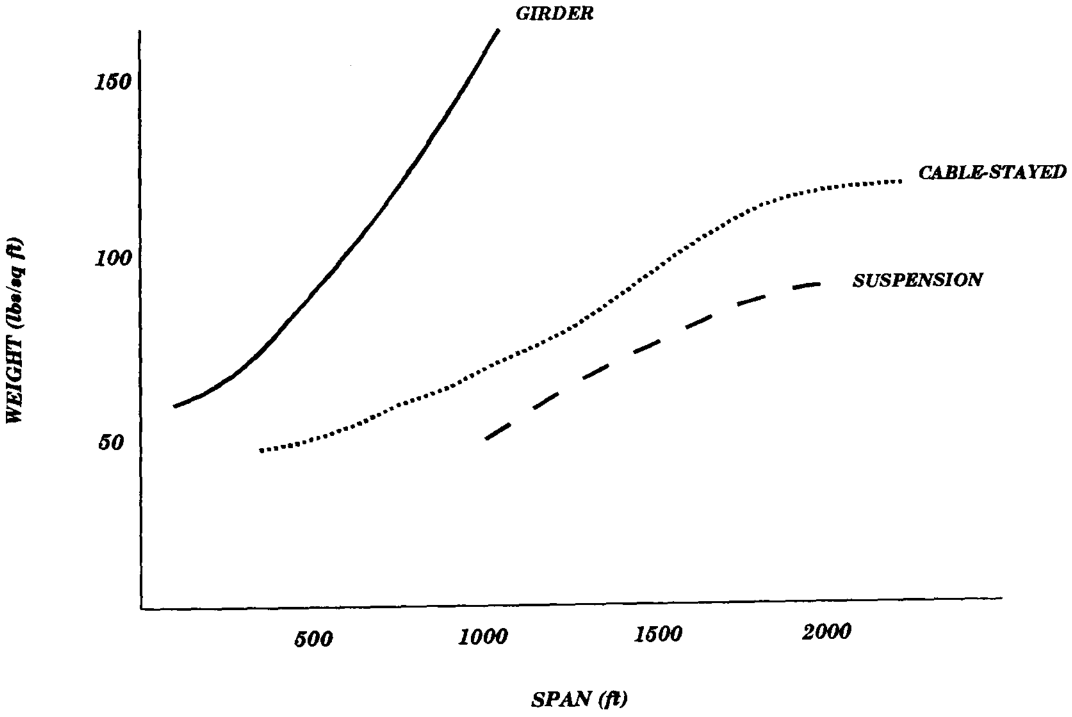


Figure 4: Correlation graph of between attributes of bridge designs [55]

Other examples of designers generating experiential knowledge can be seen in car design [56] and architecture [53, 54].

Pighini et al in 1983 [56] carried out a design study in the domain of car design. This study involved gathering information on 32 city car models, introduced between the years 1954 and 1980, totalling 45 past designs (see Table 2). From these past designs, multiple generalisations of experiential knowledge were generated. For example: 53% of city cars are electrically driven; 36% of city cars have three wheels; the average length of electric cars is 211cm; and 12% of them have only two seats. Table 3 presents some of the resulting knowledge. The importance of this study was that Pighini et al stressed that this knowledge was essential in the identification and definition of the most important requirements of the intended design.

In architecture, designers can represent experiential knowledge in the form of heuristics, which in architectural terminology are called *patterns* [53, 54]. Patterns are 'well-defined chunks of knowledge' and represent defined experiential architectural knowledge that is modular and reusable [53]. For example, the entrance and exit zone to a building (particularly a dwelling) should emphasise the transition between the outside world and the building's environment; a recognised way of achieving this is to change a zone's lighting or paving. Alexander et al [54] carried out the pioneering work of patterns in architecture.

Knowledge of design experiences provides a valuable resource to designers. The importance of experiences is two fold: they can be used to *render* general experiential knowledge or to *supplement* such general knowledge. The latter of these means that when no general knowledge is suitable or can be generated for use in a new design, designers can utilise knowledge of specific experiences. For example, MacLaughlin & Gero support this by arguing that it is only when designers' defined experiential knowledge proves inappropriate that they utilise more specific experiences of past designs [57]. However, certain experiences cannot be used in isolation. To limit the amount of er-

Model	Country of Origin	Date of presentation (year)	Price (in thousand $'s)	Length (cm)	Width (cm)	Height (cm)	. . .	Number of doors	Number of wheels	Maximum velocity (km/h)
City Car	France			170	115	-			3	-
Brio	Italy	1980	2(1980)	180	-	-		2	3	40
Charly	Italy	1974	1.2(1977)	210	125	-		2	3	40
Algol	Italy	1972	-	-	-	-		2	3	-
Sulky	Italy	1974	1.24(1977)	187	116	133		2	3	40
Amica	Italy	1974	1.6(1977)	213	135	122		2	3	55-75
Nova Amica	Italy	1980	3.5(1980)	213	-	-		2	3	60-80
Commuter	Japan	1972	-	175	130	130		2	3	60
Lem	Italy	1974	-	235	135	137		2	3	60
Bella	Italy	1980	-	245	-	-		2	3	-
Citadine	France	1972	-	212	138	155		1	3	50
City Car	Hungary	1959	-	-	-	-		2	3	-
Vespa 400	Italy	1957	-	285	127	125		2	4	85
City Car	Italy	1969	-	-	-	-		1	4	-
City Car	USA	-	-	-	-	-		2	4	30
City Car	UK	1974	-	-	-	-		2	4	-
City Car	Italy	1979	-	-	-	-		2	4	-
City Car	Italy-France	1972	-	180	-	-		2	4	-
Markette	USA	-	-	290	138	-		2	4	40
Caddy	Italy	1977	-	-	-	-		2	4	-
City Car	Italy	1969	1.6(1977)	-	-	-		2	4	70
Milanina	Italy	1972	-	192	134	161		2	4	50
EX 005	Japan	1972	-	233	-	-		2	4	40
City Car	UK	1972	-	195	-	-		2	4	80
Zelz 1000	Italy	1972	-	192	134	161		2	4	40
Log	Italy	1975	-	207	135	-		2	4	80
Comuta	USA	1966	-	203	-	142		2	4	40
BCX II	Japan	1973	-	240	127	-		2	4	20
Isetta	Italy-Germany	1954	-	-	-	-		1	4	85
Urban Car	Italy-Ford	-	-	258	149	138		2	4	-
GM 512	USA	1973	-	-	-	-		1	4	70
City Car	Italy	1972	-	134	-	-		2	4	80

Table 2: Portion of Pighini et al's [56] gathered information of 32 existing city car designs

Item	Possibilities	Average	Percentage
Type of motor	Electrical		53
	Internal combustion		47
Power	Greater than or equal to 2kW		52
	Less than 2kW		48
Dimensions of car	Length	221	
	Width	134	
	Height	146	
Number of seats	Single seat		12
	Two seats		60
	Four seats		28
Number of wheels	Three wheels		36
	Four wheels		54
Maximum velocity	Greater or equal than 55km/h		59
	less than 55 km/h		41

Table 3: Portion of generalisations generated and used by Pighini et al [56]

ror in experimental knowledge, designers normally carry out experiments in batches. From these specific experiments, designers can generate general knowledge. Therefore, designers maximise the utility of experimental knowledge by using general rather than specific (i.e. originating from one single experiment) knowledge. Consequently, experiential knowledge consists of useful specific and general knowledge of experiences.

This section has discussed some of the main components of experiential knowledge that a designer can utilise, i.e. source, trends and forms. However, it has not discussed how designers acquire and maintain this knowledge to provide an up to date and flexible knowledge resource, and how designers manage the complexity of generated experiential knowledge. The following two sections focus on how designers acquire and maintain their experiential knowledge by discussing the purpose of learning in design and how designers manage the complexity of this knowledge by structuring it into viewpoints.

4.2 Learning in Design

The flexibility and utility of experiential knowledge in future design exercises are dependent on associated knowledge reflecting the most recent design practices and it being applicable to the current design problem. To achieve this designers' learn new and general experiential knowledge. This section presents evidence of three processes that maintain the flexibility and utility of experiential knowledge and, when coupled with design, promote the application of experiential knowledge to develop new design solutions. These processes support the utilisation of learning in design and involve: **acquisition, generation and modification**.

4.2.1 Learning

Acquisition, generation and **modification** characterise three learning processes, according to Persidis & Duffy [58]. Figure 5 illustrates the differences among these three processes; acquisition represents the process of receiving new knowledge, generation represents the process of creating new from existing knowledge and modification represents the process of altering existing knowledge.

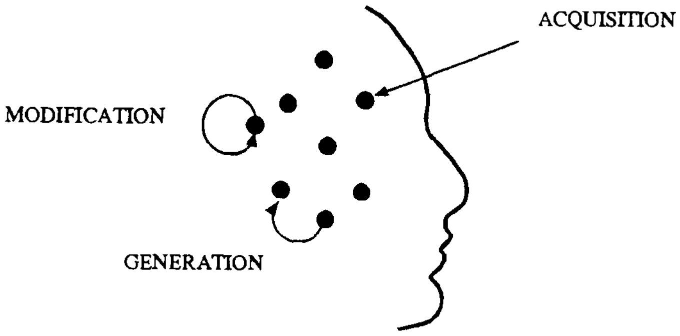


Figure 5: Learning processes [58]

The acquisition of knowledge prevents designers' experiential knowledge from becoming obsolete and provides designers with up to date or new knowledge from which to manipulate existing knowledge. For example, designers can learn about new products and technologies from journals [59] or new problem solving approaches from more experienced designers. In other words, acquisition of knowledge is dependent on knowledge originating from some source external to human memory.

Once designers acquire knowledge, they can then manipulate it. The manipulation of knowledge abstracts and generalises experiential knowledge. This promotes the flexibility of experiential knowledge by removing highly specific details, thereby making knowledge more generally applicable. There are two ways in which manipulation can occur: generation or modification. These two processes are dependent on utilising existing knowledge, i.e. that which has already been acquired. The manipulation of knowledge by generation involves the creation of new from existing knowledge, e.g. a new heuristic (e.g. rule of thumb), relationship (e.g. physical, empirical, statistical and constraining ([60]) [61]), a concept, (e.g. type, class) or structure of concepts (e.g. classification hierarchy, component hierarchy). However, manipulation of knowledge by modification involves altering existing knowledge, for example: by accommodating new knowledge (i.e. acquired or generated) into a previously generated conceptual structure, incorporating exceptions into the applicability of a heuristic or relationship, or extending the description of a concept to encompass a new design. Modification of experiential knowledge is a process that is acknowledged by Himmelblau as a continual process of revision [52].

Evidence of acquisition, generation and modification of experiential design knowledge can be found in, for example:

- *Motard's description of the development of experiential knowledge or what he called professional common sense [9]*
  Motard's 'expansion' of professional common sense maps onto the process of acquisition and his 'organisation and elucidation' to the process of generation and modification.

- *Watson and Watson & Gilfillan's development of empirical equations in ship design [40, 42]*
  In 1962, Watson published a paper detailing the utility of empirical equations in estimating ship dimensions during preliminary ship design [40]; these empirical equations embodied experiential knowledge generated from past ship designs. Then in 1977, Watson & Gilfillan discussed the need for revision of these earlier empirical equations to accommodate changes in ship design [42]. Watson's earlier paper of 1962 illustrates the generation of experiential knowledge whereas the later paper of 1977 illustrates the modification of experiential knowledge.

- *de Siervo & de Leva and Lugaresi & Massa's development of empirical equations in turbine design [62, 63, 64, 65]*
  In 1976 and 1977, de Siervo & de Leva published two papers detailing empirical equations generated for preliminary Francis turbine design [62] and Kaplan turbine design [63], respectively. Then in 1987 [64] and 1988 [65], Lugaresi & Massa published two similar papers that took into consideration the updating of de Siervo & de Leva's earlier empirical equations.

- *Galle et al and Galle & Kovacs' development of patterns in architecture [53, 66]*
  In the domain of architecture, Galle et al [66] presented a number of generalisations of their experiences. Then Galle & Kovacs [53] provided amendments to these generalisations, subject to further design experience.

4.2.2 Coupling Design and Learning

Acknowledging the existence of acquisition, generation and modification requires the acknowledgement that designers can learn during, as a result of designing, and indeed may need to learn in order to design. This thesis considers learning to be a process of acquiring and manipulating knowledge from the specific to the abstract, and design to be a process of defining knowledge from the abstract to the specific. Therefore, although learning and design are distinctly different activities [67], they are inextricably linked [58, 68].

Figure 6 represents the coupling of these activities. The coupling consists of two loops: one looping experiential knowledge and a final design solution, and one looping experiential knowledge onto itself. The first loop links design and learning activities to reflect the iterative approach to design and learning during the development of a design solution, i.e. where the designer continually learns from a design solution at various stages of development and uses learned knowledge to further develop the design solution. The second loop (i.e. that which backs onto itself) represents designers' ability to explore and learn from their own experiential knowledge.

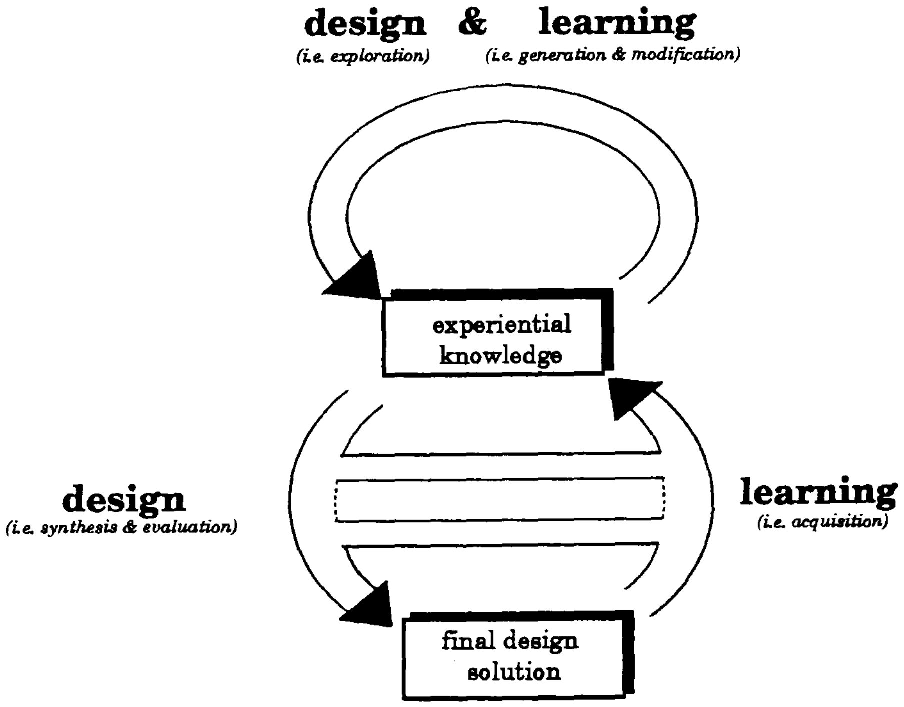


Figure 6: Coupling design and learning activities

The application of experiential knowledge is dependent on the identification, extraction and use of suitable knowledge according to designers' knowledge requirements. To take advantage of the utility of past design experiences, designers learn by acquiring, generating and modifying experiential knowledge. Consequently, it is proposed that these three processes promote the utility of experiential knowledge when coupled with design.

4.3 Viewpoints of Experiential Knowledge

Designers manage experiential knowledge complexity (i.e. implicit and explicit) and variety (i.e. specific and abstract) by structuring knowledge into viewpoints that are inherently conditional on designers' needs for knowledge. This is an important ability that is recognised by MacCallum [51] and Schank [69] in the following:

> "An important feature of humans is the ready ability to re-organise structures [of knowledge]" [51]
>
> "... a self-conscious being. He (the expert) knows when he knows something and he can make observations about what he knows. He can thus alter the memory structures that catalog what he knows if the need arises." [69]

In other words, the ability to generate and modify experiential knowledge when the need arises, i.e. knowing the best way to structure the knowledge and when such a structure is applicable, characterises an expert designer.

Past designs and experiments provide a source of explicit information from which implicit knowledge can be identified and extracted in the form of potentially useful abstract and general experiential knowledge. Consequently, a structure of knowledge contains abstract and general knowledge that is understandable to and usable by a designer.

The importance of and need for viewpoints in design have been frequently argued in design [21, 70, 71, 72, 73] and therefore will not be laboured here. However, to assist in the explanation of how designers apply experiential knowledge in design, a brief discussion of viewpoints is necessary.

4.3.1 Types of Focus

Designers can focus on their experiences, to abstract experiential knowledge, in many ways. Such types of focus depend on designers' needs. For example, designers might generate viewpoints by focusing on geometrical, numerical, spatial, electrical, electronic, mechanical, functional, structural, behavioural, compositional or taxonomic knowledge. These types of focus are well recognised in literature; however no attempt, as yet, has been made to formulate them into some sort of classification.

The following list presents such a classification by categorising these examples of foci into a number of types which a designer might employ.

- **Aspects:** relate to knowledge that is for example geometrical, numerical, spatial in nature.
- **Disciplines:** relate to the scientific domain with which the designer is interested, e.g. electrical, electronic, mechanical, civil.
- **Factors:** relate to knowledge such as function, structure and behaviour.
- **Associations:** relate to the associations between elements of experiential knowledge, i.e. compositional (i.e. 'part-of'), taxonomic (e.g. 'group-of', 'kind-of'), relational (e.g. 'connected-to', 'above', 'below').

Viewpoints of experiential knowledge originating from past designs are commonly used in design. For example, the general concept of a car consists of a body, engine, wheels and a chassis [68]. This example originated from specific experiences of cars when focusing on structural factors with a compositional association.

Figures 7, 8 and 9 show further examples of viewpoints in design. Each figure represents a different viewpoint generated from a source of experiential knowledge consisting of specific past car designs (represented as a sphere). Figure 7 represents three associations (i.e. 'part-of', 'group-of' and 'kind-of') whereas Figures 8 and 9 represent only one (i.e. 'kind-of'). Inherent in each viewpoint is the designer's need for knowledge or perspective [74] and is signified in these figures by the symbolic eye. That is, based on designers' needs, designers structure the associated experiential knowledge to reflect what they deem interesting and/or important depending upon the chosen focus (i.e. aspects, disciplines, associations, etc.). For example, the resulting viewpoint of employing a 'group-of' association to the attribute *speed* and then *miles per gallon, (m.p.g.)* of cars can be seen in the 'group-of' viewpoint of Figure 7; alternatively the 'part-of' viewpoint results in the partitioning of car knowledge to identify two classes of components, *bodies* and *engines*; and the 'kind-of' viewpoint results in the identification of useful classes of *cars* and *engines*, (e.g. *sports, family* and *in-line, vee, box* respectively), based on a degree of similarity between past *car* designs.

Figure 8 helps to explain how designers can view experiential knowledge of cars from a numerical aspect. The figure illustrates this experiential knowledge as dependency networks at various levels of abstraction. The levels of abstraction are the result of designers removing design attributes and associated dependencies, and generalising the relationships between attributes.

Figure 9 illustrates an abstraction and generalisation of the car domain from a geometrical aspect. The resulting viewpoint shows two geometrical concepts, which encompass four specific geometrical designs.

Design literature frequently refers to viewpoints. However, it is not proposed here to discuss and formalise all the types of foci employed by designers. The purpose of this section has been to emphasise the diversity of potentially useful viewpoints and to classify some of the more prevalent examples of focus into a classification.

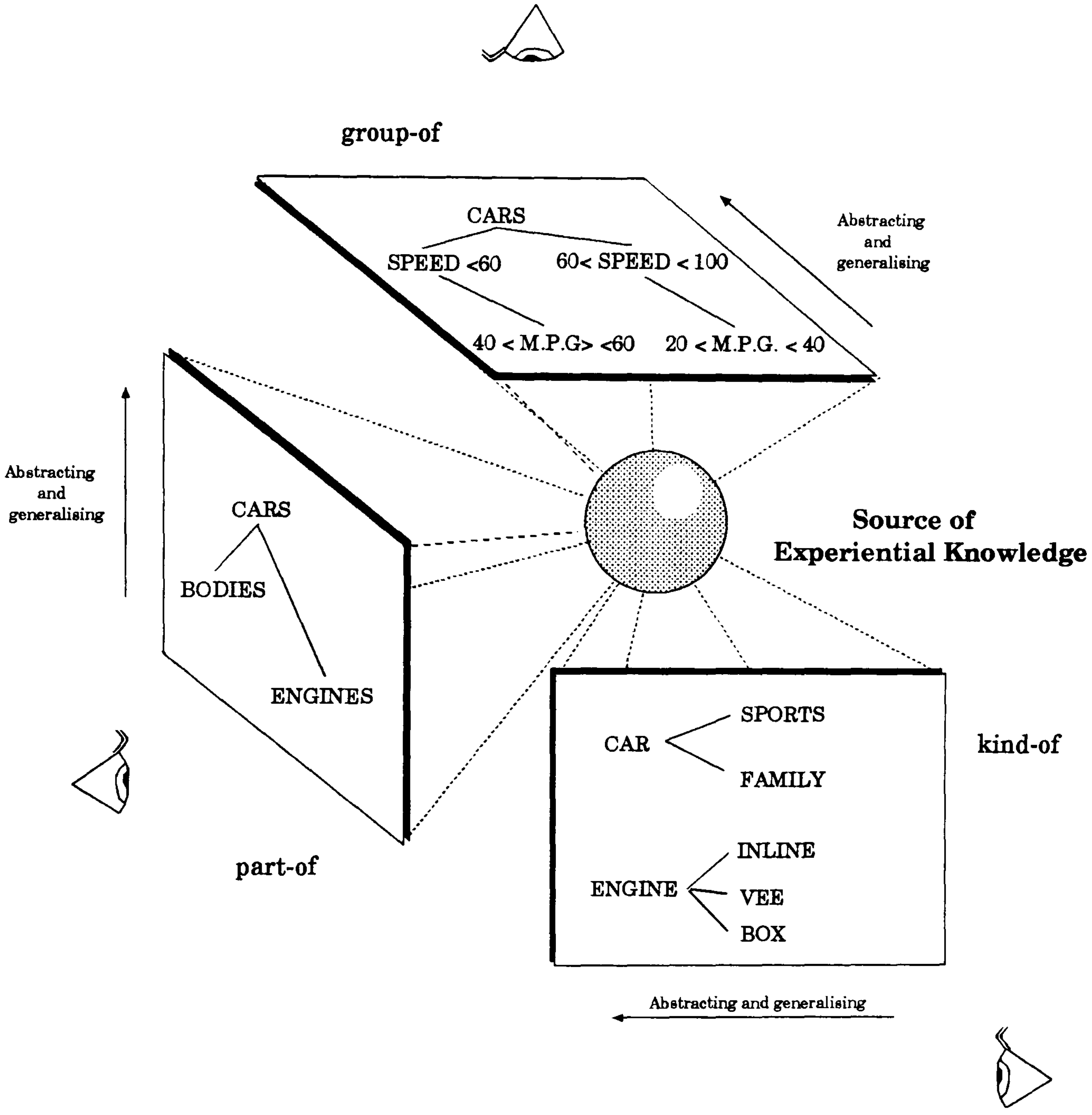


Figure 7: Three viewpoints of experiential knowledge of cars

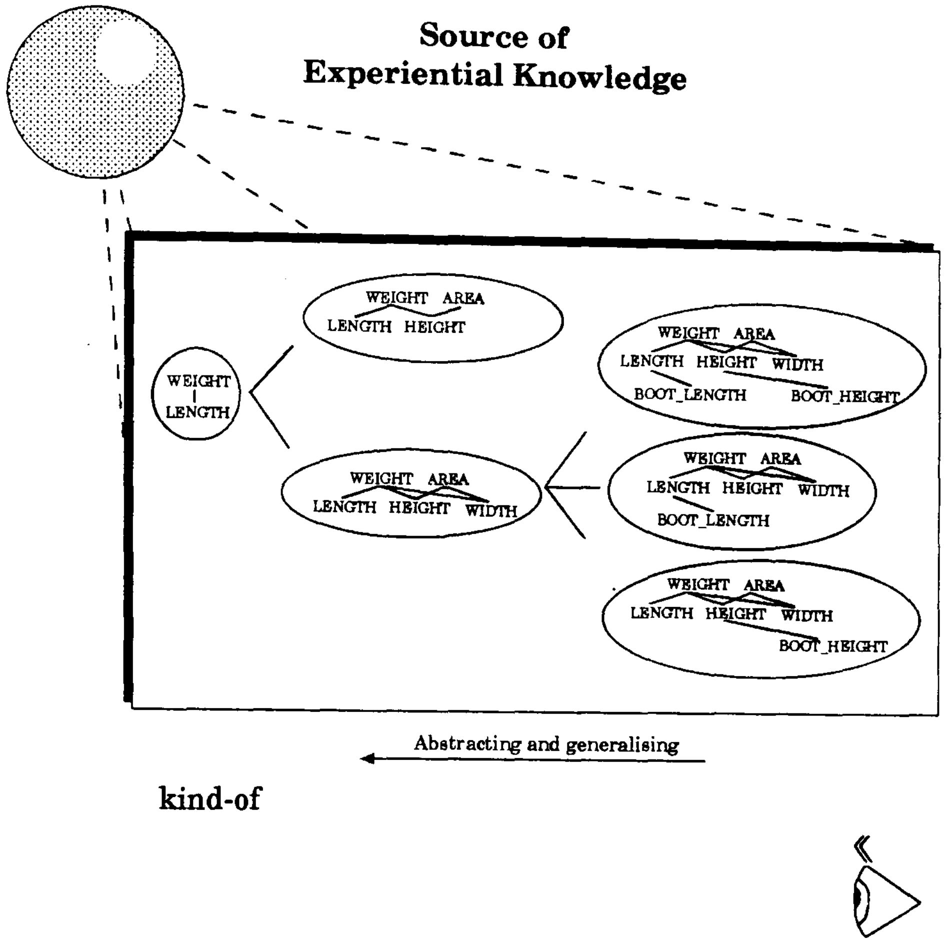


Figure 8: 'Kind-of' associative viewpoint of cars from a numerical aspect

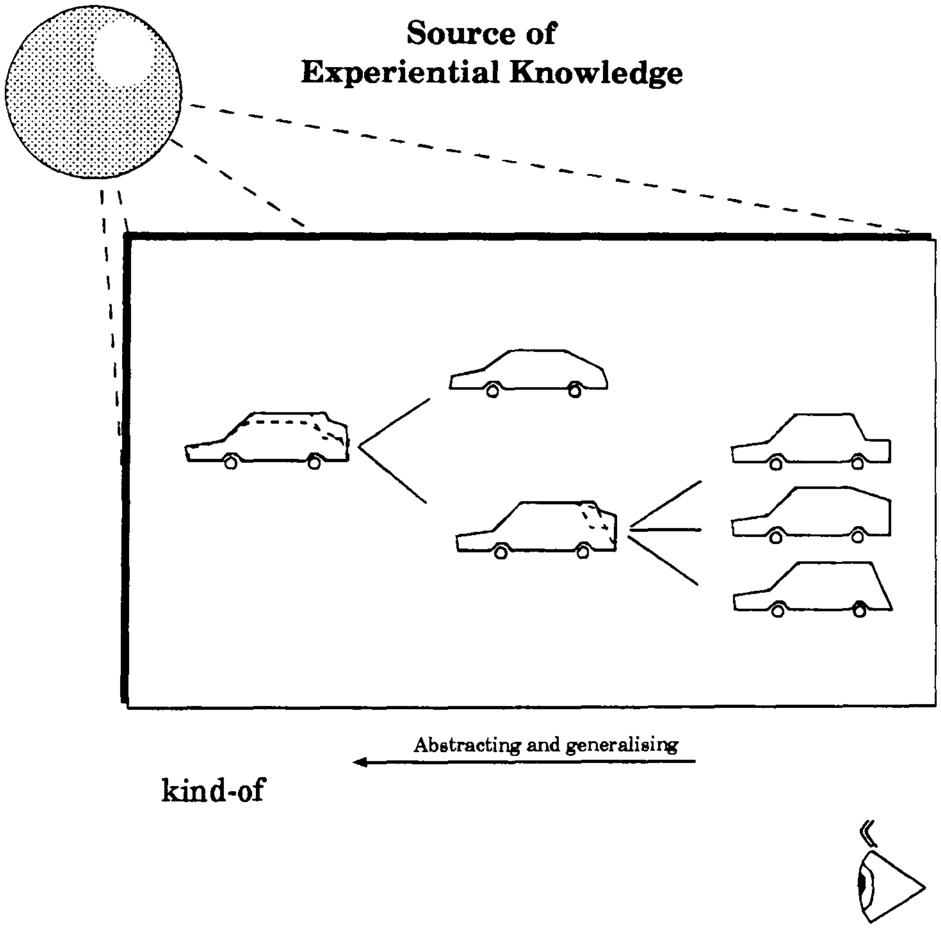


Figure 9: 'Kind-of' associative viewpoint of cars from a geometrical aspect

4.3.2 Types of Viewpoints

As well as taking a single focus of experiential knowledge to generate a viewpoint, designers can change focus from within a viewpoint. Therefore there exists two types of viewpoints, i.e. **single** and **nested** viewpoints. Single viewpoints incorporate knowledge of past designs by focusing on a single group of characteristics. Nested viewpoints incorporate knowledge of past designs by changing focus from one group of characteristics to another, as many times as necessary. Thus, designers construct nested viewpoints from sequential single viewpoints. The following describes these types of viewpoints by using examples from design literature.

Single

Single viewpoints are the result of a single focusing. For example, Suh [71] provides two examples of single viewpoints of lathes. Figure 10 illustrates a focus on functional factors, whereas Figure 11 illustrates a focus on structural factors.

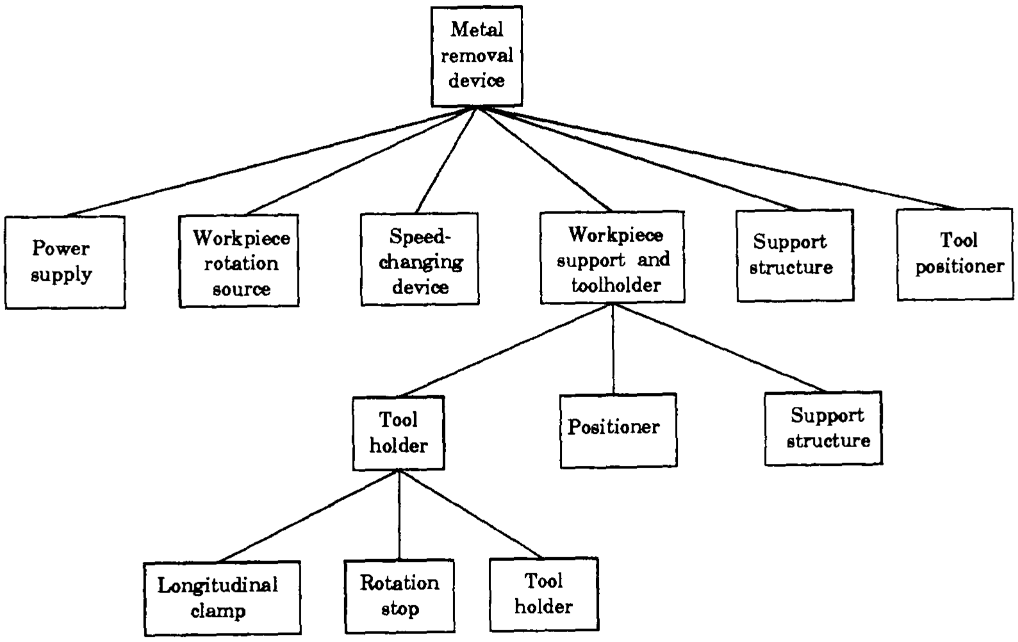


Figure 10: A functional viewpoint of lathes [71]

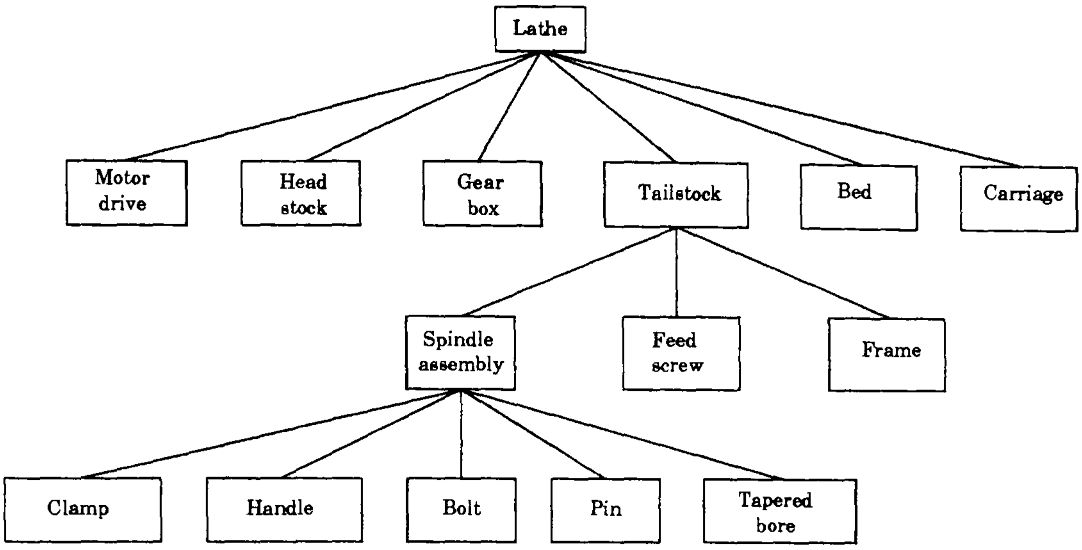


Figure 11: A structural viewpoint of lathes [71]

Nested

In Oxman's discussion concerning the role of high level (abstract) concepts in "non-routine" design [39], a knowledge structure is illustrated (see Figure 12). This structure consists of:

- precedents (i.e. past designs),
- functional concepts (i.e. concepts generated by focusing on the function of past designs), and
- typological concepts (i.e. concepts generated by focusing on spatial aspects).

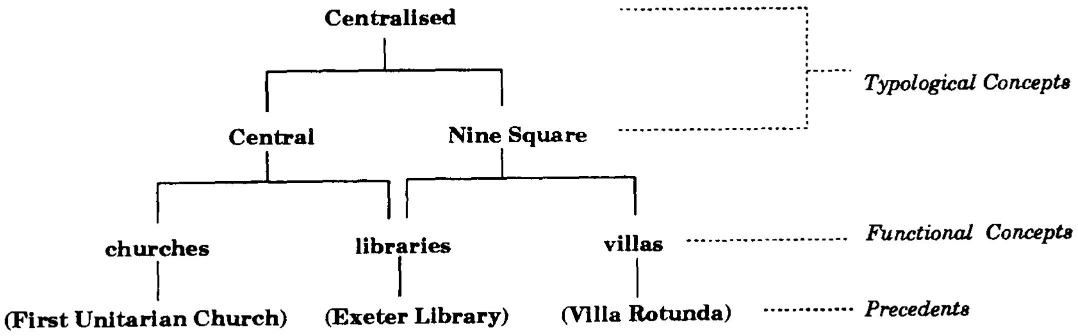


Figure 12: Example of a nested viewpoint, from a knowledge structure by Oxman [39]

In effect, this knowledge structure reflects a nested viewpoint consisting of two foci: one focusing on spatial aspects and the other focusing on functional factors of past designs.

Taylor's comparative study of bridges [55] illustrates another example of a nested viewpoint in design. Figure 13 shows some of the results from this study, where the following bridge attributes were focused upon: method of support, weight and span. In this example, the concept of a nested viewpoint is used to identify regions in the graph characteristic of bridges adopting a specific method of support, i.e. girders, cable-stayed or suspension.

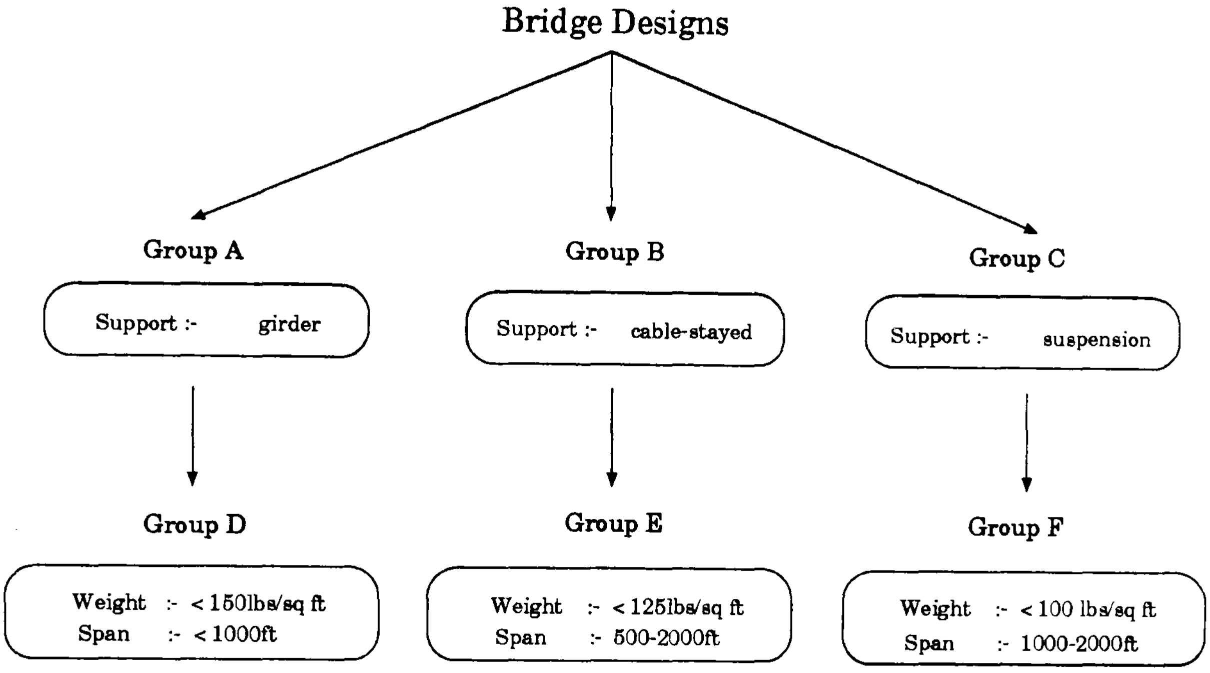


Figure 13: Example of a nested viewpoint, derived from Figure 4

Further evidence of nested viewpoints can be seen from the generation of design trends in turbine design. For example, Figure 14 illustrates an example of a nested viewpoint existing in Francis turbine designs [62]. Focusing upon the design date for a number of Francis designs, designers generated three groups: Group A incorporating designs from the years 1960-1964, Group B from the years 1965-1969 and Group C from the years 1970-1975. Focusing on the design attributes *specific speed* and *design head* for each of these groups, designers identified a further three groups; the resulting groups are represented in Figure 14 as Groups D, E and F. These groups of designs are describable by generating individual empirical equations that represent the trend between *specific speed* and *design head* attributes of associated designs. Figure 14 shows these equations with their respective group of designs. Similar nested viewpoints have been generated for Kaplan turbines [63, 75], reversible pump-turbines [76], pump-turbines [77] and bulb turbines [78].

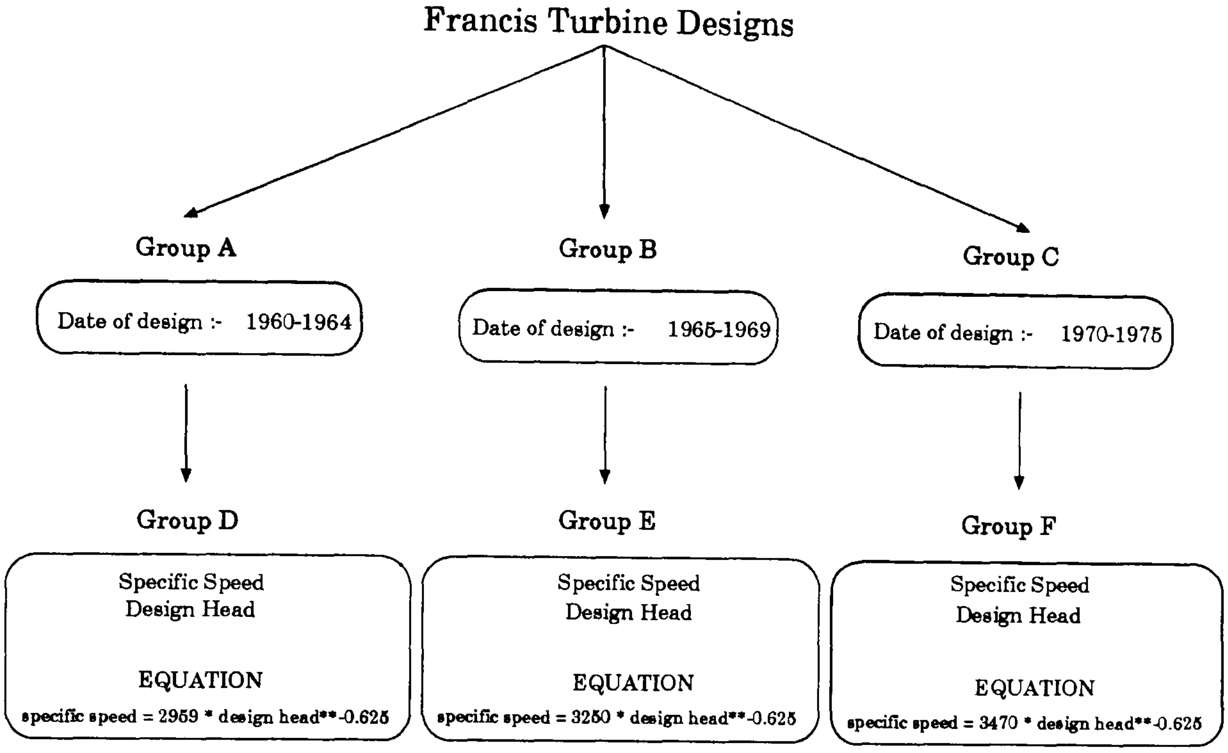


Figure 14: Illustration of a nested viewpoint consisting of date of design and then specific speed and design head with generated associated empirical equations, (derived from [62])

Similarly, examples of nested viewpoints can be seen in ship design. For example, see Figure 15. Empirical equations can sometimes be applicable only within a specific group of designs. For bulker ship designs with a Cp value less than 0.85, the empirical equation for calculating Cw is $(0.878 \times Cp) + 0.1733$; alternatively when the Cp value is between 0.85 and 1.00 the applicable equation is $(0.6 \times Cp) + 0.4$ [34].

4.3.3 Summary

This section has emphasised that designers manage the complexity and variety of experiential knowledge by structuring knowledge into viewpoints, according to particular types of focus, as governed by their design needs. In addition, it has defined examples of focus types to help describe some concerns that designers may concentrate on when

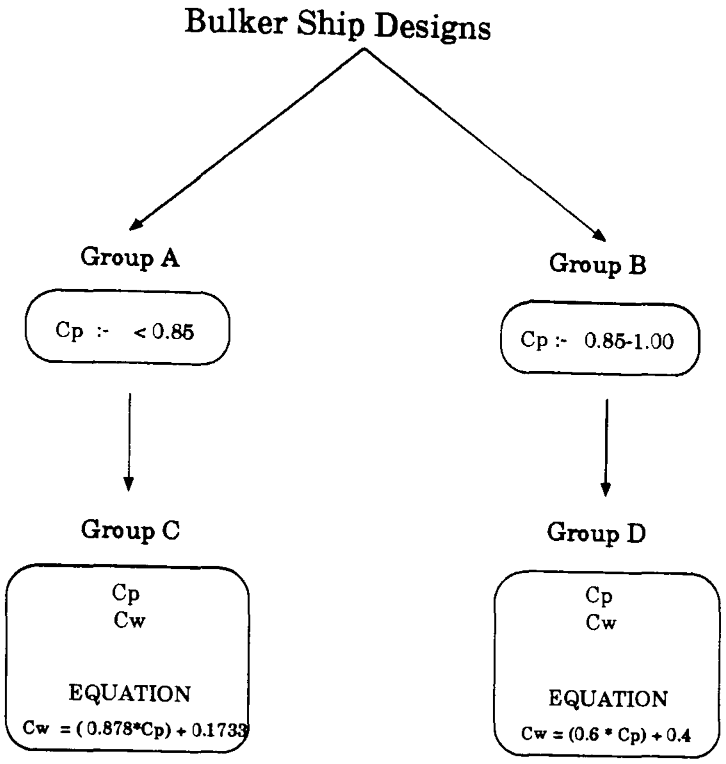


Figure 15: Illustration of a nested viewpoint consisting of Cp, then Cp and Cw with associated empirical equations

they generate viewpoints of experiential knowledge. Designers not only generate viewpoints using a single focus; they can change focus within viewpoints. Consequently, this section introduces two types of viewpoints (i.e. single and nested) that designers use to simplify the complexity and variety of experiential knowledge, by giving examples from design literature.

4.4 Computer Aided Support for Utilising Experiential Knowledge

Based on the identification that designers execute learning activities (see Section 4.2) and the emphasis that such activities play an important role in the effective application of experiential knowledge in design, this thesis investigates how computers support the utilisation of experiential design knowledge.

Early computer aided support for design [79, 80, 81, 82, 83] tended to focus on analytical aspects (e.g. optimisation [84], finite element analysis [85]). In other words, they supported design judgement. Gradually systems have emerged that focused on the generative aspects (e.g. synthesis [68], configuration [86]), i.e. design creation. However, these generative systems represent experiential knowledge in the form of pre-defined structures prepared by a knowledge engineer's preconceptions of designers' needs for knowledge. For a more thorough critical review see the following chapter.

This thesis argues the importance of knowledge **application** as opposed to knowledge **regurgitation** in CAD systems. Current support by CAD promotes knowledge regurgitation and involves the direct reuse of knowledge as originally acquired and represented. Knowledge engineers identify required knowledge and represent this knowledge for its subsequent reuse in design. Consequently, these CAD systems do not manipulate (i.e. generate or manipulate) knowledge; they reuse knowledge as it was originally stored. However, encountered design problems may be the same, similar or completely different to those previously encountered. Alternatively, designers' knowledge requirements can change. Therefore the pre-compiled knowledge, used by current CAD systems, can become inappropriate for designers' new knowledge requirements. To overcome the redundancy of pre-compiled knowledge and promote the effective use of knowledge by meeting designers' requirements, future CAD systems should be able to support the flexible use of knowledge, i.e. the *application* of knowledge. This requires 'knowledge transformation', i.e. the automatic manipulation of knowledge from that originally acquired and represented. Knowledge *application* involves the identification and modification and/or structuring of knowledge suitable for solving new problems. Therefore, to utilise knowledge more effectively, future support by CAD should be able to promote knowledge *application* thereby making knowledge potentially more useful in subsequent design problems.

The discussion, in this chapter, on how designers apply experiential design knowledge has emphasised that designers need to use up to date and appropriate general experiential knowledge. To achieve this, designers learn from past experiences. They acquire new knowledge to keep up to date with current practices, and generate and modify existing knowledge to make it more generally applicable. As experiential knowledge is complex, designers structure generated knowledge into appropriate viewpoints from which they can generate applicable experiential knowledge.

The *application* of experiential knowledge is dependent on the knowledge requirements of a designer. Designers cannot fully predict their knowledge requirements. Therefore, the approach of using solely a knowledge engineer to extract required knowledge, to construct viewpoints of experiential knowledge and to represent this in a computer is considered an inappropriate one for the development of future CAD systems.

Future design systems should be able to effectively utilise all viewpoints of experiential knowledge according to the knowledge requirements of designers and not just the preconceived viewpoints that are defined by knowledge engineers. Consequently, the following list states a number of requirements for future CAD systems:

- support the utilisation of multiple sources and forms of experiential knowledge.
- support the coupling of design and learning activities.
- support the utilisation of multiple viewpoints of experiential knowledge that reflects the needs of designers.

Although computer support has been developing in design evaluation and synthesis, computer support for design exploration has remained relatively unsupported. Consequently, computational support for utilising experiential design knowledge is limited. This thesis argues that a computer system supporting the exploration activity has the advantage of bypassing the preconceptions of a knowledge engineer and promoting the effective utilisation of experiential knowledge, according to individual designer's knowledge requirements.

4.5 Summary

This chapter has investigated the application of experiential knowledge in design by addressing *how* designers apply experiential design knowledge. The chapter presented a framework of experiential knowledge consisting of the main components designers utilise during design. Table 4 summarises this framework under the headings of component and meaning.

Component	**Meaning**
Sources	Origins of generalised experiential knowledge, e.g. past designs and design experiments.
Trends	Implicit experiential knowledge.
Forms	Defined abstractions of experiential knowledge, e.g. heuristics, empirical equations, generalisations.

Table 4: Framework of experiential knowledge utilised by designers

The flexibility and utility of experiential knowledge are dependent on it being up to date and reflecting designers' own particular needs. Designers achieve this by learning experiential knowledge (i.e. by acquiring and generating new knowledge and modifying existing knowledge) and structuring knowledge into viewpoints according to their own particular requirements. In other words, designers increase the utility of experiential knowledge by coupling learning with design, and manage the complexity of experiential knowledge by generating viewpoints. They use many types of focus to generate viewpoints of experiential knowledge, which this chapter classifies. In addition, this chapter introduces two types of viewpoints (i.e. single and nested) and uses these viewpoints to explain how designers can change their focus from within a viewpoint. Finally, the discussion on how designers apply experiential design knowledge concludes by defining a number of computer aided design system requirements, which this thesis adopts as being necessary for the support of experiential knowledge in design.

5 Critical Review of Computer Support for the Utilisation of Experiential Knowledge

Warman, in 1978, predicted that the utilisation of experiential knowledge would be a "major component of future CAD systems" [87]. Based on the importance of experiential knowledge utility, presented in Chapter 3, this thesis agrees with Warman's conjecture.

As a result of discussing how designers apply experiential knowledge, Chapter 4 presented a number of requirements for effective experiential design knowledge support. These requirements are to support (a) the utilisation of multiple sources and forms of experiential knowledge, (b) the coupling of design and learning activities, and (c) the utilisation of multiple viewpoints of experiential knowledge that reflect the needs of designers. This chapter uses these identified requirements as reference points from which to critically review how well computers support the utilisation of experiential knowledge, to identify the deficiencies in current support and thereby emphasise the need for improvement.

This chapter discusses CAD systems and, in particular, the approaches and techniques they employ to utilise experiential knowledge. The success of these systems is dependent upon the effective **representation** and **processing** of this knowledge. Therefore, this review focuses on these two aspects. Section 5.1 discusses the development of computer-based approaches that have facilitated the storage, retrieval and utilisation of experiential design knowledge, namely conventional databases and knowledge based formalisms. The trend in these developments has been to increase the richness of represented knowledge (e.g. to represent multiple types of focus). However, this is only one requirement of effective experiential knowledge utilisation. These approaches are deficient in that (a) through time the richness of represented experiential knowledge will become out of date, and (b) the represented knowledge reflects knowledge engineers' preconceptions of designers' knowledge needs. In other words, these approaches provide little support for the generation of viewpoints tailored to suit particular designer's needs (i.e. the processing of this knowledge). Consequently, Section 5.2 presents an overview of CAD systems that employ techniques to process experiential knowledge in design, i.e. systems that learn generalised experiential knowledge to keep it up to date and avoid the need for a knowledge engineer to define the experiential knowledge to be represented. This section consists of three subsections, according to CAD systems that employ data analysis, machine learning and the integration of these two techniques. Section 5.3 concludes the discussion on computer support by summarising the main deficiencies of support for experiential knowledge utilisation in CAD.

5.1 Modelling and Representation of Experiential Knowledge

One of the major motivating factors in CAD research has been to enrich computer-based representation of knowledge and thereby provide designers with assistance in the utilisation of knowledge during design [38, 73, 88, 89, 90, 91, 92, 93]. Experiential knowledge is a vastly rich type of knowledge, which designers can utilise during design, as discussed in Chapter 4. Therefore, this section critically reviews CAD approaches that are specific to representing experiential knowledge according to designers' knowledge requirements. It discusses the developments of some of the main computer based approaches to storing and utilising experiential knowledge by considering conventional databases and knowledge base formalisms. Although the development of CAD approaches has attempted to increase the content of represented knowledge, this only reflects the trend in attempting to partially satisfy the requirements of effective experiential knowledge support. These approaches are still deficient in supporting the learning of experiential knowledge and representing experiential knowledge according to designers' particular knowledge requirements.

5.1.1 Conventional Databases

Prior to the use of computers, designers stored experiential knowledge in paper base form, e.g. *paper based drawings, rules, log books, reports, diaries, memos* and *tables.* However, with the advent of the computer, the development of databases promised more efficient data storage and retrieval mechanisms [94] than these traditional storage approaches.

Although computer support of experiential knowledge in the form of conventional databases provided 'low level' support (i.e. the storage of and access to knowledge), they provided little assistance in interpreting or reasoning with that knowledge or its implicit meaning, i.e. 'high level' support. For example, Yazaki and Sumiyoshi [95] in their information retrieval system used information of ship model tank tests, sea trials and propeller model tank tests (i.e. experiments) to aid in basic numerical ship design, estimation of propulsive performance and propeller choice. From input data, detailing required characteristics and/or dimensions of the new ship design, the system helped designers obtain optimum dimensions and hull forms by identifying and ranking similar ship designs. In this application, specific experimental information from a numerical aspect was used to estimate, by interpolation, the performance of the new design. In other words, the system did not utilise, let alone generate, *general* experiential knowledge. However, the Interactive Ship Design System (ISDS) [96] utilised general experiential knowledge originating from information of past ship designs and experimental model ships. Designers initiated the process of ship design by defining design requirements from which the system helped to generate a tentative set of dimensions using experiential knowledge in the form of correlation graphs. The experiential knowledge utilised in these two systems represented well-documented specific information of experiments but limited general knowledge. In particular, these early CAD systems represented static storage locations of knowledge. In other words, they provided no assistance in the interpretation of specific experiences to generate further general experiential knowledge.

One application of a conventional database in CAD that does attempt to extract general knowledge from specific experiences is the STRUPLE system [14]. STRUPLE, a system that supports structural design, automatically identifies design vocabulary from a source of past designs and subsequently utilises such knowledge to develop a new design. Using a relational database of past structural design solutions, STRUPLE automatically identifies and ranks past designs that are similar to a new design. From these identified similar past designs, the system calculates the frequency of the design vocabulary used to describe each design and uses that which occurs most frequently to develop the new design solution. Although STRUPLE identifies knowledge of similar past designs to be utilised in developing a new design solution, it does not store this knowledge for subsequent use in another design problem. In addition, the system's learning capability is guided directly using knowledge of the new design and not by designers' knowledge requirements.

Generally, these types of CAD systems represent limited knowledge richness. For example, they do not represent the concept of viewpoints as presented in Chapter 4.2. The following sections focus on more descriptive design representation formalisms that attempt to support more of the 'richness' of experiential knowledge.

5.1.2 Knowledge based formalisms

Various formalisms (e.g. rules, logic, semantic networks) for representing declarative (e.g. past designs and design experiments) and procedural (e.g. design processes) knowledge have evolved from work carried out in artificial intelligence and knowledge based systems [97]. However, this section reviews a number of knowledge based approaches to representing (declarative) experiential knowledge.

One of the prevalent formalisms for representing experiential knowledge is frames. Frames provide a schema for representing declarative knowledge of a concept or object described by attributes and values. (Frames can also be used to represent procedural knowledge by demons or methods). They can be structured into a hierarchy or network of nodes connected by relations. A basic relation in frames is one of refinement or 'kind-of'. This relation indicates that the frames at the top of a hierarchy are more abstract than those lower down in the hierarchy. Consequently, frames are useful for representing *classes* and *subclasses*. *Class* is a term used to denote a generalised set of concepts of a certain type, e.g. engines, people, reports. A *subclass* is a specialisation of its parent class and may *inherit* knowledge from its parent. Therefore, a subclass represents a narrowing of a concept but also an expansion of the knowledge about that concept.

Frames present a natural and efficient way to categorise, structure and represent experiential knowledge, and provide a powerful approach for analysing problems, facilitating rapid prototyping and modification, and reducing knowledge duplication. They also support multiple relations/links between concepts, which system users can view, and accommodate incremental additions to the taxonomy. However, their support to the overall engineering design process is significantly limited. Namely:

- only pre-defined viewpoints of experiential knowledge are available. Designers cannot view implicit knowledge of experiences, unless a knowledge engineer or a designer explicitly defines such knowledge.
- knowledge of specific experiences can only be inherited and the taxonomies reflect an abstract-to-specific (top-down) ordering. There are no inherent mechanisms for automatically creating and modifying knowledge structures or for generalising from actual design experiences (bottom-up), which is considered here a necessary facility for coupling learning and design activities.

Frames are a general representation technique. However, they have a number of limitations in their application within design. To enhance their suitability to the particular requirements of design, work in the field of Int.CAD has developed systems and approaches directed at the use of past design knowledge. Some of the more notable are

Prototypes [25, 98, 99, 100, 101], Concept Libraries [58, 68], Case-based design systems [102, 103, 104, 105, 106, 107], Precedent-based design systems [108] and Neural Network-based design systems [109, 110, 111].

Prototypes [25, 98, 99, 100, 101] are schemas for representing generalised experiential knowledge and, according to Gero, can include in their definition parameterised design descriptions, interpretations (goals and requirements), a vocabulary of design elements (such as beams and columns in structural engineering) and knowledge relating the interpretations and the vocabulary. Design prototypes consist of information concerning function, structure and behaviour, and store knowledge that is relational, qualitative, computational and contextual in nature. They can represent knowledge at multiple levels of abstraction, and their effective use in design is dependent on a search for one that reflects the detailed level of the problem specification. However, this level is compositional rather than taxonomic; prototypes do not represent general knowledge that refers to general design concepts (see discussion on NODES in next paragraph). In addition, prototypes do not represent information associated with individual past designs and their levels of abstraction are defined by knowledge engineers.

In contrast to Prototypes is the Numerical and Object-based DESign (NODES) system [58, 68]. NODES provides modelling and synthesis support during the creation and modification of a design solution. It provides designers with the ability to represent abstractions of a design solution using concepts stored in a Concept Library. Concept Libraries provide a source of past design instances and concepts, which a designer can use to synthesis to new design. Concepts represent 'chunks of experiential knowledge' commonly used in a design domain (e.g. car and pump design) and are organised into taxonomic hierarchies. A knowledge engineer specifies the structure of these hierarchies; therefore, their structure is manually defined. However, their knowledge content can be automatically modified by machine learning techniques (see Section 5.2) to update and augment the knowledge of the design domain with new designs. In other words, NODES induces generalised knowledge (e.g. value ranges and nominal features of past design abstractions) from newly defined design solutions. Thus, although NODES uses a pre-defined knowledge structure, it can automatically update the content of this structure to reflect newly created design solutions and thereby exhibit a degree of learning capability. In addition, NODES supports analysis by incorporating the DESIGNER system [34]. DESIGNER is a CAD system for numerical design. It supports the definition of the design domain using numerical characteristics and relationships between these characteristics. In other words, it builds a model of the design domain using pre-defined experiential knowledge in the form of empirical equations. The process of numerical design can be rather complex as the designer is required to remember applicable relationships, influences and consequent alterations caused by a change in a characteristic's value. Therefore, to obtain a design solution that satisfies design goals and requirements, DESIGNER assists the designer to carry out extensive calculations and numerical alterations.

Case-based design systems [102, 103, 104, 105, 106, 107] can help designers identify and modify specific past designs (cases) to suit a new design problem. These systems employ case-based reasoning [112] to identify and retrieve appropriate cases as promising solutions to new design problems. They can use taxonomic hierarchies to represent knowledge of past designs. However, these hierarchies are preset and cannot be modified. Therefore, although these systems represent experiential knowledge for its subsequent utilisation in design, they do not support the automatic generation of viewpoints of experiential knowledge according to the needs of a designer. Consequently, the search for a past design case is dependent on a search through a preset taxonomic hierarchy of past designs. If this hierarchy does not reflect designers' needs, the identification of a suitable past design case would be difficult.

Precedent-based design systems are different from case-based design systems. They represent past designs as *design stories* [108] rather than taxonomic hierarchies. These design stories, developed in the domain of architectural design, represent past design ideas and consist of three parts: issues (i.e. general design objectives), concepts (i.e.

general solutions that achieve design objectives) and form (i.e. specific past designs that illustrate these general solutions). The design story representation provides an alternative formalism for representing experiential knowledge that designers or knowledge engineers have directly extracted from specific past designs. Consequently, this new representation scheme is heavily dependent on a manual analysis of textual annotations of past designs and elicitation of knowledge to be represented as design stories.

Neural network-based design systems [109, 110, 111] represent knowledge in interconnected independent processing units, i.e. nodes. Each interconnection has a weighting associated to it, which can be either fixed or varied. A weighting specifies the degree of interconnection strength between nodes and can have a positive, negative or zero value. Positive values specify excitatory interconnections between units, adding to interconnection strength. Negative values specify inhibitory interconnections between units, reducing interconnection strength, and zero values specify interconnections not utilised, i.e. no strength between units. The units and their interconnections define a network; see Figure 16 for an example of a neural network [113] showing nodes as circles and interconnections as links between nodes.

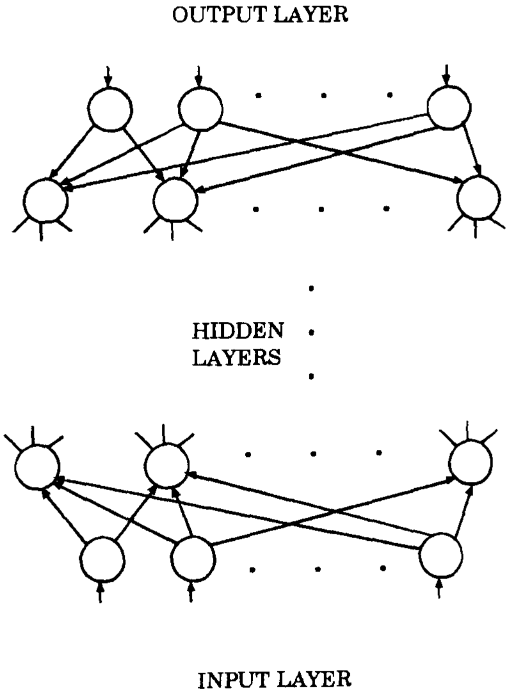


Figure 16: Neural Network Model showing input, output and hidden layers [113]

Layers can exist within a network. A network consists of an input and an output layer and any number of hidden layers. Inhibitory (negative weighting values) or excitatory interconnections (positive weighting values) can exist between units in different layers. However, units in the same layer either have no interconnections (zero weighting values) or inhibitory interconnections (negative weighting values).

These systems can learn knowledge from specific design experiences (i.e. past designs or experiments) and represent the resulting learned knowledge throughout the whole of the network. Since neural networks represent knowledge across its nodes and links, designers cannot scrutinise knowledge. Therefore, the application of neural networks as a representation formalism for experiential knowledge can be described as a 'black box of experiential knowledge'. In addition, the preparation of a network, i.e. choosing the number of input and output units is preset by the system builder before the process of learning. These units can refer to portions of experiential knowledge. Consequently, a network-based design system makes assumptions of the knowledge to be utilised and generated.

The artificial intelligence and knowledge based systems research areas have developed powerful reasoning capabilities and knowledge representations but provide little support for large knowledge bases or efficient storage and retrieval of data [114]. Since the mid eighties considerable effort has gone into the integration of database and artificial intelligence technology to provide a more efficient and effective mechanism for data modelling and knowledge representation [115, 116, 117, 118, 119]. Today, techniques such as object-oriented databases [120] and theoretical, logic based models [114] provide the field of CAD with various tools. These approaches however suffer from the same criticisms as presented above for the conventional databases and knowledge based formalisms.

From the above, the main criticism of existing approaches is that they represent knowledge according to knowledge engineers' preconceptions of designers' knowledge requirements. These approaches do not accommodate the representation of designers' own particular viewpoints of experiential knowledge.

5.2 Processing Experiential Knowledge

Computer approaches to modelling and representing knowledge have helped to promote the utilisation of experiential design knowledge by, for example, supporting the access and retrieval of specific and general knowledge. However, CAD still requires support for promoting more flexible utilisation of experiential knowledge according to a particular designers' needs. Chapter 4 proposed the flexibility and utility of this knowledge to be achieved by coupling design with three learning processes: acquisition, generation and modification. Therefore, to assess comprehensively computer support for the utilisation of experiential knowledge, this section reviews how computational support processes knowledge according to these processes.

This section critically reviews support for experiential knowledge acquisition, generation and modification from information of past designs and experiments. The review addresses three main fields of study that are applicable to the processing of experiential knowledge: **data analysis**, **machine learning** and **integration of data analysis and machine learning**. These techniques are presented in order of maturity, i.e. data analysis being the most mature, machine learning that attempts to overcome the limitation of data analysis, and the integration of these techniques to benefit from the advantages of both. The discussion highlights the utility of these techniques by explaining a number of design system that employ these technique to utilise experiential knowledge.

5.2.1 Data Analysis

Data Analysis, a subject of Statistics, is the study and application of methods that analyse a set of information and provide a hypothesis based on the analysis of this presented information set. An information set consists of information about a group of instances; each instance consisting of a set of variables and associated values. Data analysis provides designers with the ability to explore experiential knowledge by investigating, identifying and quantifying relationships between variables that describe a set of instances.

There are many data analysis methods available for analysing instances. For example: principal component analysis [121], factor analysis [121, 122], cluster analysis [121, 123], discriminant analysis [121], univariate regression (multiple regression) and multivariate regression [124]. These methods can be divided into those which analyse *"interdependency"* of variables (e.g. principal component analysis and factor analysis), those which analyse *"dependency"* of variables (e.g. multiple and multivariate regression) and those which *"group"* and *"classify"* instances (e.g. cluster analysis and discriminant analysis respectively) [124].

"Interdependency" methods aim to transform the variables of instances to a new set of variables that are uncorrelated and arranged in decreasing order of importance. In other words, the aim is to reduce the dimensionality of the set of instances. "Dependency" methods aim to describe the dependency a number of variables have on some other variable(s). "Group" methods separate the instances into clusters of near proximity. "Classifying" methods assign instances to existing clusters based on some measure of similarity.

Designers have employed data analysis in the domains of ship design [40, 42, 125] and turbine design [62, 63, 75, 76, 77, 78], to explore past designs and generate knowledge concerning the trends that exist within past designs. In these design applications, the

the set of variables and associated values of the instances are design attributes and associated values.

Yoshikawa & Koyama [126] analysed a set of past ship designs by employing *cluster analysis* and *principal component analysis.* They used clustering to classify ship designs and mapped the results onto a space defined by the first two principal components to described the complete set of ship designs. Their work provides some interesting results because the resulting groups of past designs map onto generally accepted groups of past designs, i.e. design domain types such as ferries, bulkers, etc. Unfortunately, their analysis was purely exploratory and consequently lacked any attempt to generate some general knowledge to be used for design.

In contrast, work of Watson and Watson & Gilfillan, as presented in Chapter 4.3, had more purposeful intentions; they employed data analysis using regression techniques to generate empirical equations for preliminary ship design. In this way they quantified the dependency of various ship design attributes (e.g. length,L, and speed, Vs,) on other attributes (e.g. block coefficient, Cb) as equations (e.g. $Cb = 0.968 - (0.269 \times Vs/(\sqrt{L/0.3048}))$. Similar work exists in the domain of turbine design [62, 63, 75, 76, 77, 78].

Designers have applied data analysis to learn general knowledge (e.g. empirical equations). However, designers have carried out this separately from the design process. Consequently, the processes of design and learning have been considered as separate activities. This thesis firmly disagrees with this view and reflects this in one of the requirements for effective computer support of experiential design knowledge, i.e. the need to support the coupling of design and learning activities within a single environment. Williams [127] and Aihara & Sugawara [128] recognised the need for this coupling and demonstrated this by using data analysis to extract of implicit knowledge from a store of explicit information. Williams analysed and stored ship propulsion data to support design optimisation. His work involved the collation of significant information for hull form and propulsion, the analysis and reduction of this information to numerical form, the storage of information in a database, and the programming of procedures for utilising such information. As a result, designers could use Williams' database to support a number of activities: initial ship design, comparative judgement and statistical analysis of resistance data. Williams adopted the coupling of design and learning using experimental information, not past design information. Similarly, Aihara and Sugawara [128] discuss a ship design system consisting, among other things of 3 databases referring to experiential information: the results of model experiments, hull offset data of ships and models, and structural arrangement data of ships. This system uses model experiments in conjunction with desired conditions of a new design to estimate the required power of the new design. In addition, designers can generate (by statistical analysis) experiential knowledge, using results of model wave making experiments, and use this knowledge to estimate the wave making resistance value of the new design. These examples illustrate the utility of automatically accumulating experiential knowledge from experiments.

The most common data analysis technique used to generate useful general knowledge for numerical design is regression. Unfortunately, the application of this technique, for the generation of experiential knowledge relating to past designs, has normally been distinct from the design process - unlike the knowledge relating to experiments. Consequently, the application of this technique to past designs has not been supportive of the coupling of design and learning activities while developing a design solution.

The inability for techniques such as principal component analysis and cluster analysis to generate useful general knowledge from specific information, as illustrated earlier by Yoshikawa & Koyama [126], severely limited their utility in exploring application domains. The subject of Machine Learning attempted to overcome the failings of these data analysis techniques by helping to extract more useful general knowledge, and is the topic of the next section.

5.2.2 Machine Learning

Machine Learning (ML) is the study of the development and application of techniques that support learning by computational means. It is a large field, consisting of many developed techniques (e.g. learning from examples, learning by analogy, learning by discovery, etc.), within which many writers have provided possible classifications of available techniques. For example: Carbonell, Michalski & Mitchell [129] classified machine learning techniques according to the underlying learning strategy (e.g. rote learning, learning by analogy, learning from examples), the representation of knowledge (e.g. decision trees, production rules, taxonomies) and the application domain (e.g. agriculture, chemistry, medical diagnosis); Reich [37] classified machine learning techniques according to a two dimensional framework addressing the "level" at which learning takes place (i.e. knowledge or symbol level) and the method of learning (i.e. analytical methods that depend on a prior understanding of a domain, or empirical methods that extract information from experience); and Kocabas [130] classified machine learning techniques according to the "level" at which knowledge representations (e.g. rules, frames, predicate logic, semantic networks, classifiers, neural networks) and learning methods (e.g. abstraction, similarity-based generalisation, conceptual clustering, genetic algorithms) can be expressed, i.e. knowledge, symbol or device level.

A number of applications that adopt these techniques have supported the exploration of experiential knowledge in design. The following reviews CAD systems that extract knowledge from experiments and past designs using ML techniques.

Using information from experiments

McLaughlin & Gero [131] use Pareto optimisation to identify groups of past building design experiments that conform to certain performance criteria, and use inductive learning methods to generate generalisations of these groups by identifying the relationship between design attributes (e.g. wall type, sunshade size and the size and construction of windows). For example, for the group of experiments that are Pareto optimal, according to three criteria (i.e. maximise daylight factor, minimise mean summer temperature and maximise mean winter temperature), sixteen experiments describe designs that are constructed with a 300mm cavity brick wall, are unplastered, have small sunshade and medium sized windows. This approach assists in the exploration design activity to identify and extract knowledge suitable for use in design. However, as in similar work [132, 133], it does not directly feed generated general experiential knowledge into design. In other words, these approaches/systems incorporate only the process of learning experiential knowledge and do not demonstrate its subsequent utilisation during design.

The importance of experiential knowledge in the form of heuristics generated from experiments is useful in the domain of motor oil design [134]. These heuristics relate motor oil ingredients to measures of related oil performances. Their use is desirable in either determining the blend of ingredients for the required performance or determining the expected performance based on a proposed blend of ingredients. Kamal et al [134] discuss and demonstrate the automatic generation of these heuristics using information from experiments to identify important design attributes (ingredients/composition and performance), to extract relationships between these design attributes (to validate or modify existing heuristics) and generate classification rules using these relationships. Their approach employs inductive learning methods to generate knowledge. In addition, Kamal et al argue that this helps to increase designers understanding of the design problem during the early stages of design.

The SPRED-1 system utilises the neural network technique and aids designers in the preliminary design of space grid structures. It uses this technique for four reasons:

1. to avoid detailed structural analysis of a design,
2. to estimate performance of design parameters and the design (as a whole),
3. to optimise a design, and
4. to estimate required design parameter modification.

The system consists of three subsystems: the prediction, evaluation and control subsystem [109]. The prediction subsystem uses previously analysed structures to "train" a neural network and consequently predict, for a new structural design, a set of preliminary design parameters (e.g. maximum internal forces, deflections, etc.) from a set of conceptual design parameters (e.g. plan dimensions, structural depth, number of different member types, support conditions, etc.). In other words, the prediction subsystem learns associations between conceptual and preliminary parameters and utilises these associations to help develop a preliminary design solution. This subsystem provides the designer with results equivalent to that of running a structural analysis (i.e. finite element analysis) program and therefore avoids the need for detailed, time consuming analysis. The evaluation subsystem uses the input (conceptual) and output (preliminary) parameters used in the prediction subsystem and an expert defined classification to "train" another network. The expert defined classification describes the performance of the output parameters according to one of five classes: "excellent", "good", "fair", "bad" and "very bad". Once "trained", this subsystem evaluates the performance of each preliminary design parameter of the new structural design. Next, a second neural network of the evaluation subsystem estimates the overall performance of the new design by combining the performance of each of its preliminary design parameters. The control subsystem then optimises the design, according to a parameter chosen by the designer, and then estimates required changes to specific parameters.

SPRED-1 explicitly avoids detailed structural analysis by using the results of previously analysed structural designs, and has shown that this less analytically exact analysis still generates accurate results. Although this system uses experiential knowledge originating from design experiments, it does not identify and extract implicit general experiential knowledge, which is open to scrutiny by a designer. Therefore, whereas SPRED-1 has supported and demonstrated the utility of experiential design knowledge by coupling design and learning activities, it does not extract generalised experiential knowledge that is understandable to a designer.

Using knowledge of past designs

The NODES system induces experiential knowledge from past designs. In this system, designers assign past designs (i.e. known instances) to a concept and the system determines the conceptual description of the concept using the information associated with the assigned past designs. This process represents a technique from ML (see Chapter 6) called *supervised concept learning*, which uses the information of known members of a concept to generate associated conceptual descriptions. The NODES system uses a ML mechanism called Maximally-Specific-Conjunctive generalisation [135] to learn conceptual descriptions. In this mechanism, attributes that describe a concept can exist only if they exist in all the associated past designs or subconcepts, and values associated with these attributes are generalised according to the Closing Interval and Climbing Double Generalisation Tree rule [136]. Figure 17 illustrates these rules in the car domain; the changes in the *speed* attribute value for the *family car* concept is achieved by incorporating *car-design2* and by applying the Closing Interval rule, and the change in the *part* attribute value for the *family car* concept from *inline-4* to *inline* is the result of incorporating *car-design2*, using knowledge that *inline* engines consists of two types (i.e. *inline-2* and *inline-4*) and applying the Climbing Double Generalisation Tree rule. The Closing Interval rule generates a range/interval for a concept's numerical attribute value to signify the maximum and minimum attribute values that associated past designs or subconcepts may fall within. The Climbing Double Generalisation Tree rule generates a conceptually nominal attribute value for a concept from the nominal attribute values possessed by its members. A conceptually nominal value refers to the nearest general component that associated past designs or subconcepts possess.

BRIDGER is a system directed at the synthesis of cable-stayed bridges [137] and implements the ML technique called *unsupervised concept learning*, alternatively known as concept formation. This technique is carried out by BRIDGER's subsystem called ECOBWEB and is used to generate two hierarchies of past designs: a *synthesis* and a *default hierarchy.* A synthesis hierarchy is one that incorporates "existing" attributes

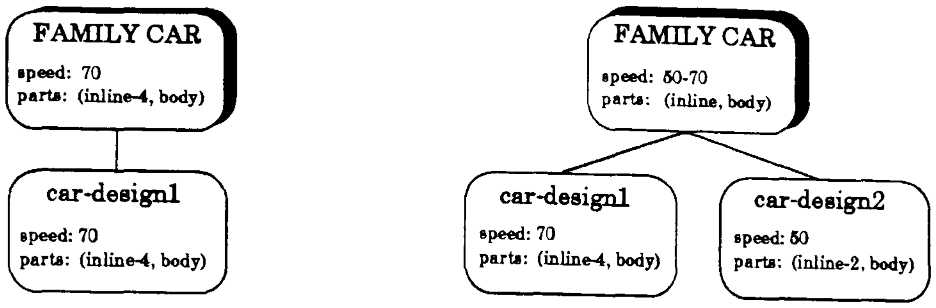


Figure 17: Example of applying the Closing Interval and Climbing Double Generalisation Tree rules

(i.e. those used to describe the past design). A default hierarchy is the label given to a hierarchy generated using "derived" attributes (i.e. those generated by the designer to describe the proportionality of past designs e.g. length to breadth ratio, depth to width ratio). The contents of these hierarchies are generalisations of past designs, which BRIDGER helps designers use to synthesis a new bridge design. It does this by supporting not only case-based reasoning but also extends this concept to support the automatic selection of abstract design classes from past design cases. Thus, BRIDGER supports the use of design cases and design abstractions. A limitation of BRIDGER, is that it uses all the attributes and their associated values of the past design cases to generate an all encompassing class structure, i.e. a structure containing generalisations of all available past design information. In addition, the representation of the classes and design cases are restricted to simple attribute-value pairs. BRIDGER cannot represent knowledge of the past designs such as functionality, structure, or behaviour. For example, if the types of past design cases are composites (i.e. consisting of parts) then BRIDGER cannot represent a sub-part structure, i.e. it cannot represent 'part-of' relations in the generated classification structure.

The neural network approach can acquire and use knowledge of past designs to generate feasible design solutions [138]. An example of such an application, uses design specifications, attributes and associated values of past designs to "train" a neural network. Once "trained", the neural network, when presented with a subset of instantiated specifications and design attributes (i.e. a partially complete new design), can estimate *a posteriori* probabilities for the values of the remaining unknown attributes of the new design. In effect, this neural network is attempting to capture the relationships between the design variables (i.e. the design requirements and design attributes). This application supports the exploration and synthesis activities by capturing the relationships between design variables (i.e. design specifications and design attributes) and by finding suitable values of design variables based upon given input values. Unfortunately, neural networks cannot explicate identified relationships or general knowledge associated with specific knowledge. Therefore, although usable, the application of neural networks hide experiential knowledge.

Similarly Coyne, Newton & Sudweeks [110, 111] show how the application of neural networks can generate room designs. This work uses information of 50 previous designs of rooms (i.e. kitchens, living-rooms, dining-rooms, bedrooms and bathrooms) to "train" a neural network. The information details the content of each room, e.g. the contents of a particular kitchen design could be a double sink, a fridge, etc. Table 5 lists the total contents that describe the past designs used to "train" the neural network. From a partial definition of a room design (i.e. designs described using only a few of the contents mentioned in Table 5), the "trained" neural network can predict the remaining contents. This application illustrates how a neural network can learn general descriptions of room designs and consequently use this to complete (i.e. synthesise) partial room designs.

1 Bath	2 Blinds	3 Bookcase	4 Carpet
5 Coffee Table	6 Comfy chair	7 Carpet	8 Desk and chair
9 Dining chairs	10 Dining table	11 Cupboards	12 Double Bed
13 Double sink	14 Drapes	15 Doorways	16 Ferns
17 Fridge	18 Lamp	19 Dresser	20 Lounge Suite
21 Medium Floor	22 Open plan	23 Large floor	24 Shower
25 Single bed	26 Sink	27 Rug	28 Small Table
29 Sofa Bed	30 Small floor	31 Tiled/timber floor	32 Television
33 Very large floor	34 Wardrobe	35 WC	36 Window
37 Worktops			

Table 5: Total contents used to describe past designs of rooms [111]

The application of neural networks in design supports the coupling of learning and designing activities. The modification of weights in a neural network reflects the modification of experiential knowledge. Unfortunately, the main limitation of neural networks in the utilisation of experiential knowledge is their inability to make implicit knowledge explicit, i.e. to extract experiential knowledge. Therefore, neural networks are not able to generate experiential knowledge that reflects designers' knowledge requirements and is understandable to and usable by designers.

5.2.3 Integrated Data Analysis and Machine Learning

KEDS (Knowledge-based Equation Discovery System) [139] is a model-driven system that uses results from experiments to quantify relationships between design attributes and performance attributes as empirical equations. In other words, KEDS generates equations that comprise a model of engineering experiments. KEDS automatically renders models by integrating machine learning and data analysis techniques. First, it uses conceptual clustering to identify groups of similar experiments. Then, it uses regression, guided by a number of pre-defined equation templates (acting as possible equation structures), to fit regression equations to associated groups. To maximise the quality of generated equations, KEDS manipulates the membership of specific experiments to more appropriate groups and re-renders the equations associated with these modified groups. Consequently, KEDS optimises the quality of rendered models. Table 6 shows three examples of some equations generated by KEDS. The left hand column details the specification, as attribute value ranges, of groups identified. The right hand column details a description, as empirical equations, of associated groups.

KEDS processes information of experiments by generating and modifying experiential knowledge in the form of empirical equations. Its approach is completely automatic. In other words, designers have no control over the process of generating and modifying empirical equations, other than the control they can have over the definition of possible

Group Specification	Group Description
$0.81 < X < 1.29$	$n = 1.65CR + 7.16PI - 28.08X + 49.92$
$X < 0.92$	$n = 1.99CR + 8.92PI + 25.82$
$7.1 < CR$	$n = 1.69CR + 6.11PI - 31.08X + 53.84$

Table 6: Examples of some generated equations from KEDS

equation templates. Consequently, this approach suffers from restricting the exploration process to a specific set of equation structures and does not support the more flexible generation of equations guided by a particular designer's requirements. In addition, KEDS does not support the looping of generated experiential knowledge back to design (i.e. the subsequent utilisation of generated experiential knowledge).

5.3 Discussion

The purpose of this critical review has been to identify the general deficiencies of computer support that utilises experiential design knowledge and thereby emphasise the need for improved support. The following list details the results of this review according to the three main support requirements identified in Chapter 4.

- **Utilisation of multiple sources and forms of experiential knowledge**
  Although some systems have been developed to support the utilisation of multiple forms of experiential knowledge (e.g. NODES which supports the utilisation of concepts, generalisations and empirical equations), none integrate the utilisation of knowledge originating from multiple sources, i.e. experiments or past designs.

- **Coupling of design and learning activities**
  Increasingly more support for the coupling of design and learning activities is being recognised and incorporated in CAD systems, e.g. BRIDGER, NODES and the neural network-based design systems (e.g. SPRED-1). These systems, other than the neural network-based, support the identification and extraction

of generalised knowledge for its subsequent utilisation during design. They can explore experiences to identify and extract knowledge for synthesis or evaluation design activities. Few CAD systems exist that support all three design activities. BRIDGER extracts knowledge from past designs to generate generalised conceptual hierarchies that are used to identify applicable knowledge to be used in a new design. This results in the generation of a number of possible design solutions that can then be analysed and assessed to identify designs requiring redesign from which acceptable designs can be generated. NODES extracts generalised descriptions of known design concepts and utilises this knowledge to help synthesise a new design and evaluate how well this new design meets its goals/requirements by incorporating the DESIGNER system. Some systems support the exploration of knowledge for the purpose of evaluation: e.g. SPRED-1, which capture relationships between the descriptions of structural designs and the results of analysis (e.g. finite element analysis) to evaluate the performance of a new design. Others only support the exploration of knowledge: e.g. KEDS, which extracts knowledge from experiments to generate empirical equations; McLaughlin & Gero, who support the identification and extraction of generalisations of past designs; and Kamal et al, who support the generation of general heuristics from experiments. Alternatively, some support the exploration of knowledge for its subsequent use in synthesis: e.g. STRUPLE, which extracts knowledge from past designs to generate general design vocabulary; Ivezic & Garret, who capture the relationships between design attributes from a set of past designs and utilise these relationships to help find suitable design attribute values; and Coyne et al who similarly capture relationships between the contents of a set of room designs and utilise these relationship to help predict the contents of new room designs.

- **Utilisation of viewpoints according to particular knowledge requirements**

  The utilisation of multiple viewpoints of knowledge, to increase the 'richness' of knowledge available in design systems, has been a very important driving force in the development of CAD systems. However, these viewpoints are generally defined by knowledge engineers and as such reflect pre-defined assumptions of what knowledge will be needed by a designer. What is required is the generation, representation and utilisation of viewpoints that reflect designers' knowledge requirements, as and when needed, and thereby free designers from the pre-defined, potentially restrictive, viewpoints that are prepared by knowledge engineers.

In summary, this chapter has reviewed some of the main approaches which computationally support the utilisation of experiential knowledge in design. It has outlined the development of representational support for experiential design knowledge from conventional databases through to knowledge based formalisms and discussed a number of techniques which have been employed to process experiential design knowledge. Consequently, the following summarises two main criticisms that have been identified in this chapter.

- No system supports any means of abstraction or generalisation of knowledge to suit the particular needs of individual designer. Most systems embrace pre-defined structures, reflecting only specific perspectives of experiential knowledge.

- Few systems support the comprehensive coupling of design and learning activities with which generalised experiential knowledge can be acquired and developed, and directly fed back into design for synthesis or evaluation purposes. Systems capable of learning generally support limited automatic generalisation of explicit knowledge to represent implicit knowledge explicitly.

6 Customised Viewpoint Approach

There are two philosophies of computer support, which reflect different extremes of thought concerning the role of computers in design. The "design assistance" philosophy considers a CAD system as a designers' subordinate [16] whereas the "design automation" philosophy considers it as a designers' substitute [140], as illustrated in Figure 18.

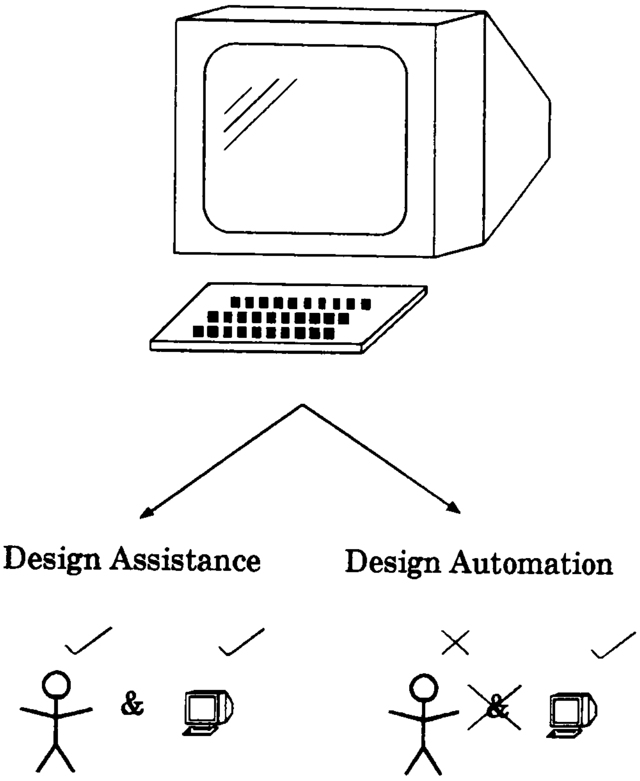


Figure 18: Two philosophies of Computer Aided Design Support

This thesis adopts the philosophy of "design assistance", since it argues this to be the more generic philosophy of the two. That is, it is believed that building systems with the goal of providing "design assistance" can encompass aspects of "design automation" (e.g. in the form of design optimisation or analysis) along with the ideas that systems can act as designers' colleagues and thereby compliment designers' abilities. Therefore, this thesis advocates that the ultimate goal of automating the design process ignores the potential of coupling the capabilities of a designer and computer, and is therefore more fundamentally restrictive.

An important feature of "design assistance", presented here, is the ability to support the effective utilisation of experiential design knowledge. This is to be realised by building systems that make experiential knowledge more flexible, according to the particular needs of designers. It is not possible to completely predict designers' knowledge requirements since each designer has different knowledge needs at different times and for different reasons; thus the requirements of a single designer change throughout the design process. Therefore, a "design assistant" should be able to utilise experiential knowledge to reflect the variety of designers' needs, and support knowledge *application* to particular design problems.

This chapter presents a new approach to utilising experiential knowledge called '**customised viewpoint"** (CV), which is proposed to compliment the "design assistance" philosophy. The main idea behind this approach is that it generalises experiential knowledge directly from specific experiences, according to designers' knowledge needs, and subsequently utilises this knowledge in design. Consequently, designers' knowledge requirements, rather than knowledge engineers' perspectives of knowledge requirements, directly govern the effective utilisation of experiential knowledge.

The utility of experiential knowledge in design is dependent on it being:

- indicative of the most recent design trends,
- appropriate to designers' needs, and
- applicable in a new design problem.

Therefore, tools that support the effective utilisation of experiential knowledge in design should be able:

- to automatically update and evolve experiential knowledge,
- to support the automatic generation of appropriate viewpoints of experiential knowledge, and
- to support the utilisation of applicable experiential knowledge in design.

Update and evolution of experiential knowledge

The continual development of experiential knowledge in a system ensures that the (specific and general) knowledge available to a designer is up to date. Therefore, a system should automatically acquire new knowledge and modify existing general knowledge to accommodate new experiences. In other words, update and evolution involves the development of experiential knowledge, in the light of new experiences, to maintain a knowledge resource that reflects the most recent design trends.

Generation of appropriate viewpoints

Designers require different viewpoints of knowledge at different times for different reasons. Chapter 4.2.1 proposed a number of focus types that designers employ to portion experiential knowledge and structure knowledge into viewpoints. Current CAD approaches try to support the utilisation of experiential knowledge by increasing the number of viewpoints available to a designer; for example by supporting the representation of geometrical, numerical, functional or structural portions of knowledge.

Traditionally, knowledge engineers (or system developers) define viewpoints. Consequently, these viewpoints reflect knowledge engineers' perspectives of designers' knowledge requirements. However, if designers' *actual* knowledge requirements differ from those *perceived* by a knowledge engineer then the result is a mismatch between a knowledge engineer's imposed perspective and a designer's required perspective. A mismatch of perspectives occurs when:

- the pre-defined viewpoint is irrelevant (e.g. a viewpoint from the geometrical aspect is required but only numerical aspect is available);
- the knowledge associated with a relevant pre-defined viewpoint is irrelevant, i.e. is the knowledge focused upon does not include the knowledge required. For example, when a structural viewpoint of a car defining the component concepts to be a body concept and engine concept but knowledge concerning the chassis or wheels is not available; or
- the form of represented knowledge is unsuitable for the intended purpose. For example, when a viewpoint contains generalisations as average attribute values rather than ranges, or generalisations rather than empirical equations).

Consequently, this traditional approach limits the ability of systems to support the effective utilisation of experiential knowledge. It presents designers with pre-defined viewpoints that may be of little use, if the imposed perspectives do not map onto those required by designers. Figure 19 illustrate this 'pre-defined viewpoint' (PDV) approach.

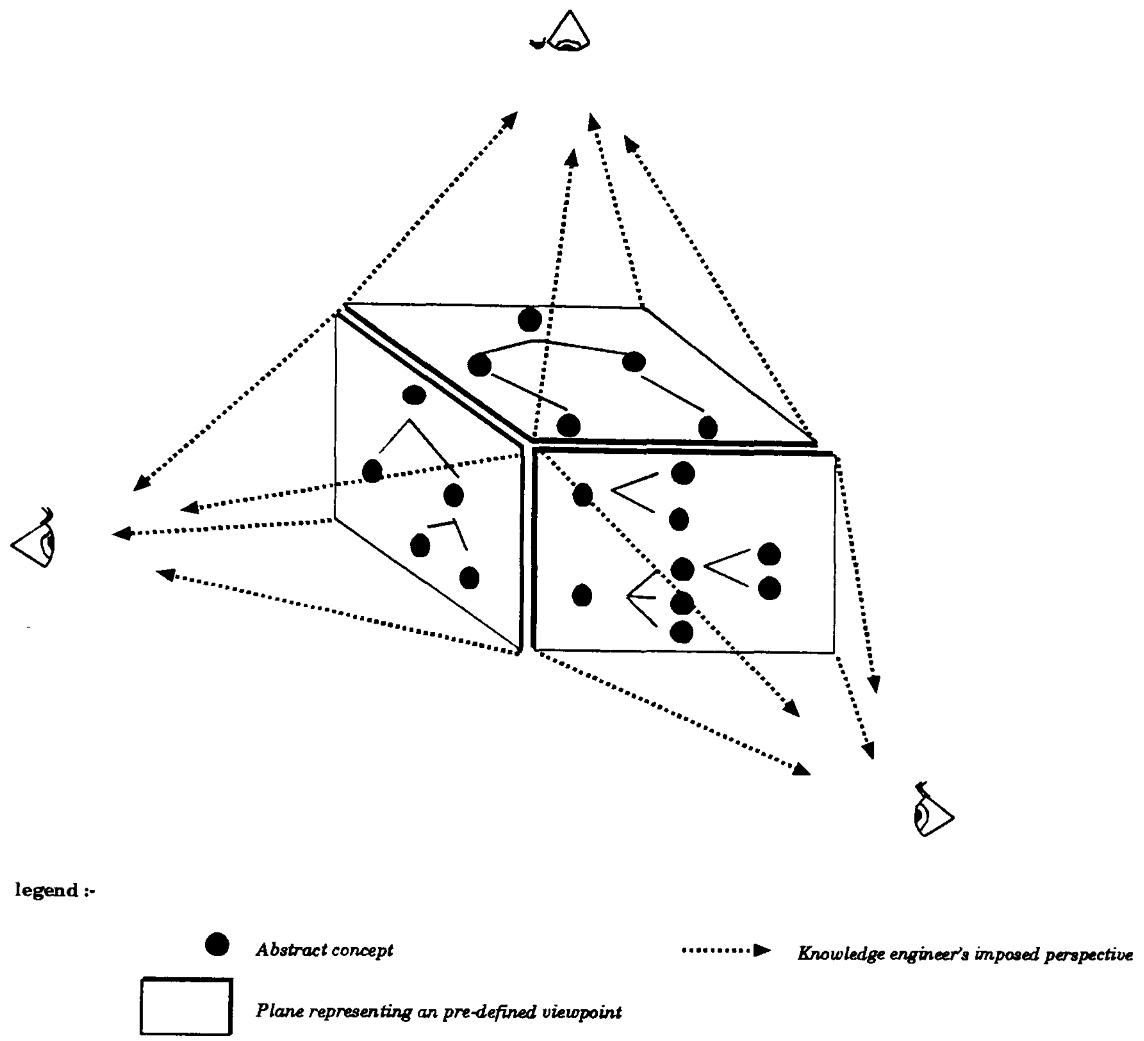


Figure 19: Illustration of 'pre-defined viewpoint' (PDV) approach showing knowledge engineers' imposed perspectives

Alternatively, the new CV approach, presented in this chapter, encourages the utilisation of more applicable knowledge by advocating the generation of viewpoints from a source of experiential knowledge, according to designers' particular knowledge requirements. Consequently, the knowledge generated from the CV approach reflects designers' perspectives not knowledge engineers'. Figure 20 illustrates this concept where the viewpoints originate from a source of experiential knowledge, represented by the sphere, and their generation is governed by designers' knowledge requirements.

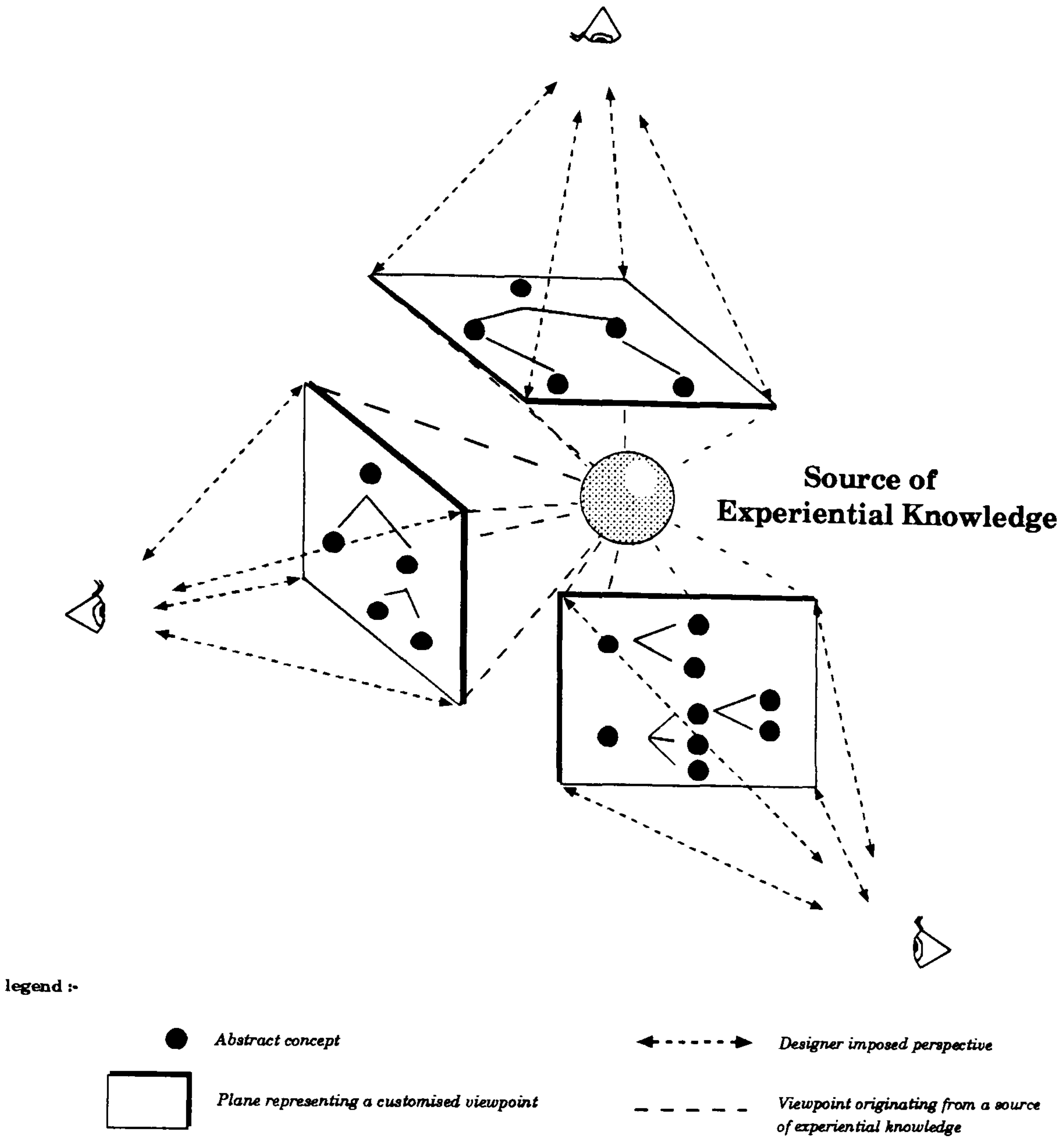


Figure 20: Illustration of 'customised viewpoint' (CV) approach showing designers' imposed perspectives

CAD systems that adopt the "design assistance" philosophy and support the effective utilisation of experiential knowledge should be more than repositories of pre-defined viewpoint and associated experiential knowledge, i.e. "It is not enough for the [Intelligent CAD] system to work with a pre-defined abstraction; it should itself be able to abstract" [16]. The application of the new CV approach is proposed to improve the effective utilisation of experiential design knowledge and thereby advance the utility of CAD systems.

Therefore, a resulting 'customised viewpoint' (CV) tool should be able:

- to reflect designers' perspectives not knowledge engineers',
- to support the structuring of knowledge,
- to support changing focus of attention within a viewpoint, i.e. support the generation of single and nested viewpoints, and
- to support the generation of viewpoints using multiple types of focus.

Utilisation of applicable knowledge

The experiential knowledge generated and stored in a viewpoint can be specific or general, according to different levels of abstraction. Consequently, a viewpoint can encompass a broad range of potentially useful knowledge, which can have varying degrees of applicability with a particular design situation. Therefore, a CV tool should also be able:

- to search for and retrieve knowledge from a viewpoint,
- to utilise specific and general knowledge for knowledge exploration, synthesis and evaluation, and
- to utilise the most applicable form and level of abstraction of experiential knowledge.

In summary, a CV tool is proposed as a "design assistant" that can play a significant role in the effective utilisation of experiential knowledge. It should maintain an up to date resource of knowledge, generate appropriate viewpoints of experiential knowledge to reflect designers' perspectives, and utilise applicable knowledge from these generated viewpoints.

7 The Components of a Numerical Customised Viewpoint Tool

It is necessary to define the scope of this thesis so that the fundamental ideas of the 'customised viewpoint' approach can be investigated. Of the two sources of experiential knowledge discussed in Chapter 4, this research had access to past designs from a particular domain type. A paper database of past designs, consisting of numerical information, provided an adequate source from which to generate general experiential knowledge. In addition, evidence of associated general experiential knowledge, in the form of empirical equations, was available from design literature. Due to the numerical nature of the available past designs and experiential knowledge, it was decided to develop, test and evaluate the utility of the 'customised viewpoint' approach within the realm of numerical design, i.e. focusing on a numerical aspect (see Chapter 4).

Numerical design is a rather complex process requiring extensive calculations and numerical alterations to obtain a design solution that satisfies particular goals. Consequently, a tool that supports numerical design has to manage applicable empirical equations, determine the influence attributes have on other attributes, record the consequent alterations of attribute values required by such influences, and manipulate attribute values to achieve an acceptable design solution. Thus, a CV tool capable of supporting numerical design should:

- manage the complexity of numerical design and experiential knowledge,
- extract and represent implicit knowledge explicitly, and
- support the opportunistic utilisation of multiple forms of experiential knowledge, i.e. empirical equations, generalisations and abstractions of empirical equations,

Using some of the approaches in Chapter 5, this chapter presents how such support can be realised. The DESIGNER system is a CAD system that supports the representation and utilisation of experiential knowledge, in the form of empirical equations, during preliminary numerical design. However, this system is dependent on pre-defined experiential knowledge. Regression, concept formation and neural networks are techniques that have supported the utilisation of implicit experiential design knowledge. Applications of regression and concept formation have, for example, used information of past designs to generate experiential knowledge in the form of empirical equations and generalisations respectively. In other words, these applications have identified and extracted implicit knowledge in an explicit form. However, applications of neural networks have not supported the explicit representation of implicit knowledge. A numerical CV tool should present generated experiential knowledge in a form that is understandable to and usable by a designer. Thus, the regression and concept formation techniques represent suitable approached to realise a numerical CV tool; the neural network technique does not. This chapter proposes that a numerical CV tool can be developed by integrating the DESIGNER system with regression and concept formation techniques to support the opportunistic utilisation of multiple forms of experiential knowledge.

Three general requirements guide the development of a CV tool to support the effective utilisation of experiential knowledge. First, the tool should support the utilisation of experiential knowledge originating from experiences (namely past designs) in the form of empirical equations and generalisations. Second, it should support the coupling of design and learning activities within a single design environment and third it should support the generation of viewpoints of experiential knowledge that reflect the needs of a designer. These requirements are fundamental to the effective utilisation of experiential knowledge in design. The DESIGNER [34], S-PLUS [141] and ECOBWEB [37] computer systems represent existing technology that was available to develop a CV tool, test the utility of the proposed approach and evaluate its benefit within the realm of numerical design.

This chapter focuses on these systems to explain how they can assist in the development of a CV tool that supports numerical design. Section 7.1 presents an overview of these three systems by discussing some of their main features. This section focuses on the support for design and learning activities, to ascertain the activities each system supports and explain their contribution to design. Section 7.2 presents the contribution of each system to the concept of a CV tool by discussing the complementary roles each can play in design. Section 7.3 presents a number of additions to the key features and strengths of the three systems, which are required to demonstrate the approach proposed in Chapter 6.

7.1 Overview of Systems

This section presents an overview of the DESIGNER [34], S-PLUS [141] and ECOBWEB [37] systems and explains their contribution to design. The DESIGNER system's origins began in knowledge based computer aided design and its intended function is to aid designers in numerical design during the preliminary design stages. The S-PLUS system is a graphical data analysis system encompassing many data analysis techniques in one compact system for use in many application domains not necessarily design (e.g. medical surveys). The ECOBWEB system, developed specifically for use in design domains, is a concept formation system that generates a hierarchy of concepts from a number of observations (i.e. instances described by attributes and values) and uses this hierarchy to help predict the descriptions of incomplete observations. This overview section provides the basis for Section 7.2, where the contribution of each system to the development of a CV tool is assessed.

7.1.1 The Designer System

The DESIGNER system is a CAD system that supports numerical preliminary design, and describes a design domain (e.g. car, ship or pump design) with numerical attributes and relationships between these attributes, and a design concept (i.e. the concept of the artefact to be designed) using these attributes and goals to be satisfied. DESIGNER assists users to define their design solution by (a) managing the application of suitable empirical equations, (b) determining the influence attributes have on other attributes and (c) assessing the degree of design goals satisfaction and attribute value uncertainty. Thus, according to the design activities presented in Chapter 3.3., DESIGNER supports synthesis and evaluation.

Details of DESIGNER's Functionality

The functionality of DESIGNER is dependent on the definition of a *domain model*, which is built using DESIGNER's *model building* facility using a *domain description* of a particular design domain. (see Figure 21)

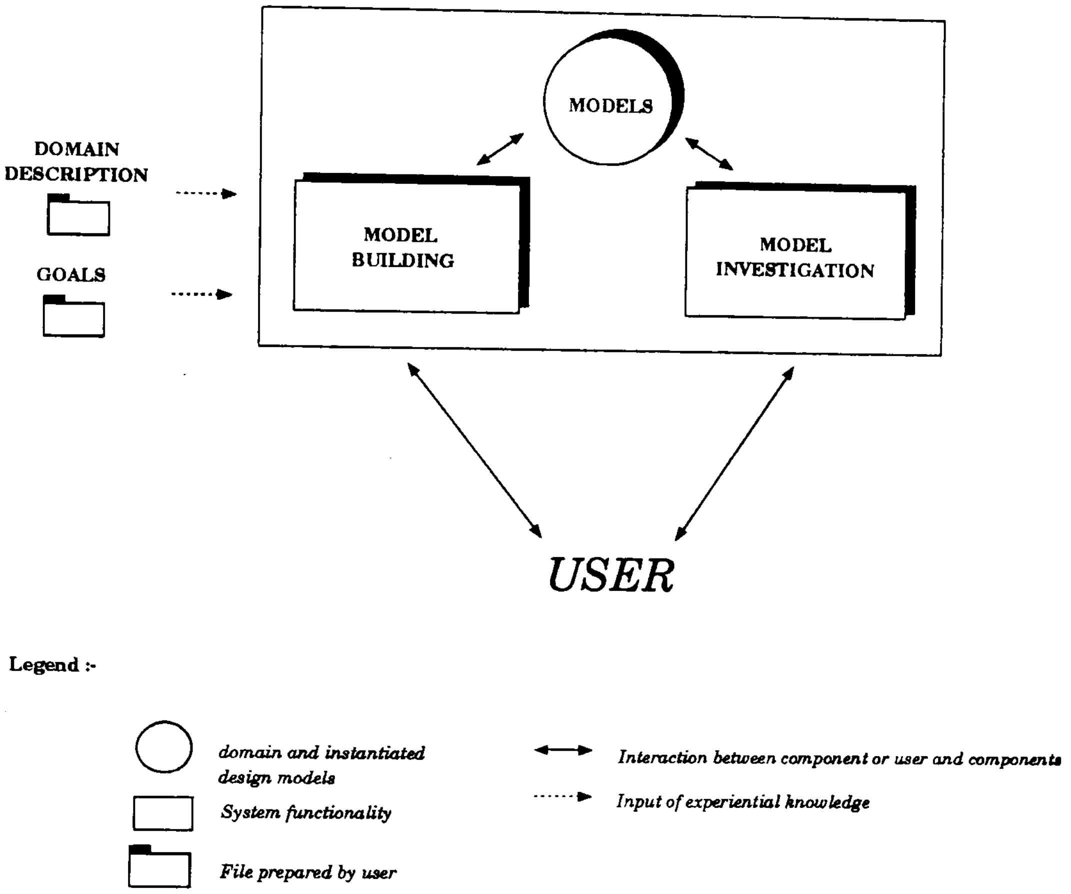


Figure 21: DESIGNER's architecture, adapted from [34]

This domain description contains explicit experiential knowledge of a domain, which a designer requires during numerical design of a new design, i.e. the name, meaning and units of attributes, empirical equations that quantify the relationship between these attributes and a measure of the unreliability of these equations. For example, in the domain of ship design, 'dwt' is the name of a ship's attribute, 'cargo dead-weight' is its meaning, 'tonnes' are its units, '$disp \times ddratio$' is an equation that quantifies the relationship between the attributes 'dwt', 'disp' and 'ddratio', and the equation's measure of unreliability is '0.1%'. (See Appendix A for an example listing of a domain description for ship bulker design)

In addition, the user can prepare a file containing knowledge of goals or interactively define goals that the new design is to achieve. The user defines goals using goal names, meanings, units, descriptions and importance values. For example 'dwtgl' is the name of a goal, 'dead-weight goal' is its meaning, 'tonnes' are its units, 'dwt $\leq$ 61000' is its description, and '80%' is its importance value. (See Appendix B for an example listing of a goal definition file) Designer can represent five types of goals. Table 7 lists these goal types and provides some examples [142].

Goal Type	Goal Structure	Example
A	C O V	$weight = 36000$
B	V1 O C O V2	$36000 < weight < 46000$
C	C1 O C2	$pressure > pressure_{required}$
D	C1 O C O C2	$temperature_{min} < temperature < temperature_{max}$
E	C1 O C2 A V	$pressure > width \times 0.06$

where:

C = a design attribute
O = an operator (e.g. $=, <, \leq, >, \geq$)
V = a value
A = an arithmetic operator (e.g. $+, -, \div, \times$)

Table 7: Goals accommodated in the DESIGNER system [142]

By inputting attribute values, a user initiates the development of a design solution. Using these values, DESIGNER can determine the values of dependent attributes, update attribute values, and provide information on the status of design goals and how to change the values of specified attributes of a *design model.*

DESIGNER's Support: Design

DESIGNER supports model building and investigation. The system helps a user build (i.e. synthesise) a DOMAIN MODEL from EXPERIENTIAL KNOWLEDGE that characterises a design domain. A user can then initiate design investigation by inputting initial values for certain attributes of a required design solution. The result is a partially complete design description of the new design represented by the **design model**. Then, using the DOMAIN MODEL, DESIGNER assists a user to evolve a **design model** towards a desired state, by determining and propogating the effect attributes have on dependent ones, and consequently expands the **design model** to represent a complete description of a NEW DESIGN. Figure 22 characterises this support.

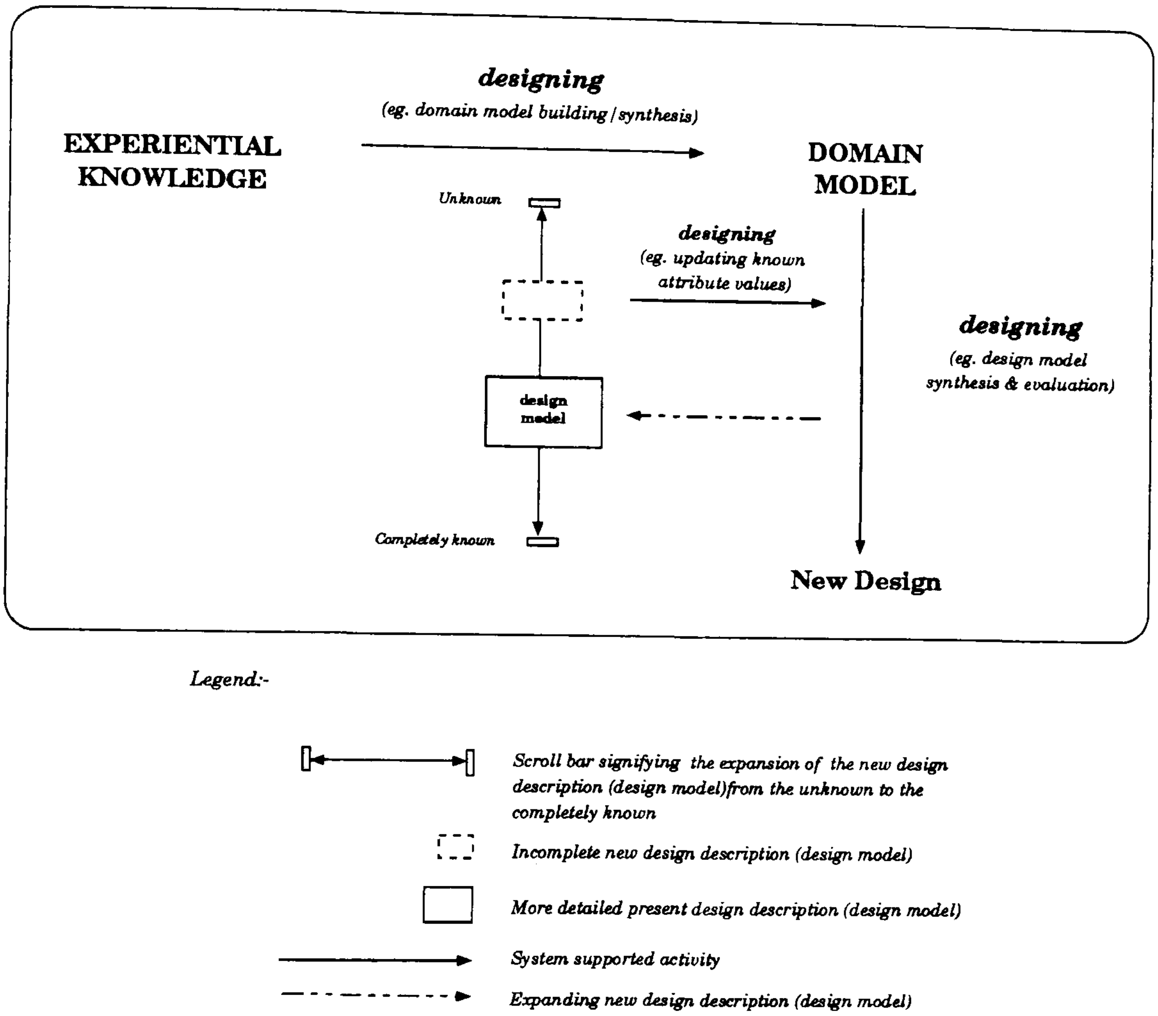


Figure 22: DESIGNER's activity diagram

The purpose of the 'scroll bar' in Figure 22 is to indicate that attribute-values used in the **design model** can be either instantiated (i.e. values set to some numerical value) or uninstantiated (i.e. values set to a non numerical value called 'unknown'). Thereby, the scroll bar represents the possibility of either increasing or decreasing the description of the new design respectively.

7.1.2 The S-Plus System

The S-PLUS system [141] is a computing environment, which provides a graphical data analysis system and an object-oriented language called S [143]. Its purpose is to provide a means of exploring information from a wide range of domains (not necessarily from a design domain). However, this section discusses S-PLUS in the context of exploring past design information, i.e. as a tool for supporting the identification and extraction of general experiential knowledge.

Details of S-PLUS's Functionality

S-PLUS supports the visualisation and analysis of past design information to generate, for example, empirical equations. Figure 23 presents an interpretation of S-PLUS's architecture (one does not exist in the literature).

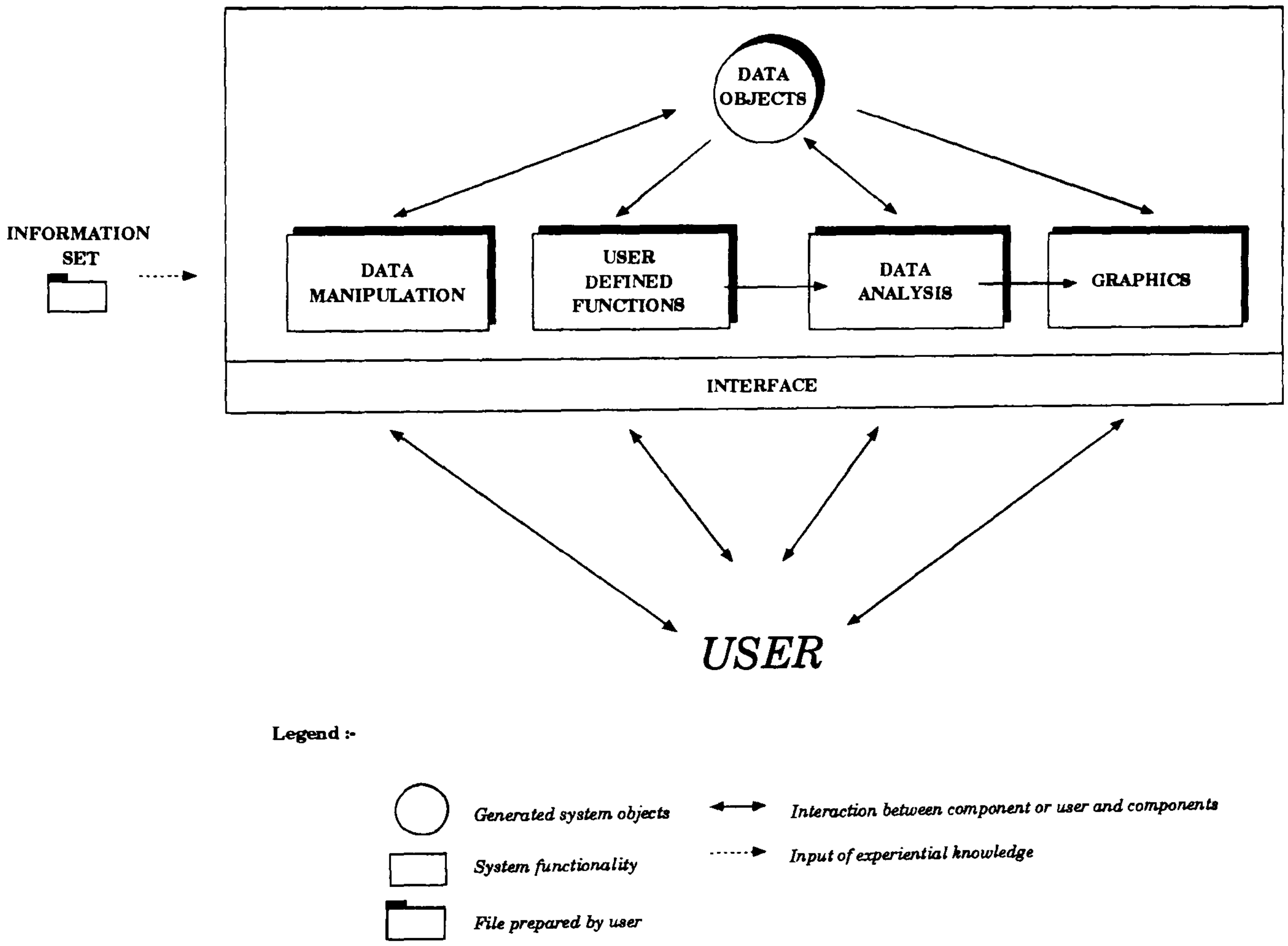


Figure 23: S-PLUS's interpreted architecture

As illustrated in Figure 23, S-PLUS translates the *information set* (see Table 8 for the syntax of the *information set* file) to be analysed (i.e. past designs) into one of its *data objects* by *data manipulation.* This functionality allows the user to view and/or manipulate the information content of these objects using S-PLUS's data editor. The data editor presents the user with a window representing a table containing the content of specified object, which users can interact with using keyboard strokes to move across columns (i.e. design attributes) and down rows (i.e. individual past designs) of the table. In addition, the *graphics* functionality visually displays information associated with data objects and the results of *data analysis.* For example, it can plot graphs in 2 or 3 dimensions where the axes of the graphs represent specified design attributes and the plots on the graph represent individual past designs. The user can access the *data analysis* techniques from the *interface* or from the *user defined functions* and can analyse the whole or portions of the information set. *User defined functions* are groups of operations, which users can define, save and modify using an incorporated VI editor (the standard UNIX screen editor). S-PLUS can directly feed back results of applying data analysis techniques either to the user or the graphics package.

IDENTIFIER	<*attribute-name*>	. . .	<*attribute-name*>
<*past-design-name*>	<*value*>	. . .	<*value*>
<*past-design-name*>	<*value*>	. . .	<*value*>
.			
.			
.			
<*past-design-name*>	<*value*>	. . .	<*value*>

Table 8: Syntax of *information set* file

S-PLUS provides various techniques for data analysis, e.g. principal component analysis, time series analysis, survival analysis and regression analysis. Regression is one of the most relevant for exploring experiential knowledge of past designs in engineering. Therefore, this thesis utilises regression to help illustrate the CV approach.

S-PLUS provides a broad range of linear and non-linear regression techniques. Regression is a technique that quantifies the relationship between specified independent (explanatory) variables and a specified dependent (response) variable(s) of an information set. For example, assume a user wishes to quantify the relationship between two variables x and y of an information set, where x is the explanatory variable and y is the response variable. Figure 24(a) represents the distribution (scatter) of instances in the information set by plotting the variables x against y.

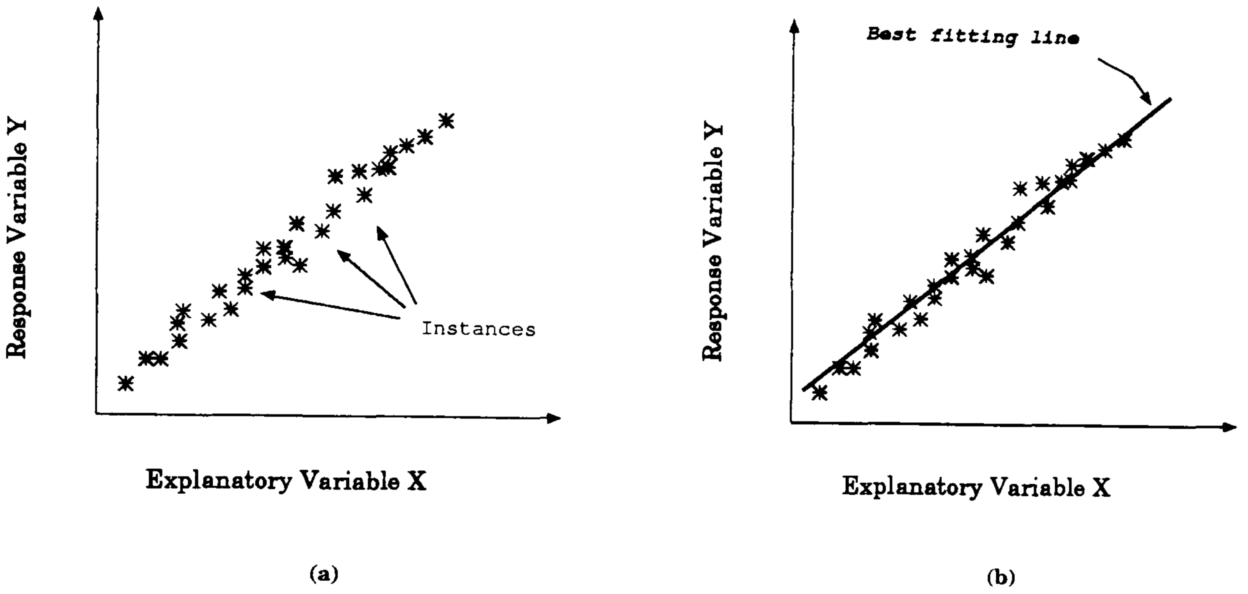


Figure 24: Scatter of instances by plotting explanatory variable x against response variable y

A relationship between these two variables exists implicitly in the scatter of instances, which regression characterises as the best fitting line that runs through the bulk of the instances (see Figure 24(b)). Consequently, using the definition of this line and a value for an explanatory variable, the value of the response variable can be estimated. In Figure 25, the majority of instances have been removed from the plot to help explain how this line is determined. X_1 and X_2 represent the explanatory values of two instances. Y_1 and Y_2 represent the respective actual response values and $Y_1^{'}$ and $Y_2^{'}$ represent the estimated response values. The quality of the fitted line is greatest when the difference between the actual value of the response variable and the estimated value of the response variable is minimised for all instances. Since some estimates can be greater than the actual value of response variable this difference can have a negative sign (as in the respective values of X_1, i.e. $Y_1 - Y_1'$ equals a negative difference); thus differences are squared to remove sign conventions. Therefore, the best fitting line is quantified statistically using the *principle of least squares*, which minimises the sum of squares of the deviations between observed and estimated values of response variables.

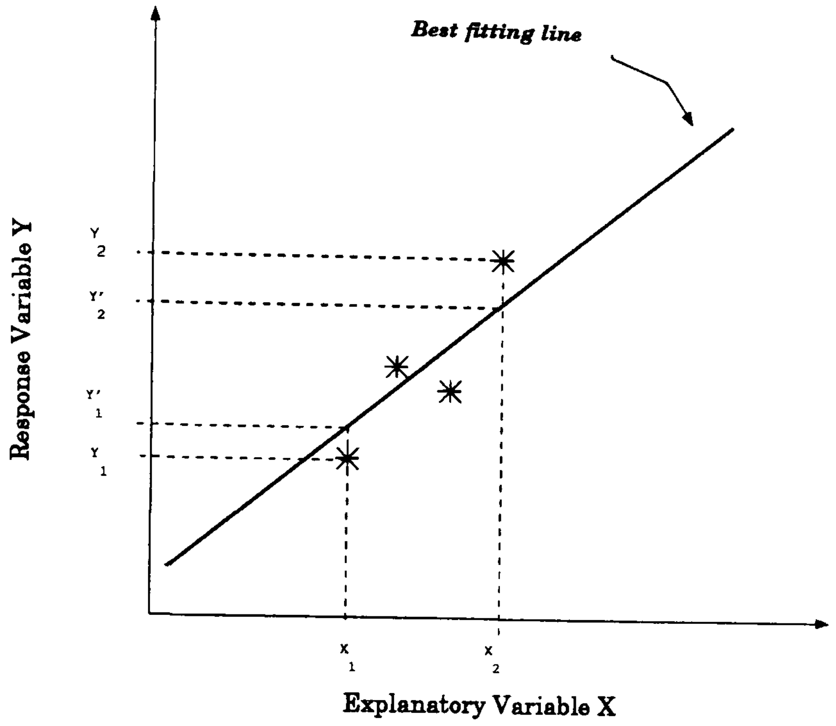


Figure 25: Illustration of least squares principle

The above simple example explains the *linear* relationship between an explanatory and the response variable. In general, this linear relationship can be expressed as:

$$y = B_0 + B_1x_1 + B_2x_2 + B_3x_3 + \cdots + B_nx_n$$

where,

y = response variable
$x_1, \ldots, x_n$ = explanatory variables
$B_0, \ldots, B_n$ = parameters to be determined by regression

S-PLUS's Support: Design and Learning

Although S-PLUS can support the exploration design activity by the presentation and analysis of past design information, it does not support either of the synthesis or evaluation design activities. Consequently, S-PLUS supports learning by the generation of experiential knowledge in the form of empirical equations.

Figure 26 illustrates how S-PLUS can render **EXPERIENTIAL KNOWLEDGE**, in the form of numerical equations, from a **Set of Past Designs**. Focusing on specific attributes of past designs, believed to be interrelated, users can use S-PLUS's regression analysis to generate empirical equations that describe the relationship between these attributes. Consequently, by applying regression to combinations of attributes, users can generate a multitude of empirical equations.

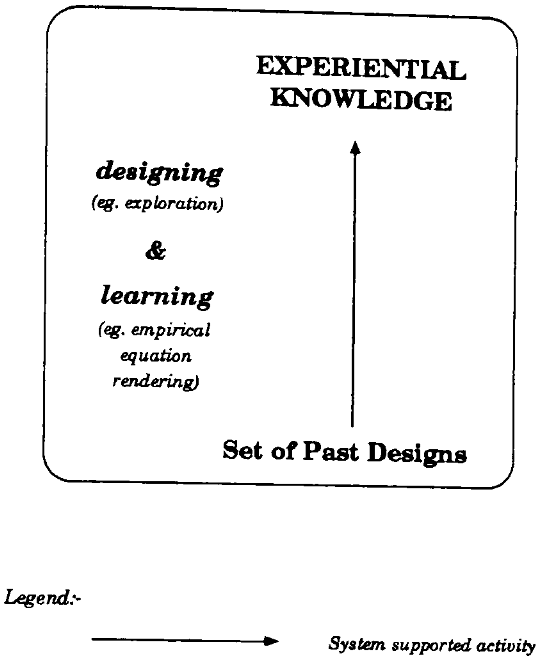


Figure 26: S-PLUS's activity diagram

7.1.3 The Ecobweb System

ECOBWEB is an extension of the concept formation system called COBWEB [144]. The extensions specifically address a number of deficiencies [37] in COBWEB's utility in learning experiential knowledge from past designs. This section discusses ECOBWEB's ability to support the exploration and synthesis design activities during numerical design. As will be discussed, support for evaluation is not possible using ECOBWEB.

Details of ECOBWEB's Functionality

ECOBWEB can assist designers to develop a design solution by automatically generating generalisations of past design information and identifying the most suitable generalisation or past design that is most applicable in a new design. Figure 27 presents a schematic of ECOBWEB's architecture.

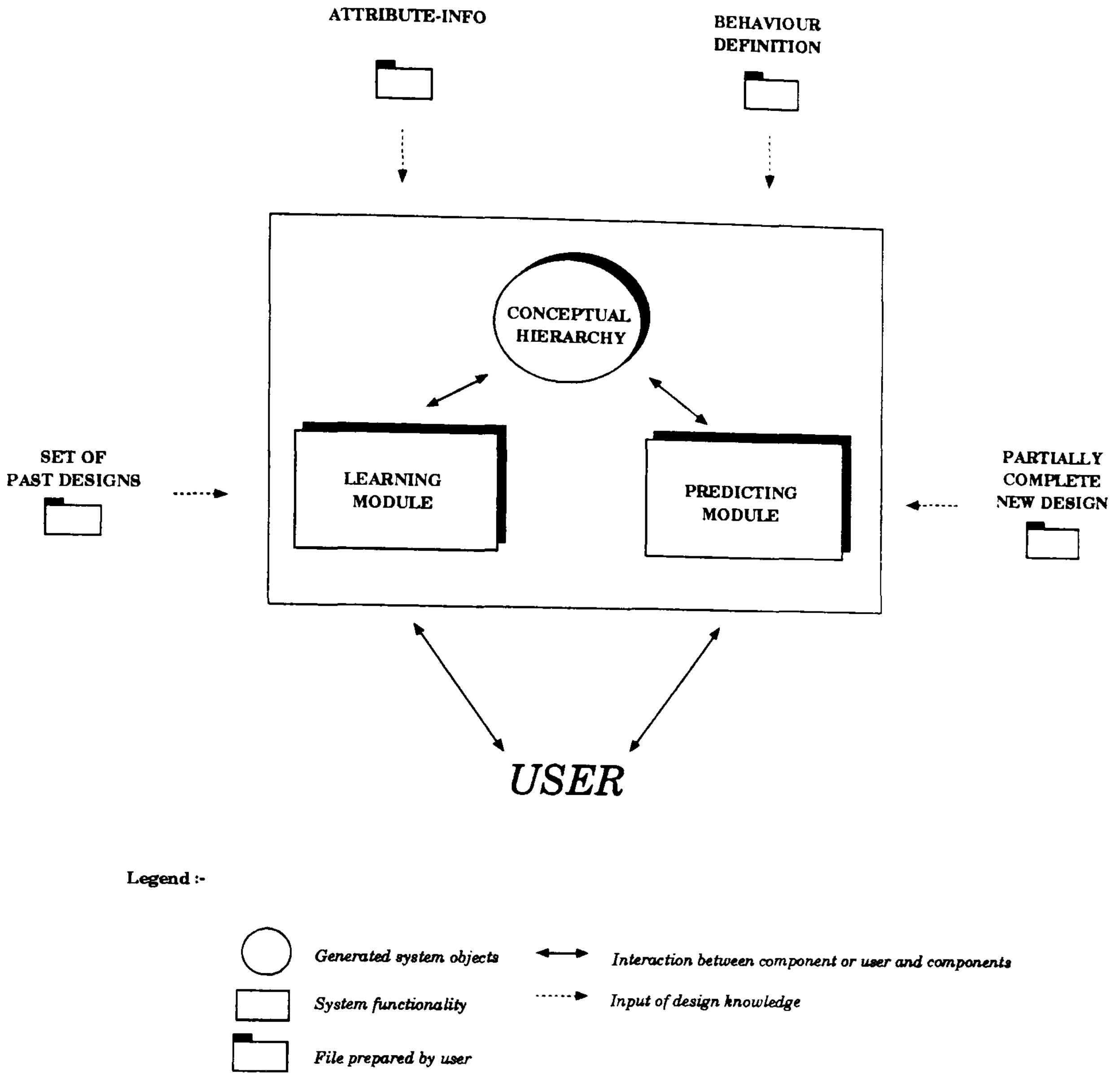


Figure 27: ECOBWEB's architecture, adapted from [37]

ECOBWEB operates in two modes: learning and predicting. The learning mode generates and organises generalisations of a *set of past designs* (see Table 9 for the syntax of this file and Appendix C for a partial example listing of such a file) into a *conceptual hierarchy* (i.e. viewpoint) using the *learning module.* The prediction module then uses this viewpoint to find a specific past designs or group of designs that are similar to a *partially complete new design.* (See Table 10 for the syntax of the *partially complete new design* file)

```
example
<past-design-name>
positive
<attribute-name>        <value>
.
.
.
end
.
.
.
example
<past-design-name>
positive
<attribute-name>        <value>
.
.
.
end
end-examples
```

Table 9: Syntax of *set of past designs* file

```
example
<new-design-name>
positive
<attribute-name>        <value>
<attribute-name>        <value>
.
.
.
<attribute-name>        <value>
end
end-examples
```

Table 10: Syntax of *partially complete new design* file

Prior to learning and predicting, the user should prepare an *attribute-info* and *behaviour definition* file or define their contents interactively. The *attribute-info* file details the attributes used in the *set of past designs* as either specifications or design-descriptors, defines the type of attributes used (e.g. continuous or nominal), and gives the interval range of continuous attributes (i.e. the difference between the maximum and minimum value for each attribute). The *behaviour-definition* file sets up a number of global variables that define how ECOBWEB will carry out learning and predicting, e.g. the method chosen to predict (ECOBWEB provides various prediction methods however they follow the same basic procedure mentioned here). The precise details of these files are not of direct concern to this thesis. However, for interest, the reader is referred to Appendix D, which details the syntax on the *attribute-info* file, and Appendix E, which gives an example listing of the *behaviour-definition* file that defines the behaviour of ECOBWEB used in this thesis.

In the learning mode, ECOBWEB classifies each example (past design) and incorporates it *permanently* into a hierarchy by incrementally changing the hierarchy's structure. At the initial stage of learning, if no hierarchy exists then ECOBWEB initiates the first example as the root concept of a hierarchy. ECOBWEB then sorts each subsequent example in the *set of past designs* down through each level of the hierarchy. To determine how best to incorporate (i.e. classified) an example at each level in a hierarchy, ECOBWEB employs five operators. (see Table 11 for the list of operators) The system evaluates each resulting classification using a *utility function* to determine a value of *category utility*, i.e. a singular measure quantifying the similarity between the components (members) of a classification. ECOBWEB chooses the classification that produces the highest category utility value, incorporates the example permanently into the hierarchy, and generates the appropriate conceptual description that encompasses the newly incorporated example. This process is repeated until there are no more concepts to incorporate the example.

- Expand the root concept if it has no subconcepts.
- Add example as a new concept.
- Add example to a concept.
- Merge the best two concepts and add the example to the newly merged concept.
- Split the best concept into its constituent subconcepts and consider best constituent concept.

Table 11: Five operators that ECOBWEB can use to incorporate an example into a hierarchy

ECOBWEB supports the classification of examples described by nominal or continuous attribute types. However, this thesis uses ECOBWEB in the context of numerical design, therefore it only considers continuous attribute types. The extended *utility function*, developed to handle continuous attribute values, is as follows:

$$\textit{Utility Function} = \frac{\sum_{k=1}^{n} P(C_k) \sum_i P(A_i = \overline{V_i} \mid C_k)^2 - \sum_i P(A_i = \overline{V_i})^2}{\mathrm{n}}$$

where :

C_k is a concept,
A_i is an attribute,
$\overline{V_i}$ is the average value of attribute A_i,
n = the number of possible concepts,
$P(C_k)$ is the probability of concept $C_k = \dfrac{\text{no. of examples in concept } C_k}{\text{no. of examples in the parent concept of } C_k}$,

$P(A_i = \overline{V_i} \mid C_k)$ is the conditional probability of attribute A_i having a mean value $\overline{V_i}$ given its membership to the concept C_k, (i.e. the average value of A_i in concept).

$P(A_i = \overline{V_i})$ is the probability of attribute A_i having a mean value $\overline{V_i}$, (i.e. the average value of A_i in the parent concept.

$P(A_i = \overline{V_i} \mid C_k)^2$ and $P(A_i = \overline{V_i})^2$ are calculated using the area under a probability distribution curve. ECOBWEB provides the facility to define the type of probability distribution (e.g. normal or cauchy). The default type is normal.

In the predicting mode, ECOBWEB classifies a partially complete new design, as in the learning mode, but without incorporating it into the hierarchy. ECOBWEB merely locates the most specific concept that describes the new design description. ECOBWEB provides two basic approaches that characterise the nature of the similar concept to be chosen during the predicting mode. It can choose a concept that is a generalisation of past designs (i.e. prototype-based design) or a specific past design (case-based design) to help develop the new design. In addition, the chosen concept description relating to a generalisation of past designs can be generated by one of two ways: by finding the parent of the best selected concept and returning the parent's description; or finding the parent of the best selected concept and returning the parent's 'weighted' description, i.e. a parents description that has been modified to take into consideration the best selected child concept.

ECOBWEB's Support: Design and Learning

Like the S-PLUS system, ECOBWEB supports both design and learning activities, as highlighted in Figure 28. ECOBWEB manipulates past design information (**Set of Past Designs**) to generate **EXPERIENTIAL KNOWLEDGE** in the form of a viewpoint detailed by generalisations. However, this process of manipulation is not guided by knowledge requirements that arise during design or that suit the particular needs of a designer. ECOBWEB uses all the attributes known for the set of past designs to render an all encompassing viewpoint. Therefore, generated concepts are generalisations of the complete past design information and represent only one viewpoint. The purpose of the 'scroll bar', in Figure 28, is to indicate the relative increase in the description of the new design (i.e. the instantiation of attribute-values used in the **design model**). Using a generated viewpoint and a partial description of a the new design, ECOBWEB can find the most similar concept (i.e. a generalisation or specific past design) with which the associated conceptual description can be used to expand the description of the new design.

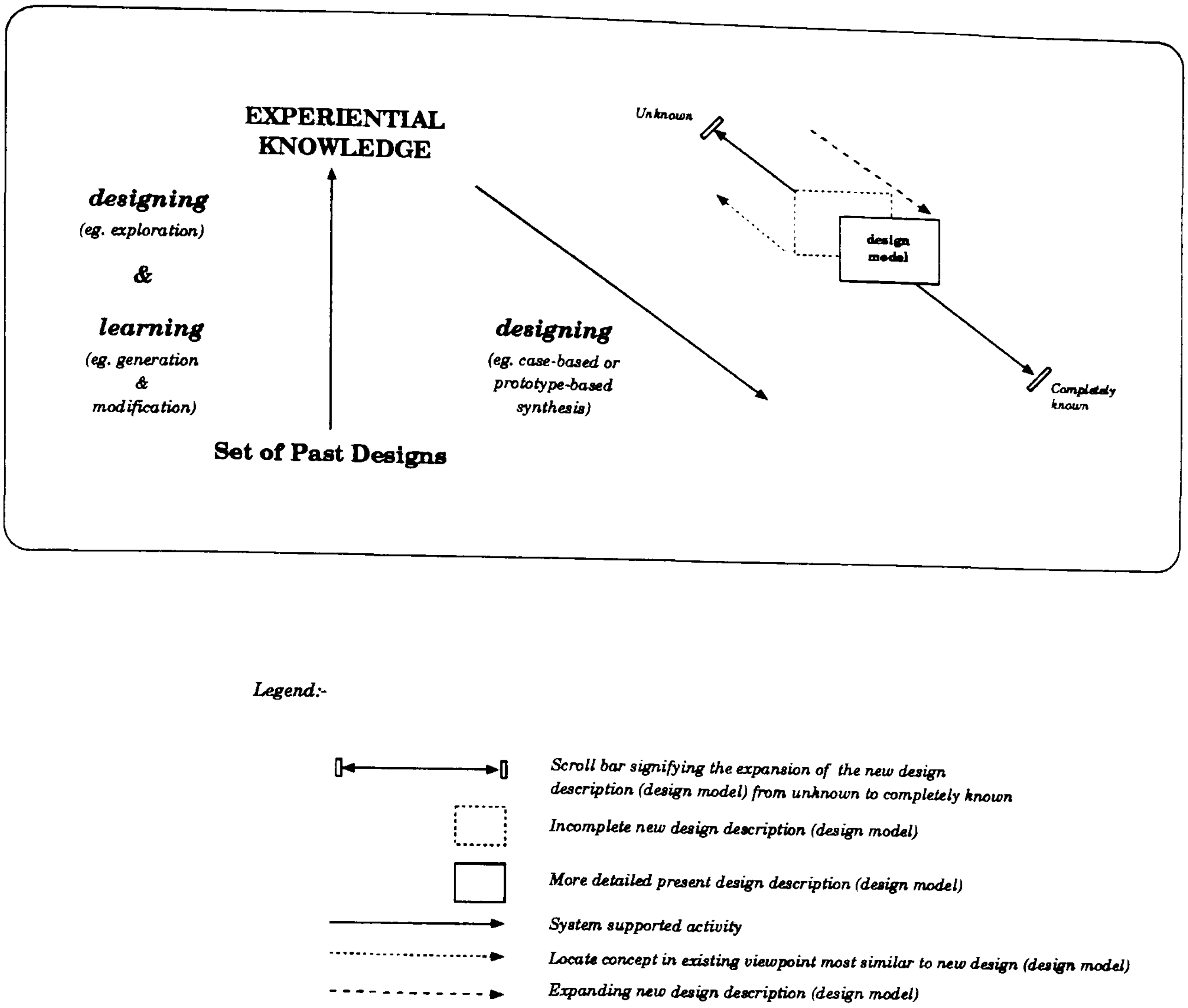


Figure 28: ECOBWEB's activity diagram

7.2 Contribution of Systems to a Numerical Customised Viewpoint Tool

DESIGNER, S-PLUS and ECOBWEB are considered to play complimentary roles in the application of experiential knowledge for numerical design, and are therefore proposed to provide valuable support towards developing a CV tool for numerical design. This section discusses the contribution of each system to the development of such a tool, by comparing and contrasting their functionality to identify their main limitations.

The functionality of the DESIGNER system is dependent on the representation of relationships between design attributes, in the form of empirical equations. Without these equations it is difficult to quantify the new design, i.e. carry out synthesis and evaluation design activities. Also if complex equations are defined, they can be difficult for the designer to use. Complex equations have many input variables. Therefore, if the designer is unable to provide values for these variables then estimation of associated attribute values is not possible. Although S-PLUS is incapable of supporting synthesis and evaluation, it can support exploration. Using S-PLUS, a designer can explore a set of past designs and render suitable empirical equations, which can be subsequently utilised by DESIGNER. Consequently, S-PLUS, when used with DESIGNER, can bypass the need to pre-define domain models, and can help to overcome the compulsory use of complex equations.

Although ECOBWEB cannot evaluate a new design, it can help during synthesis without the use of empirical equations. By identifying groups of similar past designs and organising them into an all encompassing single viewpoint of experiential knowledge, ECOBWEB can find a group or past design similar to a new design and and use the associated description to further quantify this new design. Having said this, ECOBWEB is unable to provide assistance in estimating the *degree* of interaction between attributes. However, DESIGNER provides this facility. In addition, ECOBWEB is unable to support the generation, representation and utilisation of *customised* single and nested viewpoints that contain experiential knowledge of empirical equations; the single viewpoint ECOBWEB generates is governed by a single pre-defined perspective prepared by the user in the *attribute-info.*

Consequently, this thesis proposes to integrate these systems and therefore benefit from the opportunistic utilisation of these different experiential knowledge forms. One of the general requirements of any support for the utilisation of experiential knowledge (presented in Chapter 4) is the need to support the representation of multiple forms of knowledge. The DESIGNER, S-PLUS and ECOBWEB systems generate and/or utilise different forms of experiential knowledge and therefore represent possible tools that if integrated can help realise such required support. ECOBWEB is proposed to provide additional support to DESIGNER by automatically generating and utilising an alternative form of experiential knowledge from past designs, i.e. generalisations. S-PLUS has the potential of (a) supporting domain model rendering, thereby freeing DESIGNER's dependency on a pre-defined domain model, and (b) enriching the viewpoints generated by ECOBWEB, i.e. providing empirical equations that can be used to further describe the viewpoints of experiential knowledge generated by ECOBWEB. Table 12 summarises a number of key features that each system supplies to preliminary design.

DESIGNER	S-PLUS	ECOBWEB
Supports no learning	Supports learning	Supports learning
Supports synthesis and evaluation	Supports exploration	Supports exploration and synthesis
Represents design model and domain model	Represents past designs and empirical equations	Represents new and past designs, and single viewpoints of experiential knowledge consisting of groups (concepts) of past designs
Represents goals, attributes and empirical equations	Represents attributes, empirical equation information	Represents attributes, generalisations
Propagates numerical changes dependent on pre-defined domain model using empirical equations	Generates empirical equation information	Generates generalisations of past designs and finds the design group/case most similar to the new design

Table 12: Summary of features supplied by DESIGNER, S-PLUS and ECOBWEB systems

In summary the limitations of these systems are:

- DESIGNER's inability to utilise experiential knowledge other than empirical equations and its dependency on a pre-defined model of the domain described using pre-compiled empirical equations.
- ECOBWEB's inability to support the generation, representation and utilisation of customised single and nested viewpoints containing experiential knowledge of empirical equations.
- S-PLUS's cumbersome means of generating empirical equations according to designers knowledge requirements.

7.3 Further Required Functionality

Chapter 6 proposed a number of specific requirements as necessary for the development of a CV tool capable of supporting numerical design. DESIGNER, S-PLUS and ECOBWEB systems represent existing technology that address these requirements and, in this chapter, have been discussed to identify their contributions and limitations to the realisation of this approach. Therefore, this thesis proposes that a numerical CAD system that employs the CV approach to promote the utilisation of experiential knowledge can be built using the DESIGNER, S-PLUS and ECOBWEB systems. However, the following support is still required to fulfil the implementation of such a tool.

- Integration of multiple forms of experiential knowledge
- Definition of perspectives
- Generation, representation, utilisation and presentation of customised single and nested viewpoints
- Rendering and abstraction of domain models.

Consequently, a tool that supports these issues can help designers to opportunistically utilise multiple forms of experiential knowledge, couple design and learning activities and manipulate past designs to extract and use suitable experiential knowledge according to designers' needs.

8 Required Implementation

The aim of this chapter is to present how the required functionality of a numerical CV tool can be achieved. Section 8.1 discusses how the integration of multiple forms of experiential knowledge can be achieved to support the opportunistic utilisation of knowledge. Section 8.2 explains how designer's perspectives, from which customised viewpoints of experiential knowledge are generated, can be defined. Section 8.3 presents how customised single and nested viewpoints of experiential knowledge can be represented (by a data structure called **rationalisation**), generated and utilised. Section 8.4 states the means by which these rationalisations can be displayed. Section 8.5 explains how a numerical domain model, as used by the DESIGNER system, can be rendered and abstracted. Section 8.6 closes this chapter with a summary.

8.1 Transfer of Information

To fully utilise the strengths of the DESIGNER, S-PLUS and ECOBWEB subsystems within one system, the proposed numerical CV tool should integrate the experiential knowledge utilised by each subsystem. This involves transferring information from one system to another.

Figure 29 illustrates this integration as a schematic diagram. This diagram represents the proposed numerical CV tool as consisting of four elements: DESIGNER, S-PLUS, ECOBWEB and a *core* element (shown by the shaded triangle). It is the *core* element that is to facilitate the transfer of useful information from each system. The *set of past designs, behaviour-definition, goals* and *domain description* input files are the same as those described in Chapter 7 in the DESIGNER and ECOBWEB overview sections 7.1.1 and 7.1.3 respectively. However, the *set of attribute-types* file is a new, required file. ECOBWEB requires the definition of all the attribute intervals. Thus, for the numerical CV tool, it was decided that this should be calculated automatically. Therefore, the *set of attribute-types* file need only define the attributes used in the *set of past designs* file as being continuous. Table 13 details the syntax of this file and Appendix F gives an example listing of such a file.

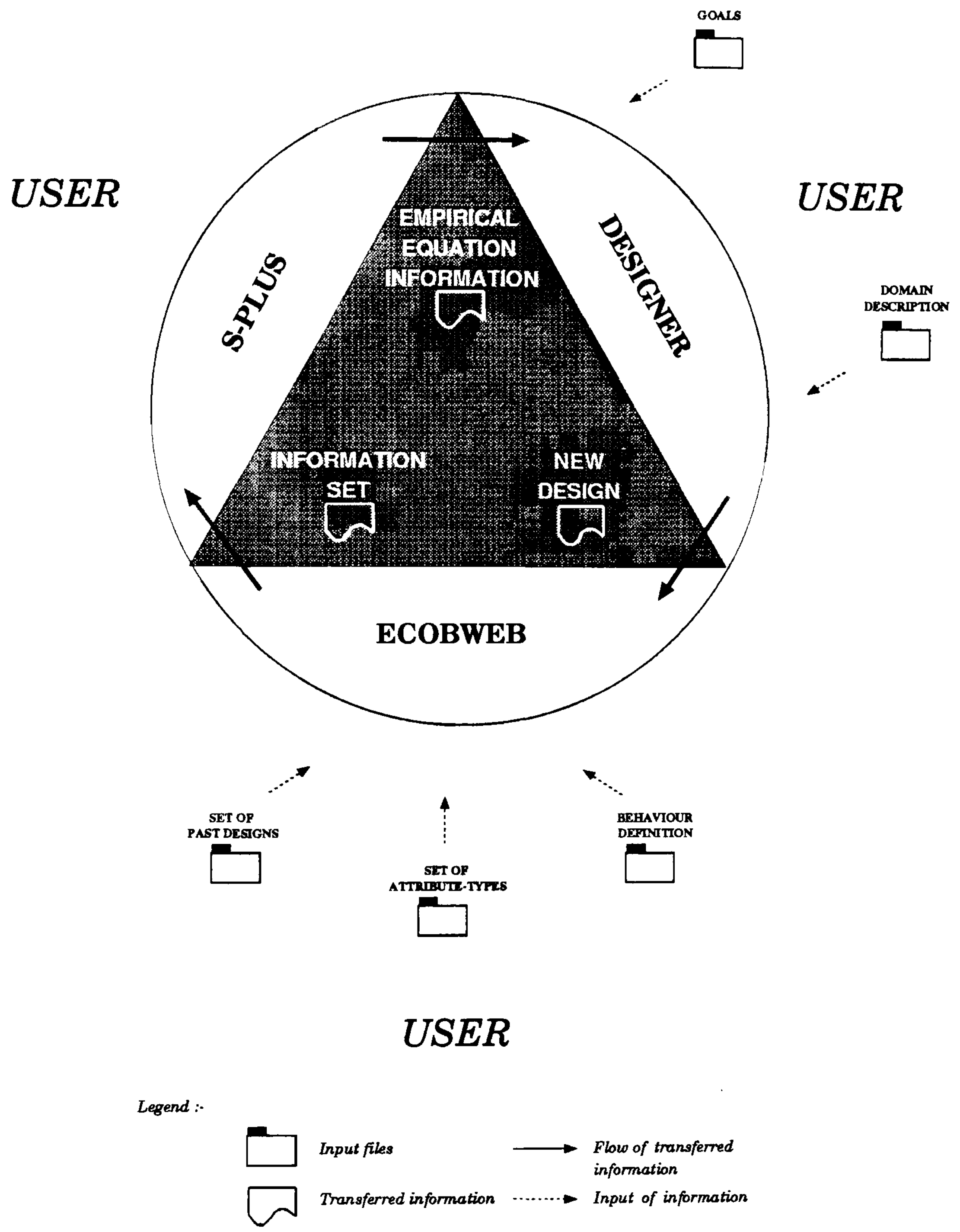


Figure 29: Flow of transferred information through DESIGNER, S-PLUS and ECOBWEB systems

attribute-types
<*attribute-name*> continuous
<*attribute-name*> continuous
.
.
.
<*attribute-name*> continuous
end

Table 13: Syntax of *set of attribute-type* file

Information transfer is required to integrate subsystem functionalities and to avoid the need for users to (a) prepare unnecessarily files that duplicate information and (b) extract and reformat appropriate information from one system to another. Therefore, three operations are required to transfer information from one system to another: ECOBWEB to S-PLUS, S-PLUS to DESIGNER and DESIGNER to ECOBWEB.

- **ECOBWEB to S-PLUS**

  Both S-PLUS's information set file and ECOBWEB's set of past designs file represent the same information associated with past designs. To avoid the unnecessary preparation of two separate files, each to be utilised by their respective system, the CV tool need only require the preparation of one file (i.e. the set of past designs to be read by ECOBWEB) and provide a facility with which to translate the contents of this file to a file usable by S-PLUS. This automatically generated file is illustrated in Figure 29 as the **information set** file. Table 14 details the syntax of this file.

- **S-PLUS to DESIGNER**

  It is intended that the DESIGNER subsystem will use the equations rendered by S-PLUS. Consequently, the results of S-PLUS (i.e. rendered empirical equation information) needs to be transferred into empirical equation knowledge and incorporated into DESIGNER's domain model. Figure 29 illustrates S-PLUS's results as the **empirical equation information** file. Table 15 details the syntax of this file, and Tables 16 and 17 give examples of rendered empirical equation information and empirical equation knowledge.

IDENTIFIER	<*attribute-name*>	. . .	<*attribute-name*>
<*past-design-name*>	<*value*>	. . .	<*value*>
<*past-design-name*>	<*value*>	. . .	<*value*>
.			
.			
.			
<*past-design-name*>	<*value*>	. . .	<*value*>

Table 14: Syntax of automatically generated *information set* file

(<*dependent-attribute-name*>	intercept	<*input-attribute-name*>	. . .	)
<*attribute-coefficient*>	.	.	.	

Table 15: Syntax of automatically generated *empirical equation information* file

(Z	intercept	X	Y)
1	1	2	

Table 16: Example of the content of empirical equation information file

Equation Name :- *to be set by user*
Equation Owner :- Z
Formula :- $1 + (1 \times X) + (2 \times Y)$
Input variables :- X, Y
Unreliability :- *to be calculated*

Table 17: Example of empirical equation knowledge

- **DESIGNER to ECOBWEB**

  DESIGNER's design model, which represents the description of the new design, is to be used by ECOBWEB to find value predictions of design attribute-values. Therefore, DESIGNER's design model needs to be translated into a syntax usable by ECOBWEB. Figure 29 illustrates the file that facilitates this as the (**new design**) file and Table 18 details the file's syntax.

```
example
<new-design-name>
positive
<attribute-name>      <value>
<attribute-name>      <value>
.
.
.
<attribute-name>      <value>
end
end-examples
```

Table 18: Syntax of automatically generated *new design* file

8.2 Defining Perspectives

Perspectives map designers' knowledge needs onto viewpoints of experiential knowledge. Two subsystems of the proposed numerical CV tool generate such viewpoints: S-PLUS and ECOBWEB. However, to help guide the generation of viewpoints that reflect designers' knowledge needs, it is necessary to provide a means whereby users can define their perspectives.

S-PLUS is capable of generating empirical equations according to designers' requirements. However, the process by which this is achieved is cumbersome. Consequently, S-PLUS requires the provision of an operation that more easily renders empirical equations based on the attributes and past designs focused on by a user. These attributes can then be used as variables of the required empirical equation and the specified past designs used during the regression process.

The functionality of ECOBWEB requires extension so the designers can define their perspectives by choosing attributes from the set of past designs, specifying a viewpoint (if any) to be nested, and defining the groups of past designs associated with the specified viewpoint to be nested. These attributes, viewpoint and groups of designs can then be used to generate customised viewpoints of experiential knowledge.

8.3 Rationalisations

ECOBWEB generates, represents and utilises a single viewpoint of experiential knowledge. However, in Chapter 4.2, the existence and utilisation of nested viewpoints in design have also been identified. Therefore, to capitalise on the existing functionality and incorporate the generation, representation and utilisation of nested viewpoints, ECOBWEB's representation and processing capabilities needs to be extended to. The resulting representation formalism is termed here as a **rationalisation**.

8.3.1 Data Structure

The data structure of rationalisations is an extension to ECOBWEB's conceptual hierarchy data structure. Figure 30 illustrates the complete rationalisation structure within which ECOBWEB's data structure resides (indicated by the structures within the dashed boundary). In total, rationalisations consist of five data sub-structures; **RATIONALISATION**, **UNIT-HIERARCHY**, **CONCEPTUAL-HIERARCHY**, **CONCEPT** and **CONCEPT-INFO**.

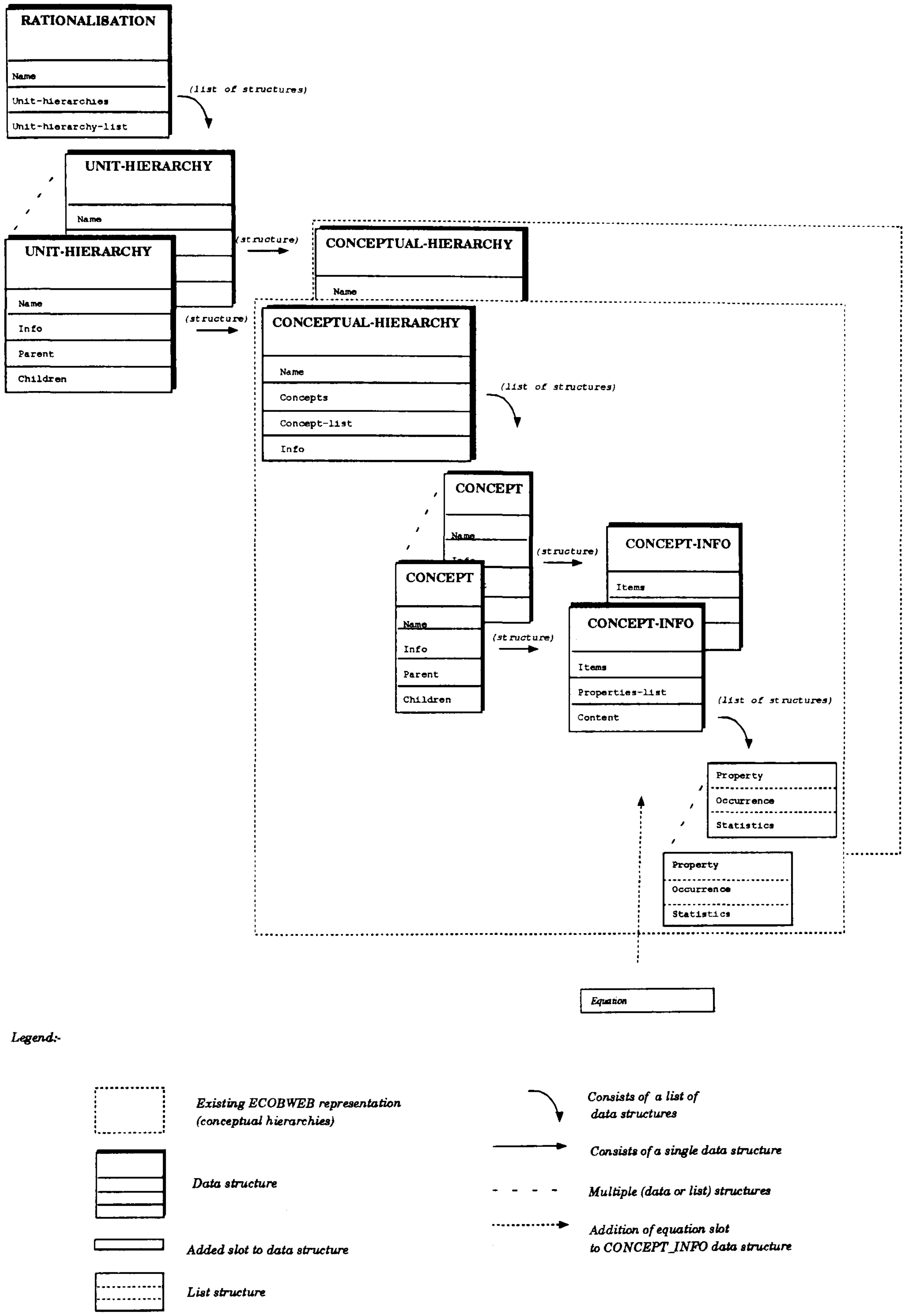


Figure 30: Rationalisation data structure showing extensions to ECOBWEB's conceptual-hierarchy data structure

RATIONALISATION and **UNIT-HIERARCHY** represent additional structures that have been added to ECOBWEB's conceptual-hierarchy data structure to facilitate the representation of nested viewpoints. *Equation* represents an additional slot that has been added to ECOBWEB's concept information structure called CONCEPT-INFO to store rendered empirical equations. Consequently, the rationalisation data structure is more powerful, since it augments the existing data structure (which represented only single viewpoints) to incorporate the representation of nested viewpoints of experiential knowledge and desired empirical equations that are extracted from specific groups of past designs using S-PLUS.

The following explains the content of each data sub-structure.

RATIONALISATION

Data structure detailing the components of a viewpoint. If a viewpoint is single, only one unit-hierarchy exists in the viewpoint. Alternatively, if a viewpoint is nested then multiple unit-hierarchies exist in the viewpoint.

- The `Name` is a slot containing the default name of the rationalisation. This name is set automatically to be NESTED.
- The `Unit-hierarchy` slot contains a series of unit-hierarchy data sub-structures and represents the components (i.e. single viewpoints) that make up a nested viewpoint.
- The `Unit-hierarchy-list` slot is a list of the unit-hierarchy names stored in the `Unit-hierarchy` slot.

UNIT-HIERARCHY

Data structure describing the relative placement of a unit-hierarchy in a rationalisation, i.e. the parents and children of a unit-hierarchy.

- The `Name` is a slot containing a uniquely generated name of a unit-hierarchy. It is important to uniquely name each generated unit-hierarchy so that they can be uniquely identified and used as components of a rationalisation. The name is of the form EXPERIMENT-X, where X is an integer. Therefore, an example of a unit-hierarchy name is EXPERIMENT-1.
- The `Info` slot contains one data sub-structure that describes this unit-hierarchy using the CONCEPT-HIERARCHY data structure of ECOBWEB.
- The `Parent` slot contains the name of the unit-hierarchy that is a parent to this unit-hierarchy.
- The `Children` slot contains the names of the unit-hierarchies that are children of this unit-hierarchy.

CONCEPTUAL HIERARCHY

Data structure detailing the components of a unit-hierarchy, i.e. the concepts that make up a single viewpoint.

- The `Name` is a slot containing the unique name of a tree in which the prefix of this name is EXPERIMENT-.
- The `Concepts` slot contains a series of concept data structures.
- The `Concepts-list` slot is a list of the concept names stored in the `Concept-list`.
- The `Info`

CONCEPT

Data structure describing the relative placement of a concept in a conceptual hierarchy, i.e. the parents and children of a concept.

- The `Name` is a slot containing either a uniquely generated name of a concept in which the prefix of this name is G (reflecting a 'group') or the name of a past design (e.g. EX-69). An example of a concept name is G45.
- The `Info` slot is one data structure that describes this concept using the CONCEPT-INFO data structure of ECOBWEB.
- The `Parent` slot contains the name of the concept that is the parent to the concept.
- The `Children` slot contains the names of the concepts that are children of this concept.

CONCEPT-INFO

Data structure representing the description of a concept, i.e. the attributes and values used to describe a concept.

- The `Items` slot represents the number of past designs that are members of a generated concept.
- The `Properties-list` slot contains a list of attribute names that describe the concept.
- The `Content` slot is a list structure containing information concerning each attribute in the `Properties-list`. This list structure is made up of an attribute name (i.e. `Property`), the number of times this attribute has been used in the children of this concept (i.e. `Occurrences`) and a list of attribute value information (i.e. `Statistics`) such as the average value, the standard deviation and the total number of past designs that are members of this concept.
- The `Equation` slot represents an additional slot to the original ECOBWEB data structure and is used to detail an empirical equation associated to the concept.

Figure 31 illustrates the nested viewpoint presented in Chapter 7, Figure 14 as a rationalisation data structure. This RATIONALISATION is a nested one consisting of four UNIT-HIERARCHIES called EXPERIMENT, EXPERIMENT-1, EXPERIMENT-2 and EXPERIMENT-3. EXPERIMENT is the root UNIT-HIERARCHY, indicated by the value nil as its `Parent` slot value, and has 3 child UNIT-HIERARCHIES, i.e. EXPERIMENT-1, EXPERIMENT-2 and EXPERIMENT-3. In addition, this root UNIT-HIERARCHY has three components (i.e. Group A, B and C) that are listed in the CONCEPTUAL-HIERARCHY structure. These components are groups of Francis turbine designs called Group A, B and C, are represented by CONCEPTS called Group A, B and C respectively, and described in their respective CONCEPT-INFO structures using the attribute *date of design.* The remaining UNIT-HIERARCHIES (i.e. EXPERIMENT-1, EXPERIMENT-2 and EXPERIMENT-3) are children of the UNIT-HIERARCHY called EXPERIMENT and are generated by focusing on *specific speed* and *design head* for each concept (i.e. Group A, Group B and Group C) of this UNIT-HIERARCHY. This results in three individual CONCEPTUAL-HIERARCHIES each containing one CONCEPT called Groups D, E and F respectively.

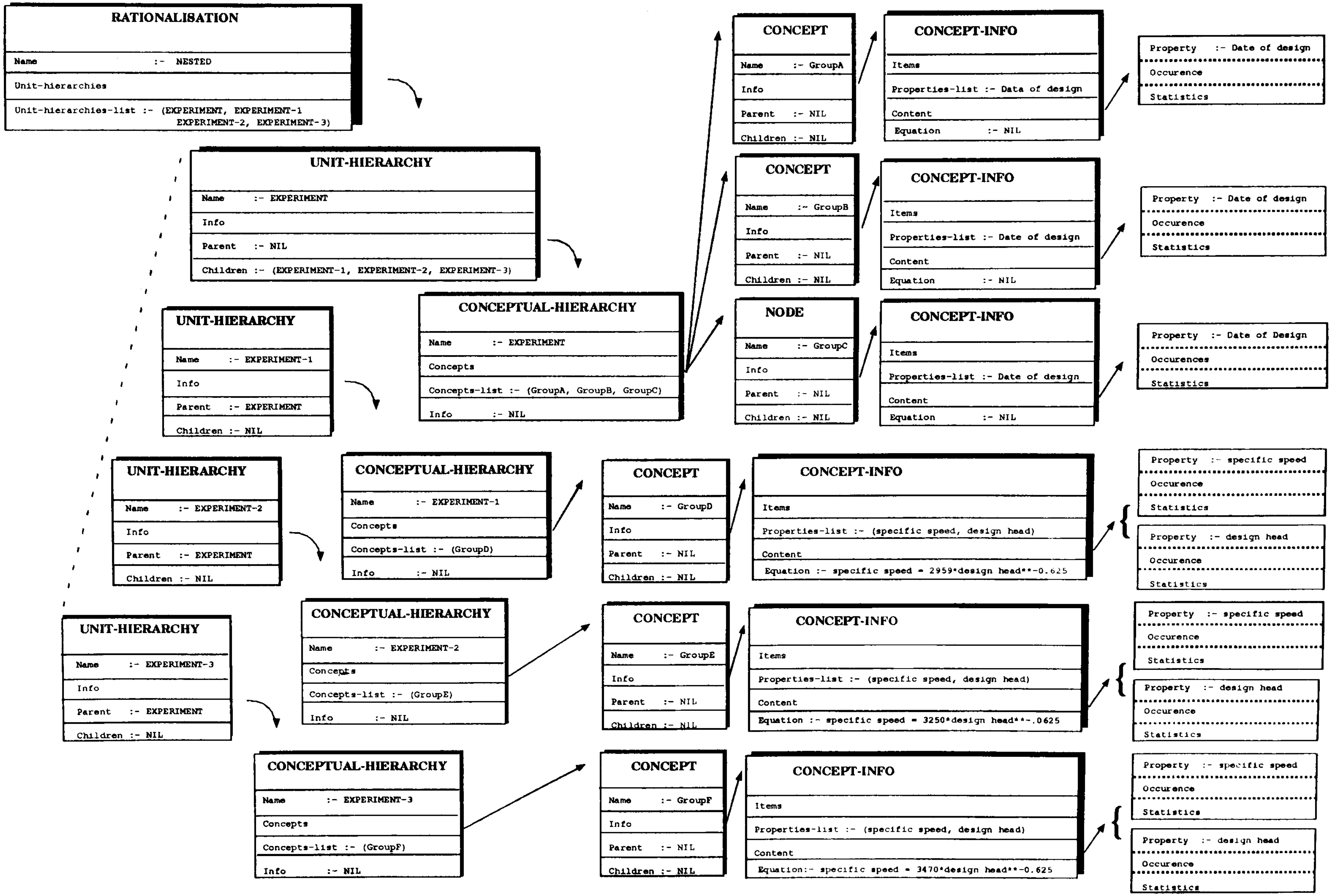


Figure 31: Example of nested viewpoint, presented in Figure 14, represented using rationalisation structure

]

8.3.2 Generation and Utilisation of Rationalisations

The proposed CV tool is to generate viewpoints given a definition of a designer's perspective and a set of past designs, i.e. customised viewpoints. This is to be achieved using the basic functionality available in ECOBWEB. The details of the processes used to generate and utilise single and nested viewpoints are slightly different.

A single viewpoint is to be generated using the ECOBWEB subsystem's learning module; however, *perspectives* (as discussed in Section 8.2) are to guide learning, i.e. by specifying the subset of design attributes to be focused upon. Similarly, nested viewpoints are to be generated using the same basic ECOBWEB functionality; however, the designer's perspective now should consists of (a) an existing viewpoint, (b) a group name (or list of group names) of past designs associated with this viewpoint and (c) a subset of design attributes, all to be used to generate a nested viewpoint. Consequently, nested viewpoints are proposed to be generated from existing viewpoints.

To find the past design most similar to the new design, the proposed CV tool first searches through the root unit-hierarchy of a chosen rationalisation using ECOBWEB's basic prediction module (see Chapter 7.1.3). If the identified similar design belongs to a group that is nested, the search process continues - this time using the associated unit-hierarchy. Similarly for this unit-hierarchy, if the identified similar design belongs to a nested group, this process is repeated until the identified similar design is no longer nested.

8.4 Displaying Viewpoints

The display of viewpoints is to be achieved by a system called GRAPHER [145]. This is the LispView Grapher toolkit and facilitates 'graph' display and editing functionality. However, in the proposed implementation GRAPHER's display capabilities only are used to visually represent viewpoints of experiential knowledge.

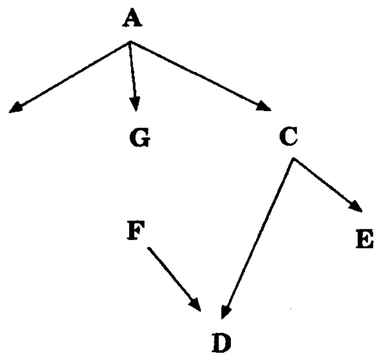
8.5 Rendering and Abstracting the Domain Model

Rendering the domain model

The DESIGNER subsystem builds a design domain model from a domain description consisting of experiential knowledge. This experiential knowledge is normally defined manually but in the proposed CV tool is to be rendered using S-PLUS and ECOBWEB and represented using the rationalisation data structure as described in Section 8.3.1.

Abstracting the domain model

A design domain model, characterised by empirical equations, can be represented as a *dependency network*. This network consists of nodes and links to represent design attributes, and the dependency between attributes respectively. For example, Figure 32 illustrates a dependency network of some theoretical domain consisting of the attributes *A, B, C, D, E, F* and *G*. This domain consists of three empirical equations for calculating A, C and F, and the network represents the attribute dependencies originating from these three equations, i.e. the dependency *A* has on *B, G* and *C*, *C* has on *D* and *E*, and *F* has on *D*

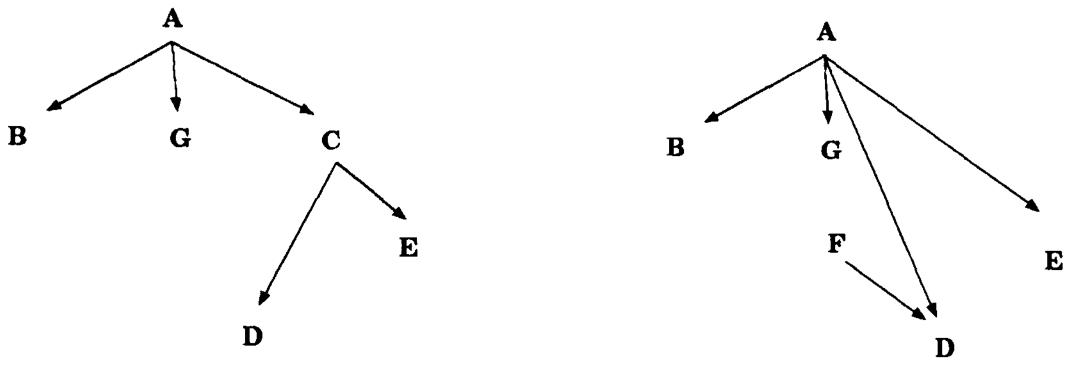


domain description of network is :-

A = f(B,G,C), C = f(D,E), F = f(D)

Figure 32: Dependency network of a theoretical domain

This theoretical domain can be abstracted and Figure 33 illustrates a number of such possible abstractions.

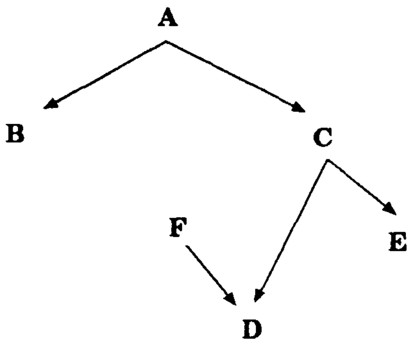


domain description of network is :-
A = f(B,G,C), C = f(D,E)

(a)

domain description of network is :-
A = f(B,G,D,E), F = f(D)

(b)



domain description of network is :-
A = f(B,C), C = f(D,E), F = f(D)

(c)

Figure 33: A selection of possible abstractions of a dependency network illustrated in Figure 32

Figure 33(a) illustrates an abstraction of the original domain consisting of two empirical equations for calculating the attributes A and C. In this example, the original domain has been abstracted by removing the knowledge associated with estimating attribute F, i.e. removing an empirical equation. Alternatively, the original domain can be abstracted to the domain illustrated in Figure 33(b) or Figure 33(c). In Figure 33(b), the domain model has been abstracted by removing an empirical equation to calculate C and re-generating a new equation for A that incorporates B, G, D and E. In Figure 33(c), rather than removing empirical equations, this figure illustrates how the abstraction and generalisation of an empirical equation can result in an abstraction of a domain.

Abstracting and generalising an empirical equation is characterised by the removal of variables and subsequent re-generation of an equation to quantify the relationship between the remaining variables. In this example, the empirical equation used to calculate the attribute A is abstracted by removing the attribute G.

Consequently, the numerical CV tool is required to support two types of abstraction: abstraction of empirical equations and abstraction of the domain model. The abstraction of the domain model is achieved in three ways. Table 19 summarises these types of abstraction. Note that an abstraction of an equation with only one input variable does not exist, e.g. see $F = f(D)$ in Table 19.

Type of Abstraction	**Definition**	**Example**	
		Original	Abstraction
Empirical Equation	Removal of variables	$A = f(B,G,C)$	$A = f(B,C)$
			$A = f(B,G)$
			$A = f(G,C)$
			$A = f(B)$
			$A = f(G)$
			$A = f(C)$
		$C = f(D,E)$	$C = f(D)$
			$C = f(E)$
		$F = f(D)$	does not exist
Domain Model	(a)i Removal of empirical equations (no regeneration of equations)	$A = f(B,G,C)$ $C = f(D,E)$ $F = f(D)$	(a)i $A = f(B,G,C)$ $C = f(D,E)$
	(a)ii Removal of empirical equations (regeneration of equations)		(a)ii $A = f(B,G,D,E)$ $F = f(D)$
	(b) Abstraction of empirical equations		(b) $A = f(B,C)$ $C = f(D,E)$ $F = f(D)$
	(c) Removal and Abstraction of empirical equations		(c) $A = f(B,G,C)$ $C = f(D)$

Table 19: Examples of empirical equation and domain model abstraction

8.6 Summary

Three existing systems (DESIGNER, S-PLUS and ECOBWEB) are to be used to develop a numerical CV tool that supports the effective utilisation of experiential design knowledge. To achieve this five necessary functions need to be addressed. First, a means of transferring information from ECOBWEB to S-PLUS, S-PLUS to DESIGNER and DESIGNER to ECOBWEB has to be developed to help integrate the extraction and utilisation of applicable experiential knowledge into one complete system. Second, designer's perspectives should be supported and used to generate viewpoints of experiential knowledge to reflect designers' knowledge needs. Third, the functionality of ECOBWEB requires extension to support the generation, representation and utilisation of single and nested viewpoints of experiential knowledge. Fourth, the viewpoints are to be displayed using the LispView GRAPHER. Fifth, a numerical domain model is to be rendered using S-PLUS and ECOBWEB, and its abstraction is to be achieved in three ways: (a) the removal of empirical equations with and without the regeneration of equations, (b) the abstraction of empirical equations, and (c) the removal and abstraction of empirical equations.

9 The PERSPECT System

PERSPECT is a numerical CV tool, which aims to support the effective utilisation of experiential knowledge in engineering design. Designers can use PERSPECT to extract and utilise general knowledge originating from past designs. In other words, they are not hindered by knowledge engineers' preconceptions of what experiential knowledge will be used in design; they are free to generate and utilise knowledge, according to their own particular needs, from a source of past designs.

PERSPECT addresses each of the requirements defined in Chapter 4. From a numerical aspect, the system utilises multiple forms (i.e. *empirical equations* and *generalisations*) originating from a single source of experiential knowledge (i.e. *past designs*); supports exploration, synthesis and evaluation design activities, and acquisition, generation and modification learning activities, through the application of *regression analysis* and *concept formation* techniques; and supports the generation, representation and utilisation of customised single and nested viewpoints. The purpose of this chapter is to present, in more detail, the design and construction of PERSPECT and explain how the system addresses the satisfaction of these requirements.

Section 9.1 presents PERSPECT's system architecture to explain how its components assist in the utilisation of multiple forms of experiential knowledge. Section 9.2 discusses how PERSPECT maps a designer's knowledge needs onto viewpoints of experiential knowledge. Section 9.3 explains how PERSPECT supports the coupling of design and learning activities by presenting PERSPECT's activity diagram. Finally, the chapter closes in Section 9.4 with a brief summary of the chapter.

9.1 System Architecture

The PERSPECT system represents two forms of experiential knowledge: empirical equations and generalisations. To support the application of these forms, PERSPECT integrates three existing computer systems: DESIGNER, S-PLUS and ECOBWEB.

Figure 34 illustrates PERSPECT's system architecture to highlight four main points: the interaction the user has with the system, the main components of the system (i.e. S-PLUS [141], ECOBWEB [37], GRAPHER [145] and DESIGNER [34]), the existence of and relationship between generated objects/entities in the system, and the approach to utilising multiple forms of experiential knowledge via the use of generated and precompiled files.

9.1.1 User Interaction

User access to the functionality of the PERSPECT system is via three interfaces: the original S-PLUS and DESIGNER interfaces, and an *additional* window interface that provides access to a portion of PERSPECT functions. The remaining PERSPECT functions are accessed from the Lisp environment shared by DESIGNER. The *additional* interface facilitates (a) the definition of perspectives, (b) the generation and display of resulting customised viewpoints, (c) the saving and reloading of viewpoints to and from files, (d) the loading of sets of past designs, and (e) the search for a past design similar to the new design using a particular viewpoint.

The *additional* interface consists of three windows: a control window (see Figure 35), a past design file information window (see Figure 36), and a viewpoint window (see Figure 37).

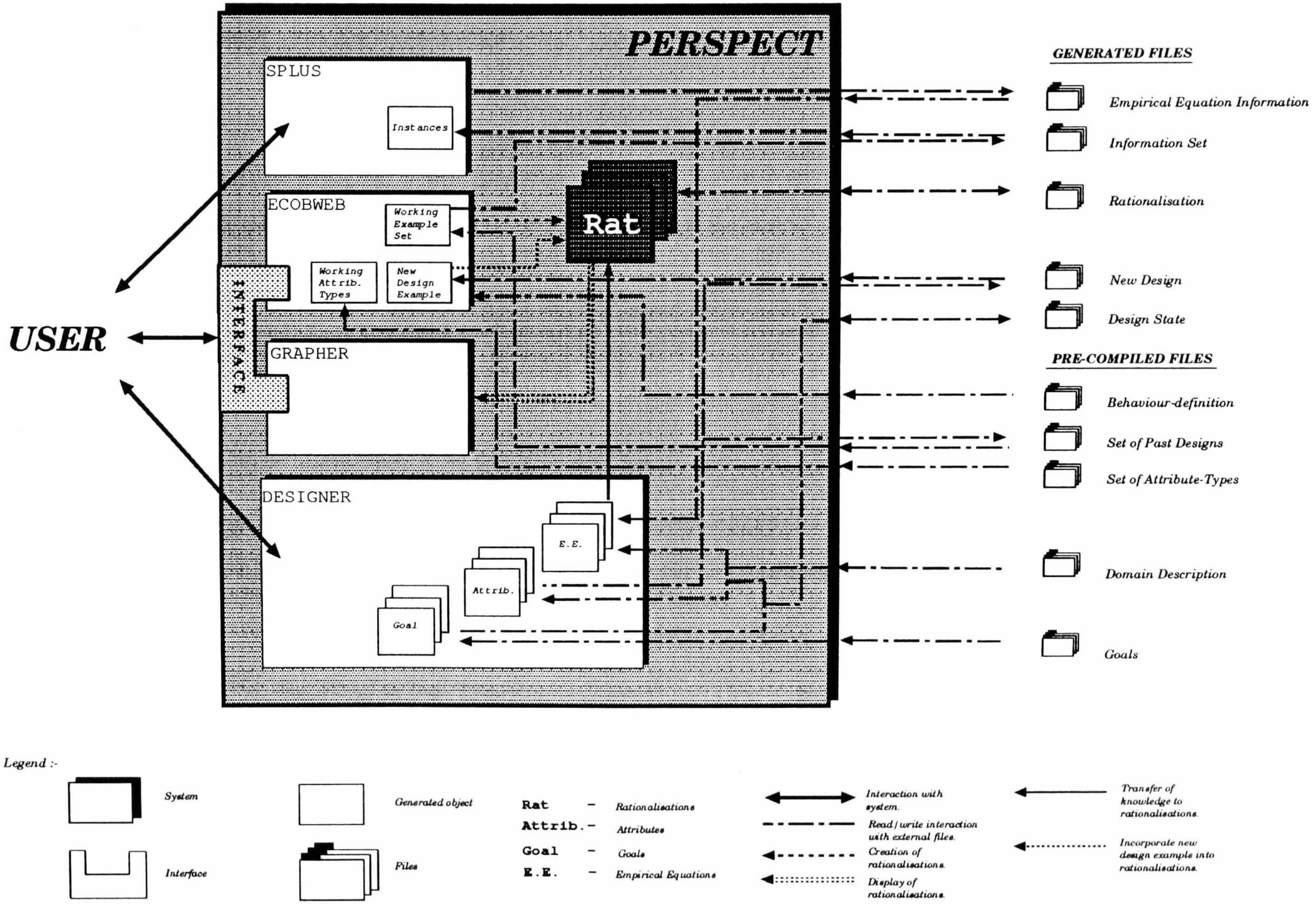


Figure 34: PERSPECT system architecture

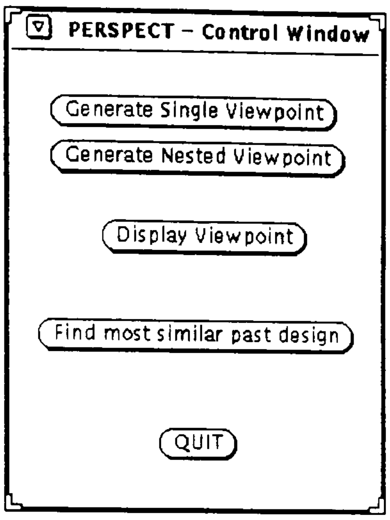


Figure 35: Control window of *additional* interface

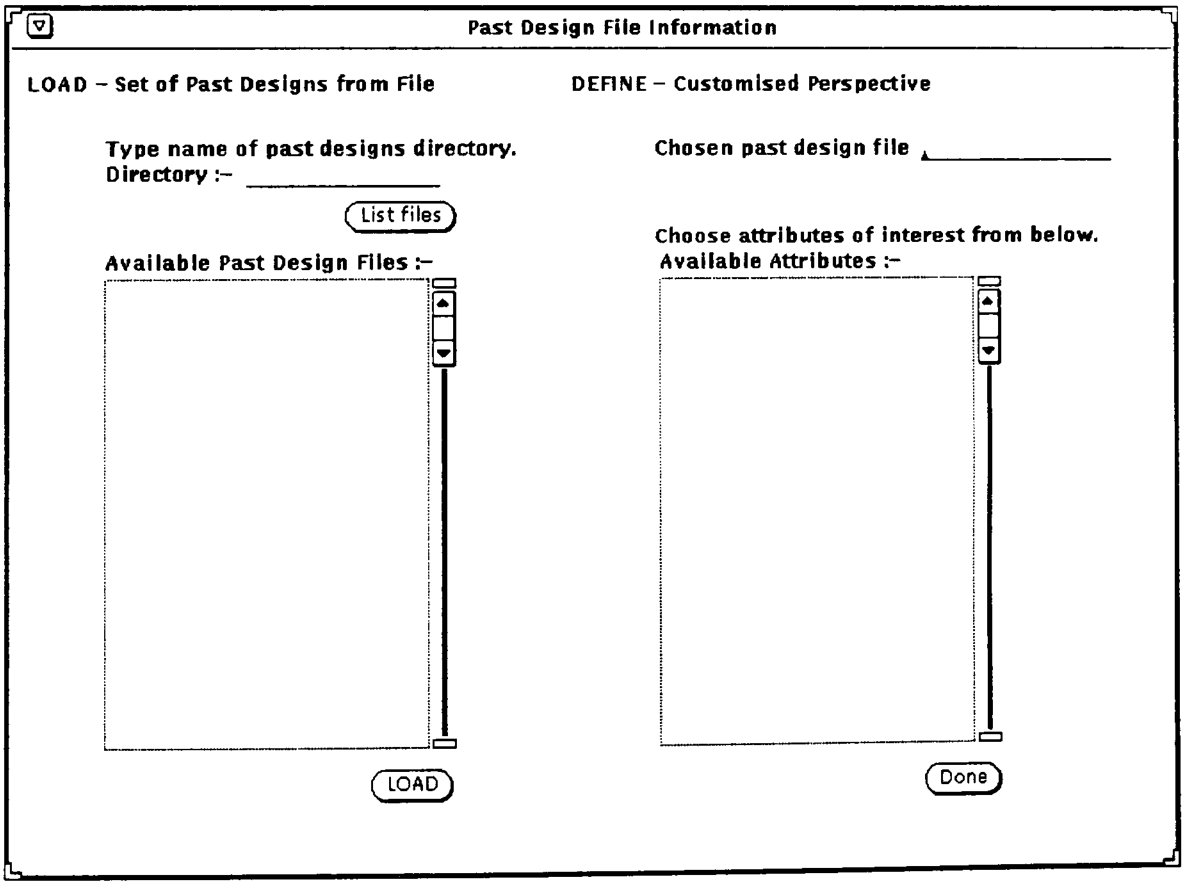


Figure 36: Past design file information window of *additional* interface

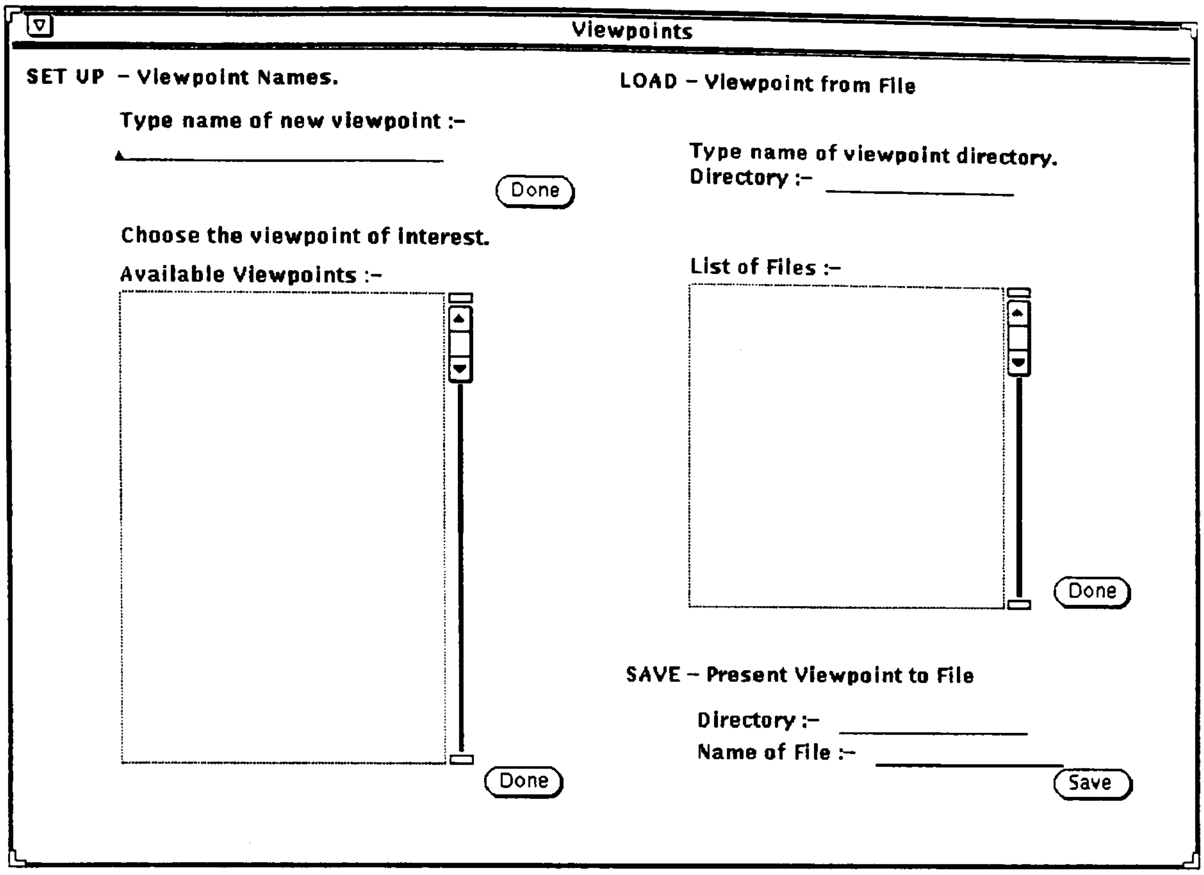


Figure 37: Viewpoint window of *additional* interface

The control window provides five push buttons, which when activated execute functions in PERSPECT, i.e. generating single and nested viewpoints, displaying viewpoints, finding the most similar past design and quitting the system. The past design file information window lists the filenames of available sets of past designs, e.g. tankers, bulkers, etc. After choosing a set of past designs to load into PERSPECT, the user is then presented with a list of attributes that describe the associated past designs. From this list, the user can then define a perspective by choosing attributes to be used to generate a viewpoint. The viewpoint window is used to name, select, load and save viewpoints.

9.1.2 Main Components

Chapters 7 and 8 have already discussed the main components of the PERSPECT system, and it is emphasised here that the complete functionality of DESIGNER, S-PLUS and ECOBWEB is available to the user via the PERSPECT system.

9.1.3 Entities

Figure 34 illustrates eight important entities that are represented in PERSPECT: *goals, attributes, empirical equations, working attribute types, new design example, working example set, instances* and *rationalisations.* Goals, attributes, and empirical equations are entities that exist in the DESIGNER subsystem, and represent numerical knowledge associated with a particular design domain and design solution. Working attribute types, new design example, working example set are entities that exist in the ECOBWEB subsystem, and represent attribute-value information associated with the new design and past designs. The instances entity exists in the S-PLUS subsystem, and represents attribute-value information associated with the set of past designs. Rationalisations are entities that exist in the PERSPECT system (see Chapter 8), and are extensions to ECOBWEB's original entity (i.e. concept-hierarchy). PERSPECT uses these rationalisations to represent customised single or nested viewpoints of past design knowledge.

There are three important relationships that exist between these entities: a *creation* relationship between the working example set and rationalisations, an *incorporation* relationship between new design example and rationalisations, and a *transfer* relationship between empirical equations and rationalisations. PERSPECT creates rationalisations using the working example set, and incorporates the new design example into chosen rationalisations to identify the most similar past design. It also transfers known generated empirical equations into the knowledge associated with created rationalisations.

9.1.4 Use of Pre-compiled and Generated Files

The system requires the designer to prepare five file types, signified by the files listed under the **pre-compiled files** heading in Figure 34, from which PERSPECT can automatically generate a further five file types listed under the **generated files** heading of Figure 34. Pre-compiled files define the behaviour definition for ECOBWEB, and provide a source of experiential knowledge (i.e. set of past designs), a description of a design domain and a definition of a new design's goals. These files are utilised by PERSPECT's DESIGNER and ECOBWEB subsystem functionality. Generated files allow experiential knowledge from one system to be used by another, and are utilised by new functionality provided by PERSPECT. In other words, existing ECOBWEB and DESIGNER functionality reads in *set of past designs, set of attribute-types and domain description* file. However, the reading and writing of *empirical equation information, rationalisations and design state* files and the writing of *information set* and *new design* files are achieved by new PERSPECT functionality. The following bulleted list describes the content of each of these files.

Pre-compiled Files

- *Behaviour-definition* file consists of global variables that define the behaviour of the ECOBWEB subsystem, i.e. how learning and prediction is carried out.
- *Set of past designs* file consists of a number of past designs described by attributes and associated values in ECOBWEB syntax. Each file can contain designs of a particular design domain type, e.g. bulker or tanker ship types.
- *Set of attribute-types* file consists of the attributes used to describe the past designs of a particular design type along with the attribute's type (i.e. continuous).
- *Domain description* file consists of a description of the domain of interest (i.e. attribute names, meanings, units, empirical equations, unreliabilities of equations, etc.). Different domain descriptions exist for each design domain type. For example, knowledge associated with the house type bungalow is very different from the knowledge of the terraced type. Therefore, the user of PERSPECT is required to ensure the appropriate design domain type.
- *Goals* file consists of the definition of a number of goals the new design is required to satisfied. (Goals can be interactively defined; however, for convenience they can be stored in a file)

Generated Files

- *Empirical equation information* file consists of information concerning rendered equations (i.e. equation variables and coefficients). The content of this file is generated from S-PLUS and used to define a new empirical equation in DESIGNER.
- *Information set* file consists of information concerning the working example set. However, the format of the file is in S-PLUS readable syntax. This file is used as a source of experiential knowledge from which S-PLUS can generate empirical equations.
- *Rationalisation* file consists of information describing rendered viewpoints of experiential knowledge.
- *New Design* file consists of attribute information detailing the state of the new design, existing in DESIGNER, written in ECOBWEB syntax. The design model can be partially (i.e. not all attribute values known) or fully (i.e. all attribute values are known) described.
- *Design state* file consists of a record of DESIGNER's working environment, i.e. all known goals, attributes (instantiated and uninstantiated) and empirical equations used in the domain description. In other words, this file includes information of the domain description and the design model for the new design.

9.2 Mapping of Needs to Viewpoints

PERSPECT provides a designer with a means of mapping knowledge needs onto viewpoints of experiential knowledge by supporting the definition of designers' perspectives. Perspectives are definitions of a designer's interest, governed by the needs for knowledge, and are used to generate viewpoints. Supporting the definition of perspectives is important as it promotes the use of customised viewpoints of experiential knowledge that reflect the unpredictable or particular needs of a designer and not predetermined viewpoints modelled by a knowledge engineer.

Unpredictable designer needs can arise out of the lack of knowledge. The lack of knowledge can have a detrimental effect on the success of a design; e.g. by preventing the execution of design, reducing the quality of the new design or increasing the design cycle time while relevant knowledge is solicited and assimilated [20]. PERSPECT remedies this situation by opportunistically providing additional knowledge, which either satisfies (see option 2 of Chapter 10.3) or temporarily avoids (see option 3 of Chapter 10.3) the need for such knowledge. PERSPECT's generated viewpoints can include generalisations of a set of past designs (according to specified attributes describing a perspective), empirical equations or abstractions of these equations. Therefore, PERSPECT presents a designer with a tool to render previously implicit knowledge in an explicit form of generalisations and empirical equations.

9.3 Design and Learning Activities

PERSPECT can be used to define perspectives, generate empirical equations and generalisation, render a domain model, and develop a numerical design solution. In other words, customised viewpoints of experiential knowledge can be generated using PERSPECT's regression analysis and/or concept formation functionality, and can be used to (a) render empirical equations to be used to build a domain model and/or (b) define a design solution (ie. design model). These two uses of PERSPECT requires the coupling of design and learning activities. In Chapter 7, PERSPECT's subsystems' activity diagrams were presented (see Figures 22, 26 and 28). In this section, PERSPECT's functionality is explained by integrating these activity diagrams into one (see Figure 38). The resulting activity diagram illustrates the coupling of design and learning activities in PERSPECT and thereby helps to highlight the additional support provided to the designer.

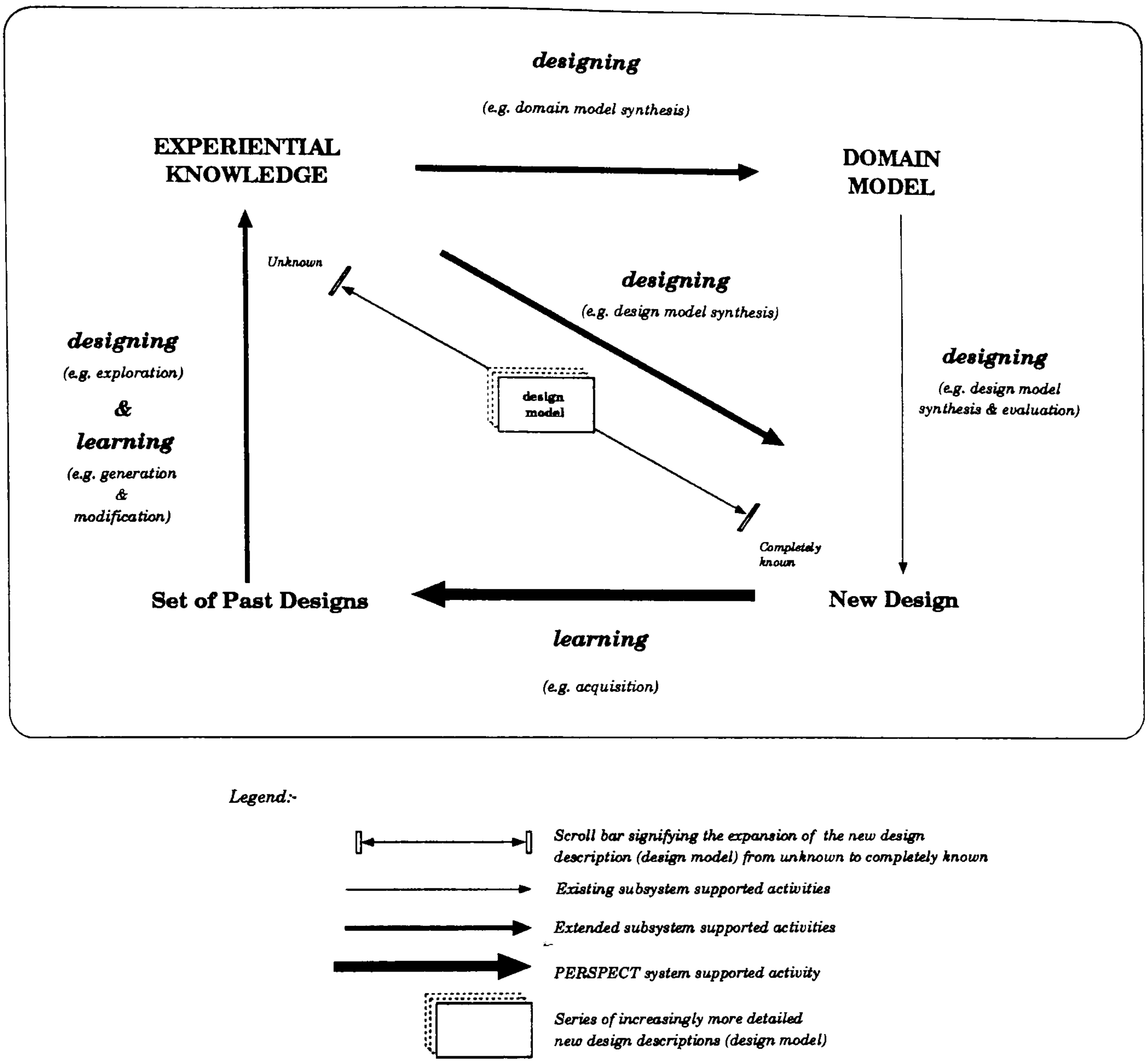


Figure 38: PERSPECT's activity diagram: coupling design and learning

PERSPECT supports the exploration of a design domain to render experiential knowledge. This knowledge can then be directly used to build a domain model from which a design solution can be developed. Figure 38 illustrates this coupling of design and learning by the **Set of Past Designs** to **EXPERIENTIAL KNOWLEDGE** to **DOMAIN MODEL** activity path.

To develop a design, PERSPECT's DESIGNER subsystem always choses the most reliable empirical equation. If two equations are of equal unreliability, the equation requiring the least amount of information is chosen. If an equation is chosen where the input attribute values are not yet known by the designer, 3 options are available: manually input values, find a suitable generalisation applicable to the new design, or remove unknown attributes for equations and re-generate a new abstracted empirical equation. The first option (shown in the figure by the **DOMAIN MODEL** to **New Design** design activity path) does not employ any of PERSPECT's additional features that support the effective *application* of experiential knowledge. However, the remaining two options are more interesting.

The designer can use PERSPECT to generate viewpoints of experiential knowledge from which information of the most similar past design or generalisations of similar past design groups can be identified and used to estimate unknown attribute values. This option is represented by the **Set of Past Designs** to **EXPERIENTIAL KNOWLEDGE** to **New Design**. Alternatively, new or less complex empirical equations can be rendered to simplify and/or supplement the existing domain model. This option is represented by the **Set of Past Designs** to **EXPERIENTIAL KNOWLEDGE** to **DOMAIN MODEL** path. Abstractions of empirical equations mean that attribute values can be assigned with fewer required attributes. For example, Figure 39 shows two examples of abstracting and generalising the same empirical equation, i.e. an equation to estimate CB that is dependent one two input variables, L and VS. These examples show the resulting equations when removing one input variable from an equation. In Figure 39(a) the L variable has been removed and in Figure 39(b) the VS variable has been removed. The user is free to remove any variable for an empirical equation; or, alternatively, DESIGNER can be used to determine the least influential input variable of an equation and thereby suggest that as the variable most suitable for removal. Therefore, the user can utilise PERSPECT to help define the design model by (a) effect propagation using DESIGNER, (b) attribute value estimation using ECOBWEB, or (c) empirical equation rendering and application using S-PLUS and DESIGNER respectively.

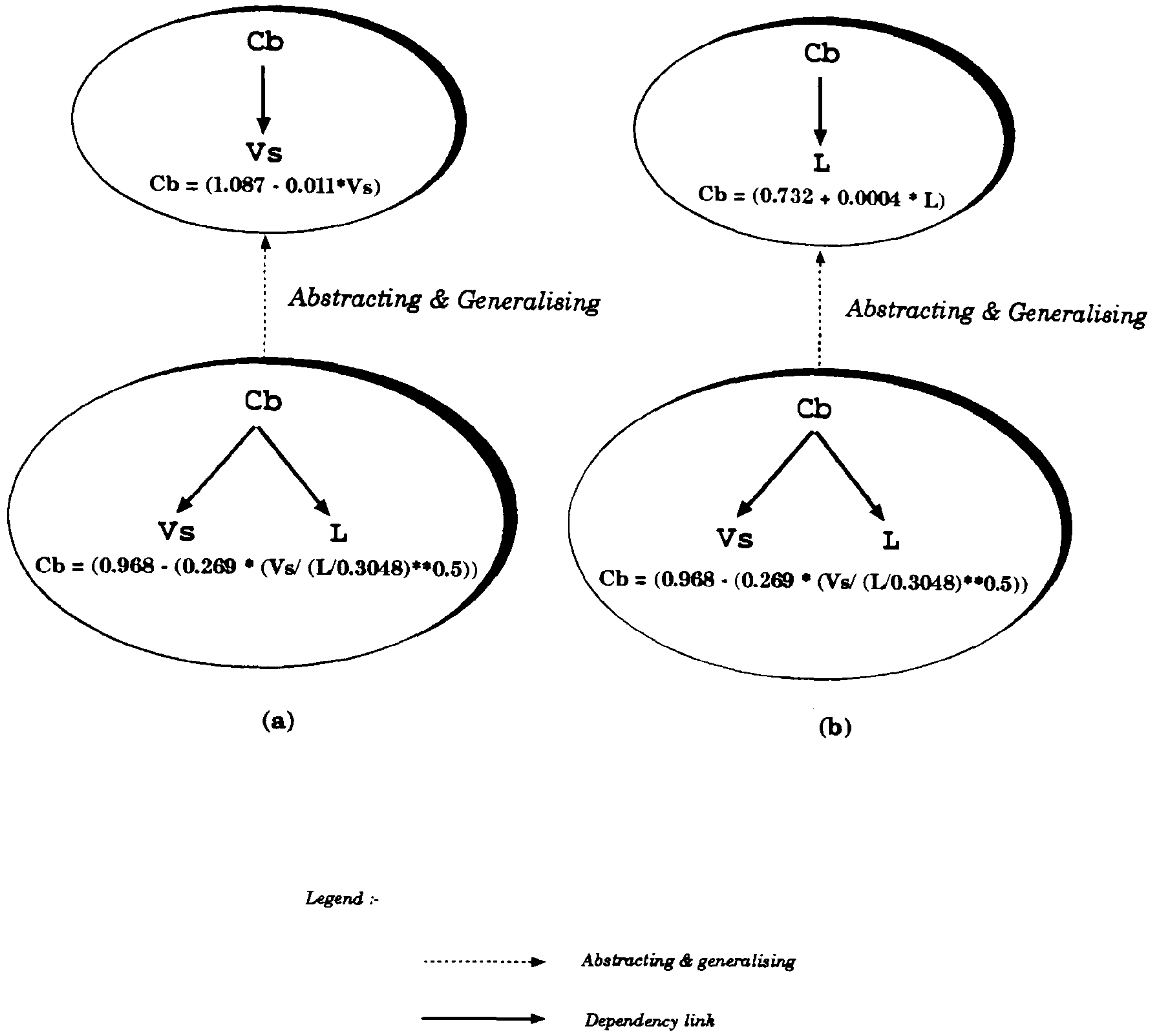


Figure 39: Two empirical dependency networks showing different abstractions of the same empirical equation

Consequently, recalling the coupling diagram of design and learning, presented in Chapter 4, Figure 6, it can be seen that PERSPECT supports both types of design/learning loops, i.e. looping to develop a design solution and looping to develop experiential knowledge.

9.4 Summary

This chapter has presented the PERSPECT system, a CV tool proposed to support the effective utilisation of experiential knowledge in numerical design. The chapter has discussed PERSPECT, in relation to the general support requirements as defined in Chapter 4, by presenting PERSPECT's system architecture, activity diagram and the mapping of design needs to viewpoints via perspectives, respectively.

The activity diagram presented in Section 9.3 provides the basis for the worked example in Chapter 10, which will illustrate the support PERSPECT provides in design during domain model preparation (ie. building and checking) and design model initiation.

10 Utility of the Customised Viewpoint Approach in Numerical Design

The aim of this thesis has been to promote the effective utilisation of experiential design knowledge by supporting its extraction and use, guided by designers' knowledge requirements. To achieve this, a new approach called 'customised viewpoint' has been presented as complimentary to the CAD philosophy of "design assistance". The realisation of this approach is proposed to provide designers with tools that:

- automatically update and evolve experiential knowledge,
- support the automatic generation of appropriate viewpoints of experiential knowledge according to designers' perspectives, and
- support the utilisation of applicable experiential knowledge in design.

Chapter 9 has presented the PERSPECT system, a CAD system that has been implemented to illustrate and evaluate the utility of such a 'customised viewpoint' approach in numerical design. Ship design has been chosen to demonstrate the utility of this approach for three reasons. First, the utilisation of experiential knowledge is a generally accepted knowledge resource used in ship design development. Second, a paper database of previously designed ships (bulkers) is available, and third a source of previously generated empirical equations is readily available for implementation purposes. However, it is important to note that the ideas presented in this thesis are of relevance to any design domain. Therefore, although the demonstration and evaluation of the 'customised viewpoint' approach is carried out within the domain of ship design, the applicability of the approach is suited to any other domain where past design information and experiential knowledge has been or could be recorded.

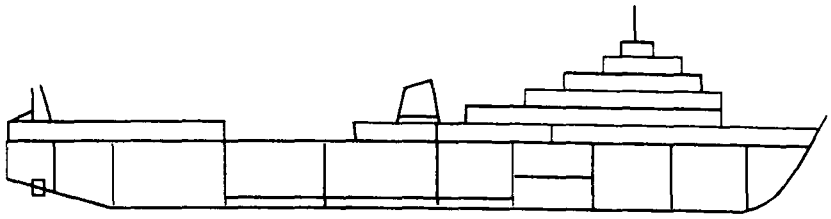
This chapter consists of four main sections. Section 10.1 provides a basic introduction to the ship design process to highlight the utilisation of experiential knowledge. Section 10.2 then illustrates the utility of PERSPECT by introducing two design tasks supported by the system and demonstrating the added functionality available to designers. Section 10.3 reports on an evaluation of the PERSPECT system that involved discussions with two designers to identify the advantages and disadvantages of the system. Section 10.4 closes the chapter with a summary.

10.1 Ship Design

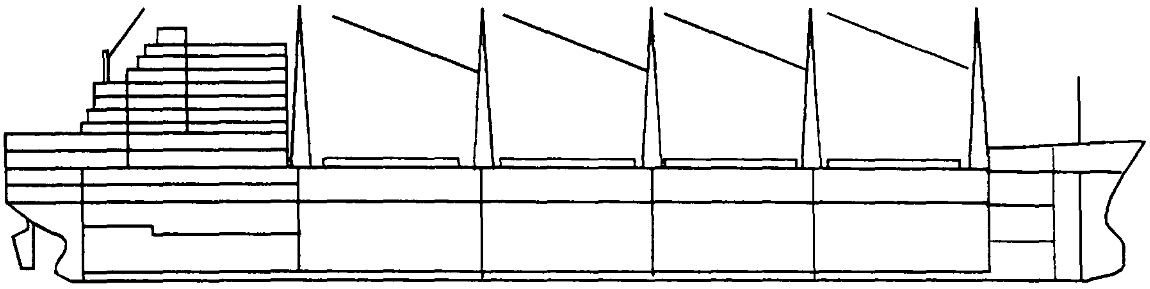
> "Preliminary design by its very nature is perhaps the most subjective aspect of naval architecture relying as it does on the accumulated experience and data of each practitioner" [42]

This quote encapsulates the nature of ship design as depending on experiential knowledge and thereby emphasises its applicability to this research.

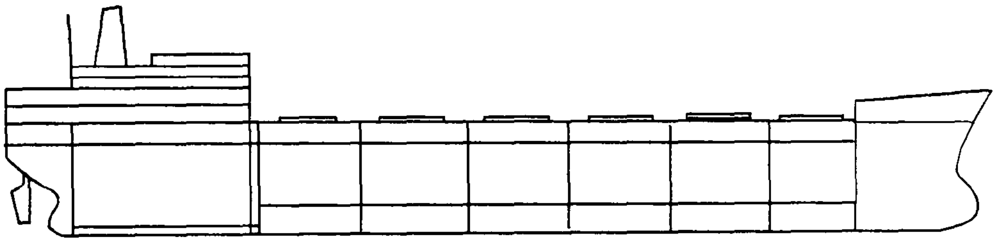
The early stages of ship design involve the identification of a type of ship that is likely to satisfy the general design needs. These design needs can be a specification of the task that the design is required to carry out. For example, to transport goods (i.e. people, oil, gas, coal or bulk), to carry out work in open sea while maintaining a stationary or mobile position, to protect the seas, to provide support for other vessels, etc. The variety of tasks result in a range of ship types that vary in shape and size. Some examples of ship types that satisfy the tasks mentioned earlier are ferries, tankers, bulk carriers, diving support vessels, trawlers, military vessels, and tugs. Examples of some of these ship types are illustrated in Figure 40. The worked example detailed in this chapter focuses upon the preliminary design of a bulker ship type to illustrate the utility of the PERSPECT system in ship design.



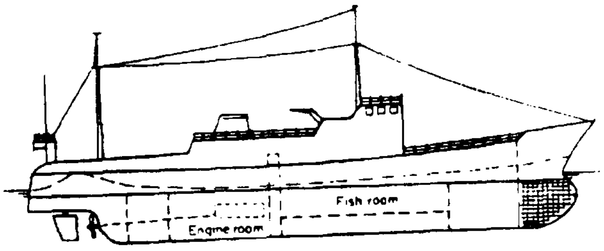
Passenger Ferry



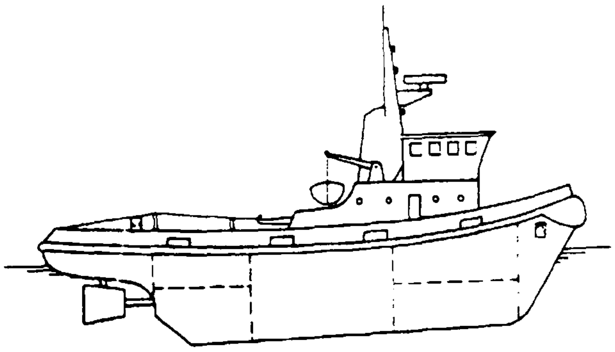
General Cargo Carrier



Bulk Carrier



Stern Trawler



Tug

Figure 40: A selection of typical geometries for a number of ship types: passenger ferry, general cargo carrier, bulk carrier, stern trawler [4] and a tug [4]

The next stage of the ship design process involves selecting suitable design attributes that describe the new ship design, e.g. *dimensions* such as length, breadth, draft and depth, and *particulars* such as block coefficient, displacement and dead-weight. The distinction between *dimensions* and *particulars* is made here to signify attributes that are *measurable* and *calculable*, respectively. Attributes can describe many aspects of a design, e.g. geometry, power, stability, weight, floating condition, etc., and the complexity in developing a new design can be attributed to the interrelation of these ship design aspects. For example, modifying a design's geometry influences its stability and weight. Consequently, the process of selecting suitable values for the ship design's attributes is complex as it requires a compromise among various design aspects to achieve an acceptable design solution.

To simplify this process, designers depend on their understanding of the influences and relationships between attributes, which they accumulate from the analysis of past designs. For example, Figure 41 illustrates the relationship between ship designs' Cb (i.e. the ratio of the volume of a ship to the volume of a rectangular block whose sides are equal to the breadth, the mean draught and a measured length of a ship) and ship designs' Froude Number (i.e. a non-dimensional measure of speed used for comparing ships of different sizes equivalent to $speed/\sqrt{length}$ [42, 146]) plotted by Watson & Gilfillan [42]. Here it can be seen that ship types cluster around certain areas in the graph.

Similarly, grouping past designs according to various attributes can present the designer with valuable information. For example, the following graphs in Figure 42 [146] focus on proportional information (i.e. L/D, L/B and B/T) to illustrate the regions (shaded areas) in which past designs of certain ship types fall.

The recognition of such trends in past designs is a valued asset, and has been utilised in the past by designers trying to 'rationalise' past ship designs [40, 41, 42]. The results can be a series of graphs from which attribute values can be directly 'read off'

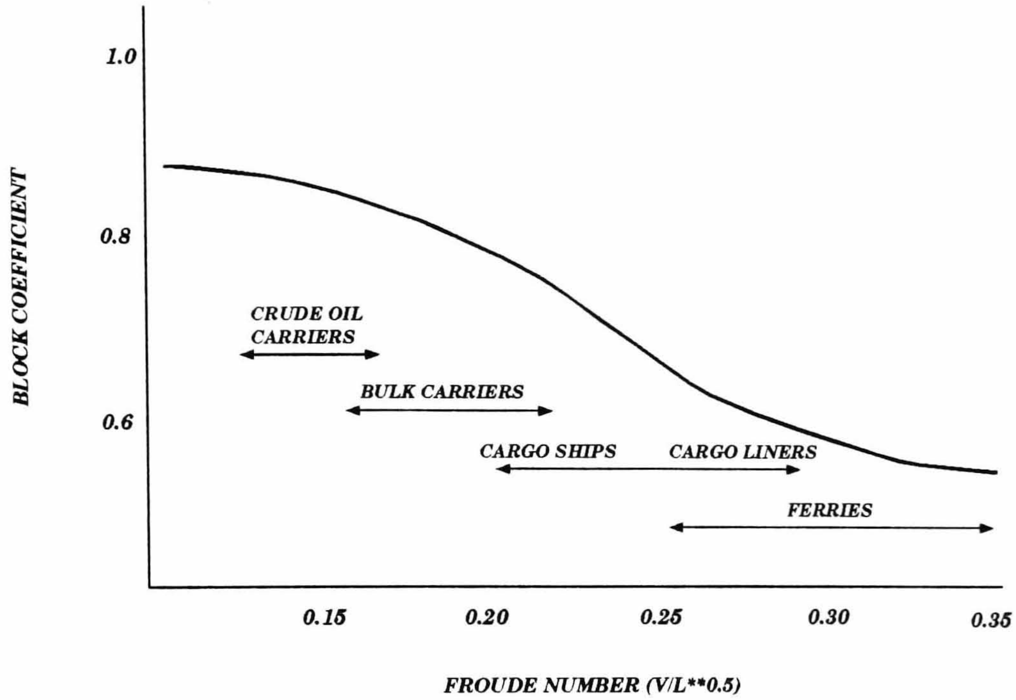


Figure 41: Graph of Cb vs Froude number showing placings/groups of ship design types, adapted from [42]

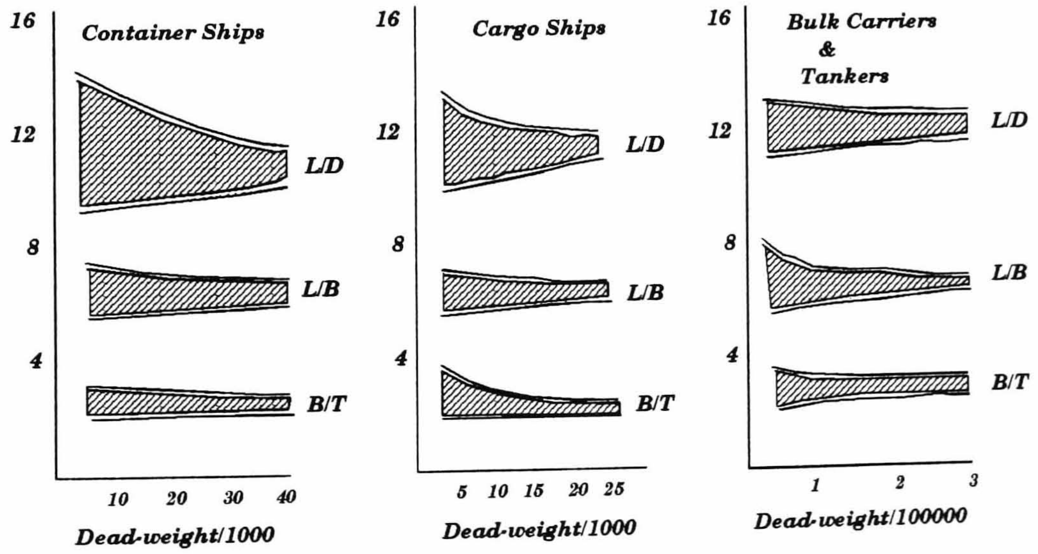


Figure 42: L/D, L/B and B/T ranges for a selection of ship types [146]

by designers (see Figure 43) or the generation of empirical equations (see Table 20) from which values of design attributes can be calculated from known attribute values. Therefore, experiential knowledge is a very important knowledge resource that helps ship designers develop a preliminary design solution.

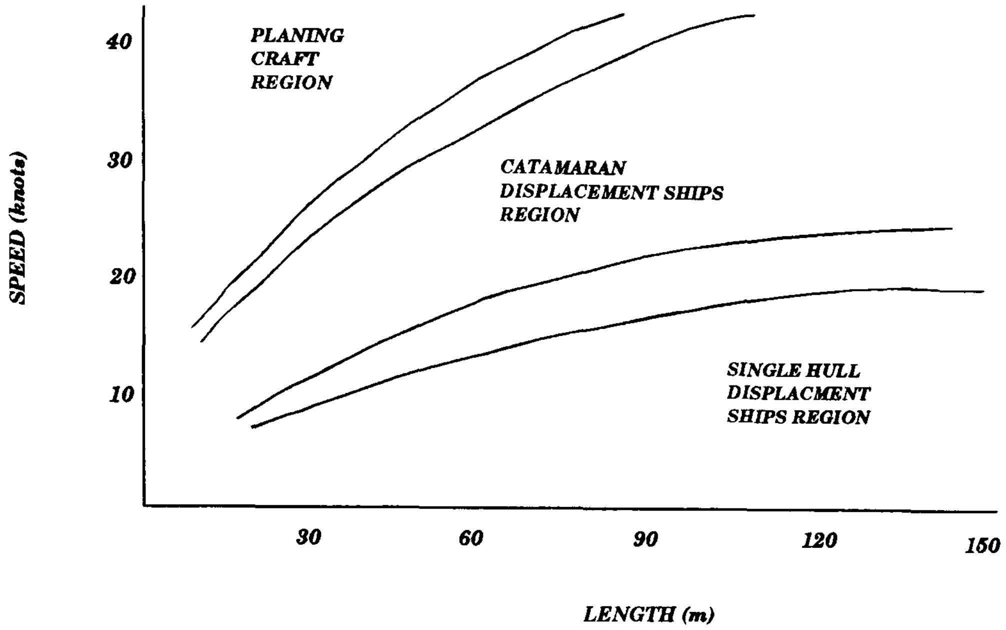


Figure 43: Areas of favourable operation by focusing on speed and length for various hull types [4]

$$Cb = 1.052 - (0.216 \times Vs/\sqrt{L})$$

$$D = 1.41 \times T$$

$$B = (1.27 \times L)/(4 + (0.025 \times (L - 30)))$$

$$L = 6.32 \times (\sqrt{(Vs/(Vs + 2))}) \times \sqrt[3]{disp}$$

where,

Cb = block coefficient, Vs = speed, L = length, D = depth, T = draught, B = breadth

---

Table 20: Empirical equations generated from the analysis of past designs, as used by MacCallum & Duffy [44]

10.2 Utility of the PERSPECT System in Design

This section introduces two basic design tasks in which PERSPECT can assist a designer to effectively utilise experiential knowledge of past designs.

- **Domain Model Preparation** - this involves rendering, or checking and updating pre-compiled experiential knowledge that describes the design domain, to promote the most applicable experiential knowledge to be used to build domain models.
- **Design Model Initiation** - this involves the commencement, continuation and completion of the design process from the definition of the design goals to the initial assignment of suitable estimates for some of the new design's attributes.

These two tasks are discussed by considering their purpose in the design process and the system features that support their operation.

10.2.1 Preparation of the Domain Model

PERSPECT provides support for rendering particular viewpoints from which domain models can be built. As discussed in Chapter 7.1.1, the DESIGNER subsystem requires the manual definition of a domain model, consisting of empirical equations, to support numerical design. However, if no domain model exists, PERSPECT can be used to build up such a model using the experiential knowledge generated from the S-PLUS subsystem. The complete functionality of S-PLUS is available within the PERSPECT system and thus presents designers with a powerful regression analysis tool. The activity of rendering a domain model is a complex one, and depends greatly on designers' understanding of regression techniques and knowledge of the domain to render useful viewpoints of experiential knowledge. However, what will be demonstrated is the support PERSPECT provides when a domain model exists.

Associated with each compiled equation in an available domain model is a measure of its unreliability based on a set of past designs. As time passes, new designs are introduced to the design domain, which can reflect different or more recent trends. Therefore, it is necessary to check the pre-compiled empirical equations in light of these new designs. Alternatively, these equations could have been rendered using a completely different set of designs to which the designer has access. Thus, it is equally necessary for designers to check the pre-compiled equations against their own set of past designs.

The existing domain model [147] was generated using an original set of 20 past designs. Additional past designs, giving an extended set of 31 designs, were available to check this model's applicability. Table 21 indicates the distribution of the original and extended set of past designs utilised along with the year they were delivered to their customer. The extended set of past designs provides more designs that are relatively recent; therefore the pre-compiled empirical equations are required to be checked to determine their suitability to the extended set of past designs now available.

Year of Delivery	Past Designs	
	Original	Extended
1982	2	5
1981	2	6
1980	3	4
1978	1	2
1977	6	7
1976	3	3
1975	2	2
1974	1	2
Total Number of Designs	20	31

Table 21: Number of past designs and distribution of year of delivery

First, the designer has to determine which empirical equations can be checked. It is inappropriate to assume that the attributes available in the set of past designs map onto those available in the domain model. If the attributes used in the set of past designs are not exactly the same as those in the domain model, certain empirical equations can not be checked. The PERSPECT system automatically identifies which empirical equations are suitable for checking by comparing the knowledge of the domain model with that of the set of past designs. Table 22 details the identified empirical equations.

$$CB = 0.968 - (0.269 \times VS/(\sqrt{L/0.3048})$$

$$KG = 0.57 * (D * 0.913)$$

$$CP = (0.96 \times CB) + 0.04$$

$$CW = (0.878 \times CP) + 0.1733$$

where,
CB = block coefficient, VS = speed, L = length, CW = waterplane coefficient,
CP = prismatic coefficient, KG = distance from keel to centre of gravity,
D = depth,

Table 22: Pre-compiled empirical equations suitable for checking against up to date set of past designs.

Next, the pre-compiled empirical equations are checked by automatically calculating the new unreliabilities of the equations using the extended set of past designs. Table 23 lists the unreliability values of the previously generated empirical equations recorded in the domain model (i.e. *old* unreliabilities), the *recalculated* unreliabilities using the original set of past designs and *new* unreliability values using the extended set of past designs.

Empirical Equations (existing)	**Unreliabilities (%)**		
	Old	Recalculated	New
CB	2	1.96 (2)	2.04 (2)
KG	4	9.57 (10)	8.77 (9)
CP	0	0.49 (1)	0.49 (1)
CW	3	2.78 (3)	2.43 (2)

Table 23: Comparison of empirical equation unreliabilities reported in pre-compiled information, generated using original and extended group of past designs (round values shown in brackets)

The unreliabilities recorded in the domain model were rounded up to the nearest whole number by the model builder. However, by comparing the *old* unreliabilities with the recalculated ones, two of the four *old* unreliabilities have been estimated incorrectly.

The *old* unreliability associated with KG and CP should have been recorded as 10% and 1% respectively instead of 4% and 0%. These mistakes in the *old* unreliabilities can be accounted as manual errors, which would have been avoided if the functionality available in the PERSPECT system had been available to the original builder of the domain model.

It is important to update significantly changed unreliabilities since they are used to provide information on the precision of estimated attribute values [34]. Therefore, using PERSPECT the designer can specify which empirical equations should be updated with *new* unreliability values. In addition, if an equation renders too high an unreliability value, the pre-compiled empirical equation can itself be updated by employing regression on the utilised set of past designs. For example, the *new* unreliability of KG may be considered to be too high since it is the only one that has a value greater than say 5%. Therefore, the PERSPECT system is used to render a new equation by (a) translating the extended set of past designs into a form readable to the S-PLUS subsystem, (b) generating (using S-PLUS) a new relationship between KG and D, and (c) transferring this relationship into the DESIGNER subsystem as a new empirical equation. Then the unreliability of this new equation can be automatically calculated and added to the model. Table 24 details the result of re-regressing KG and D, shown as an updated empirical equation with associated unreliability. This table shows that a more reliable equation has been rendered. (Note that there is no need to delete the original pre-compiled equation since DESIGNER will always utilise the most reliable empirical equation given the available information.)

Empirical Equation (updated)	**Unreliability** (%)
$KG = 0.5817 + 0.5298 \times D$	3.697

Table 24: Updated empirical equations and unreliabilities using extended set of past designs

The design procedure can be initiated by the exploration of past designs to extract general experiential knowledge to be used subsequently during synthesis. All the capabilities of the S-PLUS system are available within the PERSPECT system. Consequently, the designer is provided with an extensive data analysis tool to interrogate the set of past designs by investigating and rendering useful empirical equations. PERSPECT also supports the automatic transfer of these rendered empirical equations into the DESIGNER subsystem. In this way, PERSPECT supports the construction of domain models 'from scratch'. Alternatively, if domain models already exist, PERSPECT can be used to check and update the applicability of these models, relative to an available set of past designs, by modifying pre-compiled equation unreliabilities or re-generating equations. Consequently, PERSPECT supports the utilisation of the most up to date and applicable experiential knowledge to be applied during design.

10.2.2 Initiation of the Design Model

Initiation of the design model requires the initial estimation of attribute values. Consequently, this stage involves the commencement, continuation and completion of the design process. This section explains how this can be supported by the PERSPECT system.

Commencement of New Design

Within numerical design, initiation may begin by considering the design goals. Table 25 details some typical goals [34] and the ones used in this worked example.

Goal type A(i) is the easiest to satisfy as it is not dependent upon any other attributes and hence the value can be input directly. The PERSPECT system can be used to find out automatically which defined goals can be used for initiation, i.e. which of the goals are of type A(i), in this case goals ENDGL and VSGL. However, assigning the values of these goals to the attributes in the design model provides little benefit. There is

Goal Name	Definition	Type
DWTGL	$DWT > 26900$ tonnes	A(ii)
ENDGL	$END = 19200$ nautical miles	A(i)
FBDGL	$FBD > REQFBD$	C
GMGL	$GM > 0.06 \times BEAM$	E
VSGL	$VS = 15$ knots	A(i)
VOLGL	$VOL > 1.27 \times DWT$	E

where,
DWT = dead-weight, END = endurance, FBD = freeboard,
REQFBD = required freeboard, GM = metacentric height,
VS = speed, VOL = volume

Table 25: Goals used in this worked example

still not enough information available to instantiate the numerical network, i.e. for an empirical equation to be fired and a new attribute to be estimated. Thus, by inputting the relevant values for ENDGL and VSGL, the state of the new design, i.e. the known attribute values of the new design, is as shown in Table 26.

Attribute Name	Value
ENDURANCE	19200
VS	15

Table 26: Known attributes of present design (current state of the design model)

At this stage, the designer can continue and complete the initiation of the design model.

Continuation and Completion of New Design

PERSPECT supports the continuation and completion of the design process by providing three options. The designer can:

- manually input values for unknown attribute values,
- employ customised perspectives (single or nested) to generate generalisations and identify the group of designs that are most similar to the new design,
- remove the need to input attribute values by abstracting the domain model and hence reduce domain model complexity.

OPTION 1 - Manual Input of Values

This option involves the user utilising the DESIGNER subsystem as a self contained system. Consequently, the PERSPECT system contributes no help to the designer who takes this option. The designer can choose either of the remaining two options for more informed assistance, and therefore supplement the knowledge associated with the domain model with additional experiential knowledge.

OPTION 2 - Customised Viewpoints

Supplementation of the domain model is required when attribute values of the new design are still unknown and compiled experiential knowledge is of no use to the new design; the designer is expected to input the values for the remaining uninstantiated attributes.

Customised viewpoints can be defined in the PERSPECT system and resulting viewpoints generated, displayed and utilised. The designer may choose this option for one of two purposes:

- To estimate values for unknown attributes.
- To identify suitable empirical equations to apply.

Estimating Attribute Values using Viewpoints (Single and Nested)

Customised viewpoints are useful when empirical equations are not available for the initiation of a design model, either because no empirical equations exist or because not enough attribute values are known to facilitate their usage. Designers can construct a customised viewpoint to estimate the values of the unknown attributes. Using their own or PERSPECT's knowledge of design attribute dependency, designers can define a perspective, consisting of the unknown attributes and related attributes, and use PERSPECT to generate and use a viewpoint of experiential knowledge to estimate a value for the unknown attributes.

Let's return to the situation where only VS and ENDURANCE are known, as in Table 26. From experiential knowledge associated with empirical equations, represented in DESIGNER, the user can identify equations that utilise these known attributes. One example of such an equation is that used to calculate CB using VS and LENGTH (see Table 22). The designer can define a perspective consisting of CB, VS and LENGTH to generate a viewpoint. This viewpoint can then be used to estimate values for CB and LENGTH using the new design (i.e. the values of VS and ENDURANCE).

Figure 44 illustrates a portion of a customised viewpoint of experiential knowledge generated by defining a perspective of CB,VS and L. This figure represents a hierarchy of groups and past designs. Groups are labelled with a name commencing with the letter "G" (referring to 'groups') and the past designs with a name commencing with the letters "EX-" (referring to 'examples'). G4, the top and most general group in the figure, represents the group that encompasses all the known past designs. This group is divided into four sub-groups G37, G27, G26 and G8 that are in turn further sub-divided and so on, as depicted by the dots in the figure. Each lower level in the figure represents a sub-division. For explanation purposes the generalised description of the illustrated groups are shown directly below the group name. The path of groups that are similar to the new design are indicated by detailing their associated descriptions and the most similar identified past design is indicated by the dashed circle.

After PERSPECT finds the past design most similar to the new design, the designer is now free to assign values for CB and L. The designer can either assign the values of the most similar past design or the values associated with any of the related more general groups. In other words, the designer can assign CB and L of the new design as 0.838 and 215.4, 0.835 and 214.5, 0.832 and 214.92 or 0.819 and 212.54 respectively. Table 27 details the possible values that could be assigned.

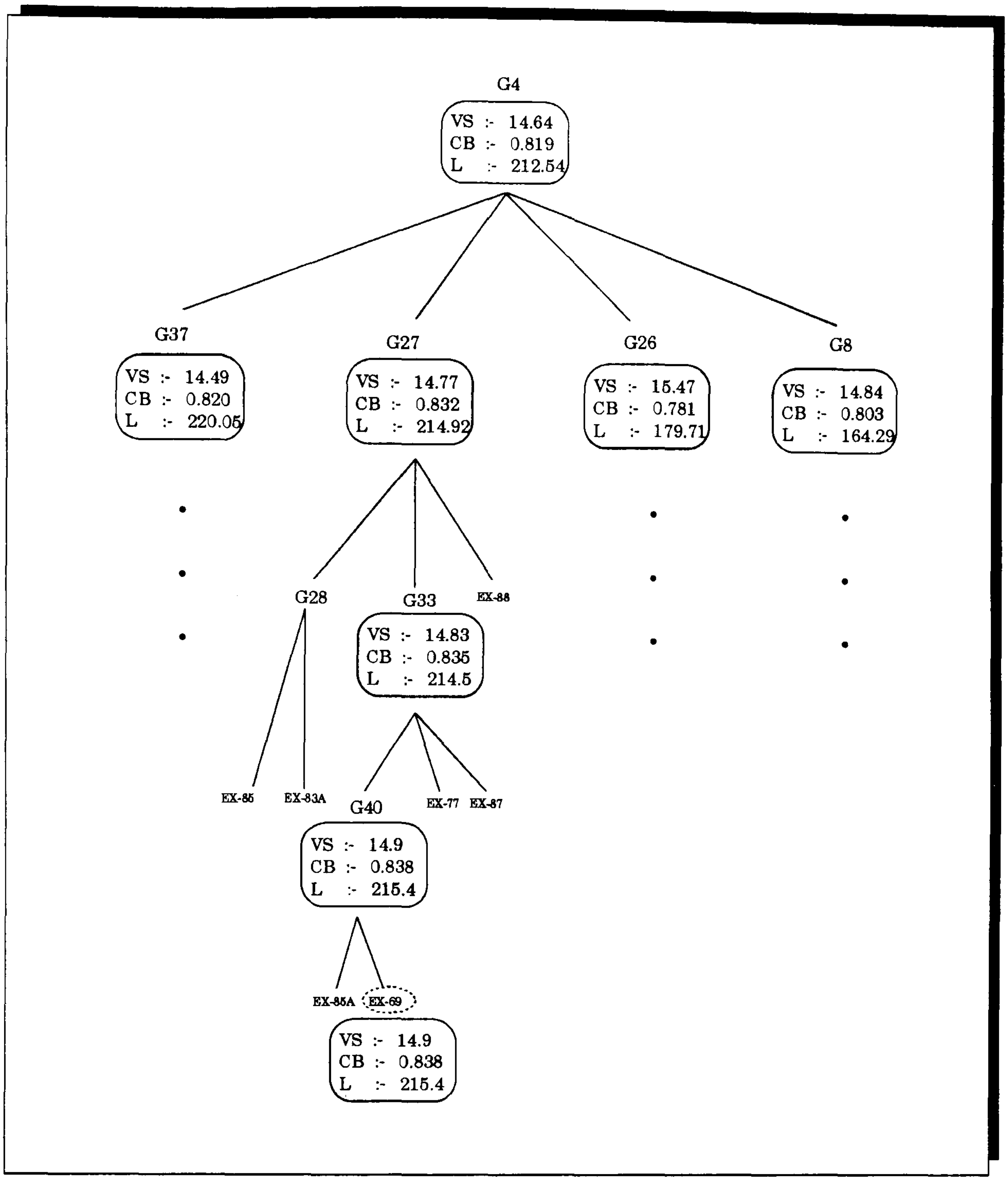


Figure 44: A portion of general experiential knowledge generated using a customised (single) viewpoint consisting of CB, VS and L

Attribute Name	Possible Values				
	EX-69	G40	G33	G27	G4
ENDURANCE	19200	19200	19200	19200	19200
VS	15	15	15	15	15
CB	0.838	0.838	0.835	0.832	0.819
LENGTH	215.4	215.4	214.5	214.92	212.54

Table 27: Possible attribute values of present design after utilising single viewpoint

In the previous viewpoint, ENDURANCE was not taken into consideration. There are two possible ways in which the consideration of ENDURANCE can be supported using PERSPECT: either a single viewpoint can be rendered by including ENDURANCE with CB, VS and L, or a nested viewpoint consisting of a single viewpoint of ENDURANCE (as shown in Figure 45) and a single viewpoint of CB, VS and L for a particular group can be generated, as shown in Figure 46.

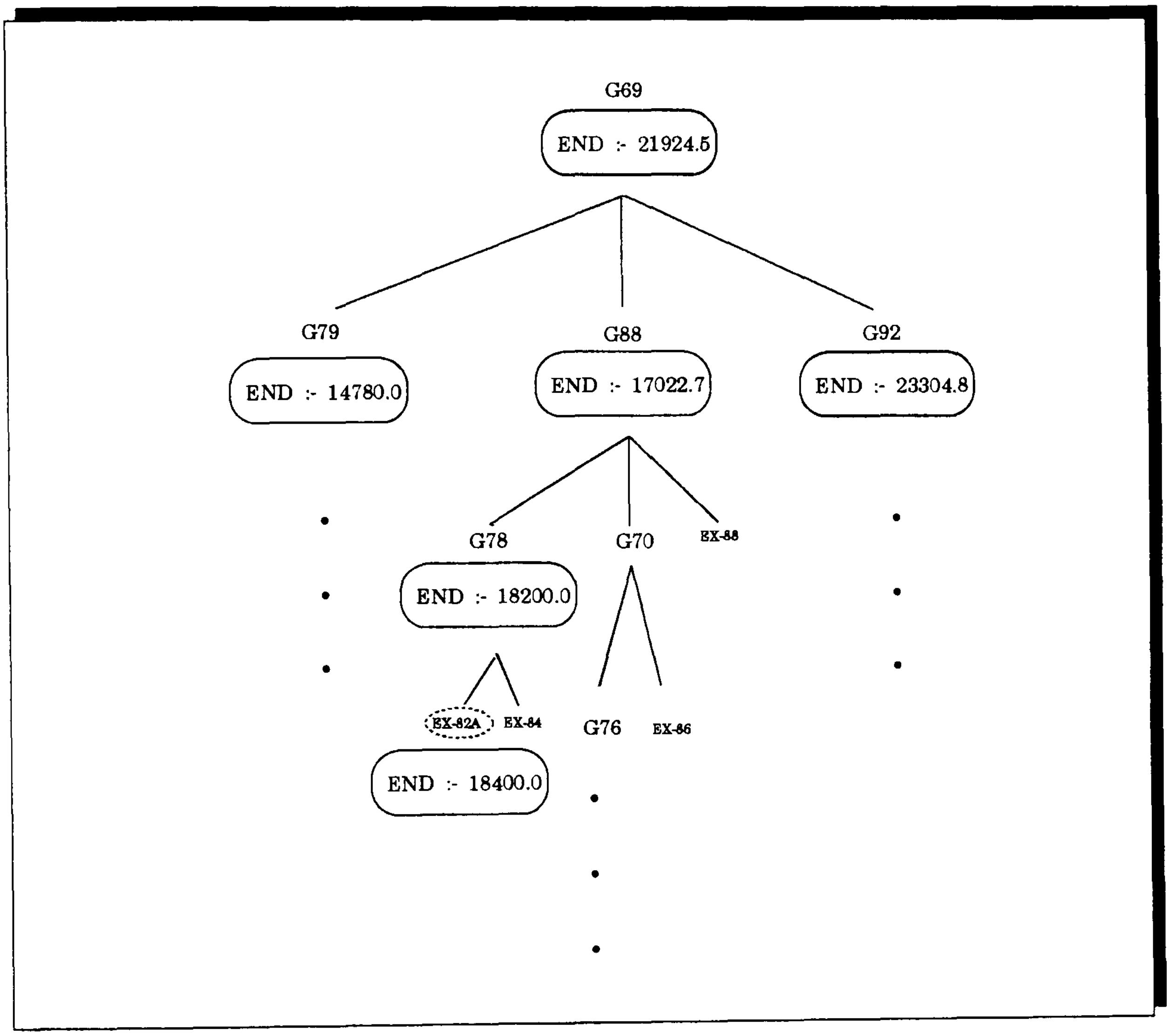


Figure 45: A portion of a customised single viewpoint of experiential knowledge consisting of ENDURANCE

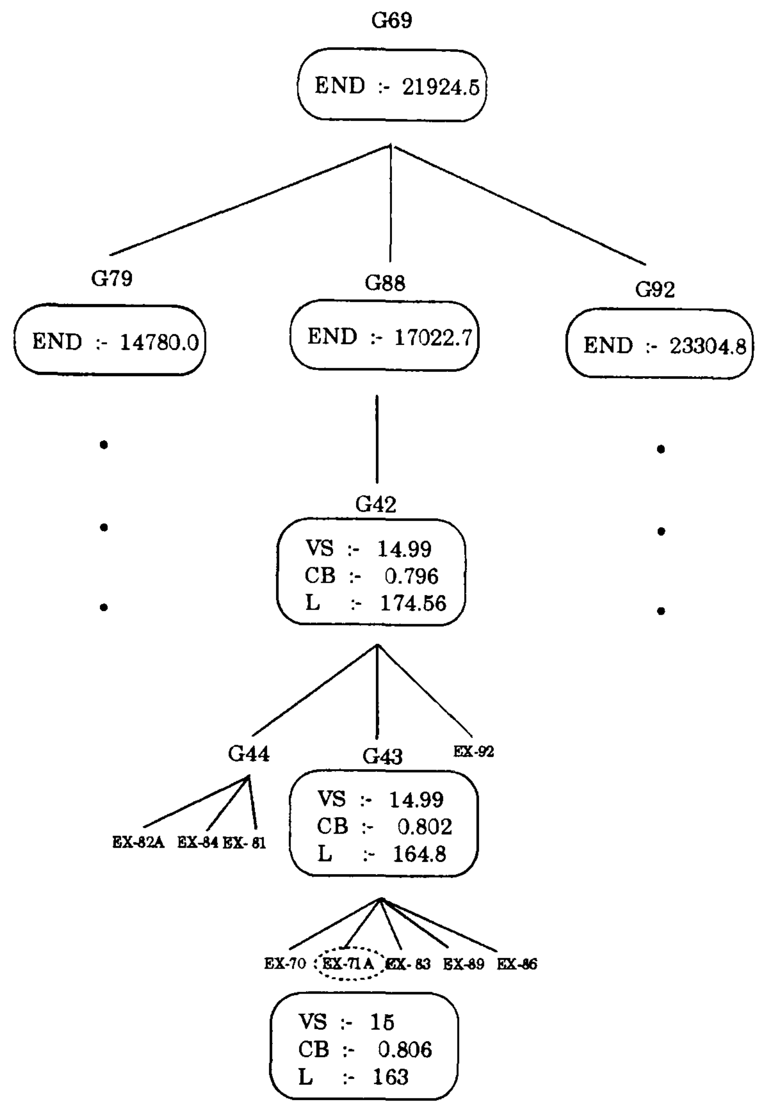


Figure 46: A portion of a customised nested viewpoint of experiential knowledge consisting of ENDURANCE, then CB, VS and L

To render and utilise a nested viewpoint requires the enhanced functionality of ECOBWEB provided by PERSPECT (see Chapter 8.3.2). Since the utilisation of a single viewpoint has already been discussed, as in the above case, the result of rendering and utilising a nested viewpoint will now be shown. First of all, PERSPECT is used to define a perspective of ENDURANCE and a single viewpoint is generated. Using the generated customised viewpoint, PERSPECT finds the most similar past design. Figure 45 shows EX-82A as the most similar past design to the new design with an ENDURANCE value of 19200.

Now, the designer is free to specify, at any level, the group to be nested so long as the group is a super group of EX-82A. Figure 46 shows a nested viewpoint generated from a design perspective of (a) the customised single viewpoint for ENDURANCE, (b) focusing on group G88 of this viewpoint and (c) focusing on CB, VS and L. Now using this nested viewpoint, PERSPECT finds the most similar past design to be EX-71A. The search for the most similar design is guided first by ENDURANCE and then by CB, VS and L. PERSPECT first found EX-82A as the most similar design by focusing on ENDURANCE only; however changing focus to CB,VS and L and considering the group of designs called G88, PERPSECT then found EX-71A. Hence the most similar of the past design in group G88 is EX-71A.

In this instance, Table 28 details the associated possible values for CB and L when using the nested viewpoint of Figure 46.

Attribute Name	**Possible Values**		
	EX-71-A	G43	G42
ENDURANCE	19200	19200	19200
VS	15	15	15
CB	0.806	0.802	0.796
LENGTH	163.0	164.8	174.56

Table 28: Possible attribute values of present design after utilising nested viewpoint

The details of Table 27 and Table 28 are different. Table 27 represent the results of focusing on the attributes CB, VS and L only, whereas Table 28 are the results of using the ENDURANCE attribute as a guide to search for the most similar design described by CB VS and L.

Identifying Suitable Empirical Equations Using Viewpoints

An alternative use of customised viewpoints is to assist in the retrieval of a stored empirical equation from an existing viewpoint, or the generation and use of suitable empirical equations using a viewpoint. The ability to retrieve a previously stored empirical equation utilises PERSPECT's more powerful representation and processing capabilities, i.e. PERSPECT's ability to generate, represent and utilise single and nested viewpoints described by generalisations and empirical equations. The procedure for generating a suitable empirical equation, rather than retrieval, will be discussed here as it is the more complex of the two and incorporates retrieval.

Recall the generated viewpoint illustrated in Figure 44 and the previous state of the new design detailed in Table 26. Using this state of the new design, PERSPECT found the most similar past design to be EX-69. Rather than use the description of this past design or associated super groups to estimate a value for the known attribute values, PERSPECT can be used to generate an empirical sub-equation. These equations quantify the relationship between attributes of a group using the associated subset of past designs that are its members. That is, for each group depicted in Figure 44 a particular empirical equation can be rendered. Rendered empirical sub-equations can then be (a) assigned to their associated group, or (b) added to the domain model to calculate their unreliabilities and subsequently used to estimate a value for the respective attributes.

Figure 47 illustrates these equations with their associated groups as a customised single viewpoint.

Table 29 details the rendered empirical sub-equations that describe the past designs labelled EX-85A, EX-69, EX-77 and EX-87 belonging to groups G40, G33, G27 and G4 along with their unreliabilities relative to the subset of past designs that were used to render them and the complete set of 31 past designs (i.e. the extended set).

Group Name	**Empirical Sub-equation** (generated)	**Unreliabilities** (subset)	(set)
G40	*does not exist*	-	-
G33	$CB = 0.095 + (0.049 \times VS)$	0.285	6.558
G27	$CB = 1.094 + (-0.011 \times VS) + (-0.0005 \times L)$	0.4915	5.445
G4	$CB = 0.924 + (-0.011 * VS) + (0.0003 \times L)$	2.034	2.034

Table 29: Generated empirical sub-equations associated with groups in the viewpoint shown in Figure 44 and unreliabilities of equations using subset and set of past designs

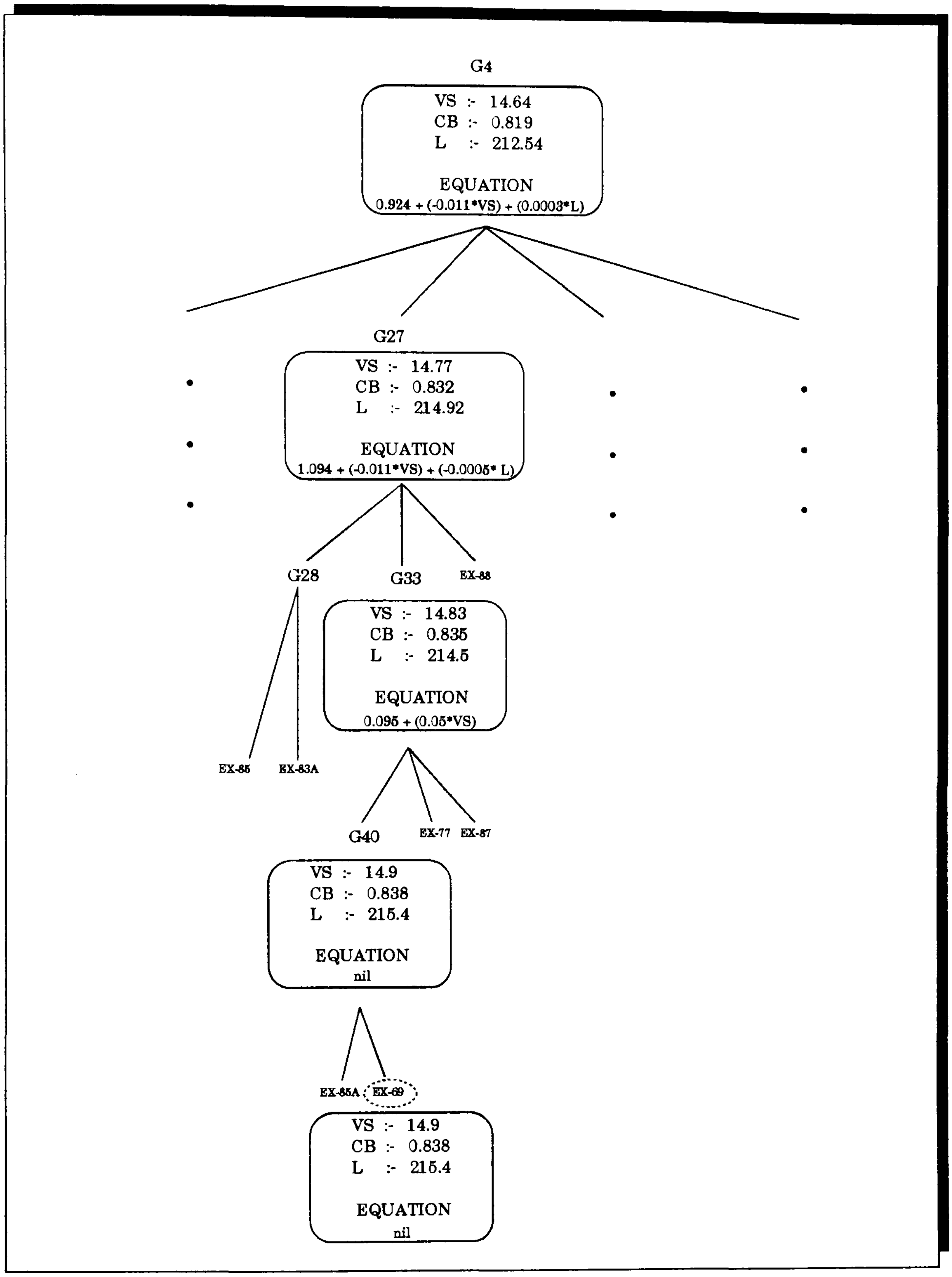


Figure 47: A portion of a customised single viewpoint of general experiential knowledge consisting of CB, VS and L showing generated associated empirical subequations

An important feature of the empirical sub-equation rendered for G33 is the removal of LENGTH, which S-PLUS has automatically done. For this group, CB can be estimated using only VS. That is, within this group of past designs, LENGTH has negligible influence on CB values. An empirical equation could not be rendered using the past designs of G40. Since G40 consists of only two past designs, all attribute values have to be known; the attribute value for CB is unknown for the past design called EX-85A. Therefore, an equation can not be rendered.

Calculating the unreliabilities of the rendered empirical sub-equations indicates that increasing the size of the set of past designs, used to render an equation, increases the unreliability of the associated equation (shown by the increasing values of the unreliability subset column). The past designs that are members of G33 are more similar than those of G27, which in turn are more similar to those of G4. Therefore, the scatter of past designs (as shown in Figure 24 of Chapter 7) from G33 will be more dispersed than G27 and G44. Thus, rendered empirical equation associated with these scatters will be more unreliable. Although equations associated to G27 and G4 might seem similar, they have been rendered using different subsets of past designs. G27 is made up of 7 past designs while G4 is made up of the complete set of past designs, i.e. 31 designs.

OPTION 3 - Abstraction of Domain Model

The domain model can be very complex, i.e. it can contain many empirical equations consisting of many input variables. This complexity often hinders designers by forcing them to input values and thereby utilise equations. The previous option showed how PERSPECT can assist designers by providing initial estimates for attributes' values. However, the option to abstract the domain model shows how the complexity of a domain model can be reduced, and thereby used to avoid the need for such initial estimates and to reduce the degree of complexity for the designer.

This option can involve three types of abstraction (see Chapter 8.5 Table 19). Because the information associated with the set of past designs is not necessarily the same as that used in the domain model, the equations suitable for abstraction must involve attributes that are utilised in the descriptions of these past designs and at least one of the input variables of the equations must already be instantiated. Therefore, PERSPECT identifies which of the pre-compiled empirical equations are suitable for abstraction. PERSPECT reports to the designer that only one equation at this stage is suitable for abstraction, i.e. the equation used to estimate CB using VS and L as input variables. To estimate a value for CB the value for L must be input by the designer. However, this can be avoided by removing the L input variable of the equation and generating a new empirical equation by regressing CB and VS. Table 30 details the precompiled empirical equation identified by PERSPECT as suitable for abstraction, based on the present state of the design model, and Table 31 gives the unreliability of the resulting generated abstraction.

Precompiled equation: $Cb = 0.968 - (0.269 \times Vs/(\sqrt{L/0.3048})$

Abstracted equation : $Cb = 1.087 - 0.019 \times Vs$

Table 30: Example of a pre-compiled and abstracted empirical equation

Empirical Equations (abstracted)	**Unreliabilities (%)** Extended
Cb	2.318

Table 31: Unreliability of abstracted empirical equation using extended set of past designs

Including this abstracted equation into the domain model facilitates the estimation of the *CB* attribute. Table 32 details the state (i.e. known attribute-values) of the design model after estimating CB.

Attribute Name	Value
ENDURANCE	19200
VS	15
CB	0.802

Table 32: Known attributes in design model

By providing a facility to abstract domain models, PERSPECT is capable of reducing complexity in the domain model. This allows the user to estimate attribute values of the design model without being forced to input all the required attribute values of a related equation. Abstraction of the domain model provides an alternative option to the designer. Instead of generating viewpoints of past designs to estimate attribute values using knowledge of similar past designs or empirical sub-equations, PERSPECT provides abstraction as a means of removing unwanted variables from empirical equations and generating less complex equations. Alternatively, if an existing domain model introduces too much complexity for a particular stage in design (e.g. at the early stages of design), PERSPECT can be used to abstract the domain model to generate less complex and potentially more useful equations.

10.2.3 Summary

This section has shown a number of ways in which PERSPECT can provide additional support over that provided by the DESIGNER system. PERSPECT can assist a designer to define a domain model or, if a model already exists, to check and evolve that model according to the available experiential knowledge. This task is known as domain model preparation. Also, PERSPECT can assist the task of design model initiation. Using PERSPECT for this task, designers have a number of options available to them. They can use the DESIGNER subsystem, which requires designers to input initial values for certain attributes as and when required leading to an instantiated design model, or they can use the added power of the PERSPECT system by utilising experiential knowledge of past designs to develop a design solution's attribute values. These values can be estimated in a number of ways: either by generating and utilising customised (single and nested) viewpoints to find generalisations of past designs or empirical sub-equations, or by abstracting the domain model to generate less complex equations.

Table 33 summarises PERSPECT's support for these two design tasks: model preparation and initiation. The table states the purpose of these tasks and provides the associated functionalities of the system that a designer can use for assistance.

Design Task	**Purpose**	**System Functionality**
Preparation	To build a design domain model	Regression Analysis Unreliability Calculations Model Update
	To check and modify a domain model	Unreliability Calculations Regression Analysis Model Update
Initiation	To instantiate a design model	Customise Viewpoints Domain Model Abstractions

Table 33: Summary of PERSPECT's support during specified design tasks

10.3 Critical Evaluation of PERSPECT

The realisation of the 'customised viewpoint' approach presented in this thesis is addressed by an implementation of a research prototype system called PERSPECT (see Chapter 9) and tested within an exemplary numerical design session (in the previous section) to illustrate the increased support provided to a designer. The result of this test has shown PERSPECT as a numerical CV tool that supports:

- **Preparation of a domain model** - by helping to render new or check and update existing models of design domains, to be used to develop a design solution.
- **Initiation of a design model** - by (a) customising viewpoints of experiential knowledge to estimate values for unknown attributes or identify suitable empirical sub-equations, or (b) reduce the complexity of existing domain models so that they are more appropriate to the level of detail required by particular designers.

The PERSPECT system has been evaluated to assess how well it supports numerical design and in particular the two tasks listed above. An aim of this evaluation has been to acquire an unbiased assessment of PERSPECT's capabilities. To achieve this, an assessment has been conducted with the assistance of two designers whose expertise lie in the extraction and use of experiential design knowledge originating from past designs and experiments but who are unfamiliar with the new approach presented in this thesis. Recorded evaluation sessions, each of 2-3 hours duration, have been carried out using one session with each individual designer to avoid the designers influencing one anothers' evaluations.

Each session consisted of three stages: *explanation, demonstration* and *discussion.* First of all, the evaluators received a basic explanation of the limitations of existing CAD systems' abilities to support the utilisation of experiential knowledge and then an introduction to the proposed 'customised viewpoint' approach; thereby familiarising the designers with the general philosophy behind the system. Next, the PERSPECT system was demonstrated, as detailed in Section 10.2, to highlight its numerical design utility. Finally, the session closed with a general discussion between the system demonstrator and the evaluator, where the demonstrator encouraged the evaluator to ask questions concerning issues that arose or were not addressed during the system demonstration.

Sessions were video recorded so that all the evaluators' comments and specific questions could be 'captured', and the resulting recordings provided material from which to develop a coherent interpretation of the evaluators' assessments. A protocol analysis of these sessions has been carried out by scrutinising each recording, identifying significant issues focused upon by the evaluators concerning the tasks PERSPECT supports and organising these issues into a coherent assessment of PERSPECT.

The following presents the most significant findings of this analysis. The results are organised according to the two main tasks that PERSPECT supports, categorised into advantages and disadvantages, along with some general comments concerning the system interface.

10.3.1 Preparation of a Domain Model

Advantages

- *Equation checking in accordance with that advocated in design literature*

  In well-defined design domains, that is domains that utilise generally accepted empirical equations, the user of PERSPECT will invariably refine an existing domain model. Therefore the importance of checking and updating existing domain models was recognised by the evaluators as being of direct benefit to a designer. For example, equations used in existing domain models may represent a domain that is outwith the scope of past designs available to a designer. In this case, it is very important to check and if necessary change existing empirical equations to reflect designers' own information. In fact, the importance of checking existing empirical equations or any experiential knowledge publicly available is characterised in a quote found in the design literature, i.e. Watson's 'final word of warning' at the end of his 1962 paper [40] and echoed by Watson & Gilfillan [42]:

  > "before any of the data or approximate formulae quoted in the paper are used, they should be checked against the user's own data." [40, 42]

- *Generation and subsequent use of equations from within a design environment*

  PERSPECT has shown the utility of providing users with the ability to explore a domain of interest and generate empirical equations, which can be directly fed into a CAD system (i.e. DESIGNER), and subsequently utilise such equations. This approach avoids the need for a programmer or knowledge engineer (i.e. 'a middle man') to elucidate and acquire knowledge for representation in the DESIGNER subsystem. This was considered a beneficial feature as it prevents the loss of information and gives the designer direct control over what is generated, represented and subsequently utilised, as indicated by the following evaluator's comment.

> "I like the idea of not losing information by making equation fits to data and then using that equation fit. If you [designers] have access to the raw data you're not losing information. You can look at it anyway you choose to look at it. That is very much a good point."

Therefore, the likelihood of errors is reduced and the representation and utilisation of more suitable equations encouraged.

- *Reduction of domain model complexity*

  In well-defined design domains, where domain models exist, PERSPECT can take advantage of existing models. In addition, it can overcome existing model complexity by generating abstract models, which may be more appropriate to the level of detail required by the design problem. The evaluators recognised PERSPECT's capability of reducing the complexity of a domain model (see section 10.2.2) as having practical use in a wide variety of domains, as expressed in the following quote.

  > "The idea of simplifying equations of something complicated right down to something simple is something that people [designers] do in a wide variety of domain applications."

  They also confirmed the need for different levels of model complexity, so that designers can utilise more complicated equations as the design solution develops, i.e.

  > "You [designers] start with simple equations which become more complicated. So its a good idea to allow them [designers] to have the different levels of investigation."

  Therefore, in addition to checking and updating existing models or rendering new models, PERSPECT helps to manipulate existing models to make their complexity reflect that required by a design problem.

Disadvantages

- *Limited confidence building*

  According to the evaluators, building up confidence in a domain model to be used in design is a very important aspect of design and one which competent designers do not overlook. In addition, confidence in generalised experiential knowledge, used to help synthesise a domain model, is required. PERSPECT's means of calculating, representing and checking the unreliabilities of empirical equations helps a designer formulate confidence in the resulting domain model. However, the evaluators expressed concern in the lack of its ability to assess the utility of generated customised viewpoints. Consequently, the evaluators suggested that PERSPECT should also include a capability of establishing confidence in the *application* of customised viewpoints.

- *Transfer of simple empirical equations only*

  PERSPECT supports the transfer of first order linear equations from S-PLUS to DESIGNER. However, these are somewhat simple equations compared to those that can be generated and used by the S-PLUS and DESIGNER subsystems respectively. The S-PLUS subsystem supports the exploration of more complex empirical equation structures, e.g. nonlinear equations. Also, these more complex equations can be used by DESIGNER. Unfortunately, these more complex equations cannot be directly fed into this subsystem from S-PLUS, via the transfer functions. Therefore, to improve support for the utilisation of more complex empirical equation structures, PERSPECT's transfer functions need to be extended.

- *Little additional assistance in rendering domain models*

  PERSPECT supports exploration, the results of which can be used to help synthesise a domain model. Rendering a domain model 'from scratch' depends greatly on designers' understanding of regression techniques and knowledge of the domain, i.e. expertise. For example, Scott [148] suggests that the number of terms used in an empirical equation should not be more than one-third of the total number of experiences. This type of expertise is not represented in PERSPECT. Therefore, the process of rendering a domain model is at present dependent on designers guiding the domain model preparation task to generate applicable experiential knowledge that can be used to synthesise a domain model. It was stressed that ideally such a system as PERSPECT should accommodate this type of expertise and thereby provide more active support in the exploration and synthesis of domain models.

10.3.2 Initiation of a Design Model

Advantages

- *Opportunistic utilisation of two forms of experiential knowledge*

  PERSPECT represents experiential knowledge in two forms: empirical equations and generalisations. Consequently, when prompted by the DESIGNER subsystem for additional information concerning design attribute values, these forms can be used to help the user estimate or temporarily avoid the need to estimate required design solution attribute values. These capabilities were considered by the evaluators as being important, especially if the user's own experiential knowledge of the design domain is limited. Therefore, the evaluators viewed PERSPECT as a tool which enhances designers' experiential knowledge.

- *Coupling of design and learning activities*

  The system has been built to incorporate the coupling of design and learning activities within a single computational environment. Consequently, learning can be carried out during, as a result of or in order to design. The evaluators considered the coupling of these activities to be a fruitful aspect of the system with which more practical design investigation should ensue.

- *Nesting as a means to help guide the search for applicable experiential knowledge*

  The ability to generate and utilise nested viewpoints was recognised as being an interesting and viable means of guiding the search for suitable empirical equations or generalisations. The process of nesting reflects a change in designers' focus and it was accepted that the sequence of nesting could be governed by the degree of priority given to particular foci, e.g. initial nesting achieved by a high priority focus with less important types of focus employed latterly.

- *Representation and utilisation of empirical sub-equations*

  The system ability to investigate and extract empirical equations that exist within subsets of past designs was recognised as being an effective meas of helping to promote the use of more applicable empirical equations and encourage the calculation of more accurate design attribute estimates. For example, a situation was proposed by one of the evaluators in which pre-compiled equations could describe the best fit of the whole set of past designs. However, if focus was placed on a specific subset of past designs, it would be feasible to identify sub-equations that are more representative of the subset; and thereby generate more accurate attribute value estimates than more global equations.

Disadvantages

- *Limited goal utilisation*

  Five types of design goals can be represented using the DESIGNER subsystem (see Table 25 of Chapter 7.1.1). Goals define requirements of the new design and are used by PERSPECT to help initiate a description of the new design model. However, PERSPECT uses only goal type A with an equality operator to suggest initial estimates for a new design's attribute values. To more fully support designers, PERSPECT has to be extended so that other types of goals can be used to suggest initial estimates.

- *No assistance for choosing between forms of experiential knowledge*

  The DESIGNER subsystem of PERSPECT manages the application of the most appropriate empirical equation to determine unknown design attribute values. Also, PERSPECT's functionality of generating and utilising single and nested viewpoints to identify similar past designs can be used to identify generalisations or equations that can be used to determine unknown design attribute values. However, PERSPECT fails to choose the best form (i.e. empirical equation or generalisation) of experiential knowledge for developing a design solution. The *application* of the most appropriate form of experiential knowledge is under the control of the user. The evaluators believed that PERSPECT should suggest which form of experiential knowledge is most appropriate.

- *Limited description of equation applicability*

  The evaluators emphasised the need to define the range of applicability of empirical equations. Within specific variable ranges an empirical equation may not be appropriate and thus could give incorrect estimates. Consequently, it is very important that the applicability of existing empirical equations is known and utilised. This important point is addressed by PERSPECT via the DESIGNER subsystem representation and an extension of ECOBWEB's representation. The DESIGNER subsystem is capable of representing and using equations according to their range of applicability. For example, two equations exist for estimating a value for CW (i.e. the waterplane coefficient); one equation is applicable when the CP (i.e. prismatic coefficient) value is less than 0.85 and the other when it is greater than 0.85. Consequently, depending on the value of CP, the DESIGNER subsystem selects the appropriate equation. An extension of ECOBWEB's representation to include the representation of empirical equations ensures that equations are associated with their appropriate groups of past designs. However, PERSPECT does not represent a description of these groups as attribute value ranges. Instead, their description consists of statistical information concerning the group, e.g. average values and standard deviations of attributes used to describe groups of past designs. Thus the representation of viewpoints of experiential knowledge needs to be extended and/or modified to take into account attribute value ranges.

- *Loss of equation applicability*

  Knowledge of the applicability of equations is stored in PERSPECT's `rationalisation` and the DESIGNER subsystem's `equation` data structures. Rationalisations, representing customised viewpoints, can be used to find a suitable equation to be transferred to the DESIGNER subsystem. However, the process of transferring an equation to DESIGNER results in the loss of knowledge concerning an equation's applicability. There are two possible solutions to avoid this loss of knowledge, either the DESIGNER subsystem should directly utilise PERSPECT's rationalisations to select appropriate empirical equations or the knowledge detailing the applicability of equations should be transferred with equations.

10.3.3 System Interface

From a user's point of view, the evaluators pointed out that the operation of PERSPECT is cumbersome. For example:

- there are two types of user-commands that depend on two different programming languages; Lisp-based and S-based commands. Lisp-based commands are used to interact with the DESIGNER subsystem and the PERSPECT system, and S-based commands are used to interact with the S-PLUS subsystem;
- the presentation of known empirical equations in the system is presented to the designer as a Lisp-based structure (e.g. $(\times\ 0.56\ DEPTH\ 0.913)$), rather than arithmetic-based structure (e.g. $0.56 \times DEPTH \times 0.913$). This Lisp-based structure is not easily understood by designers, especially if the equation is complex;
- differences between rendered and pre-compiled equations are not effectively presented to the designer; and
- generated generalisations and empirical equations stored in viewpoints are not readily available to a designer and accessing such information is awkward.

To ease the burden of interaction, additional work is required to enhance the user interface and provide a unified interface from which all PERSPECT subsystems can be accessed and information can be presented in an accessible and easily understood form to designers.

In general terms, the importance of developing CAD tools that remove a difficult design task with simple operations using computers was stressed by the evaluators. This might be an obvious statement however it has implications in the evaluation of research prototype systems. These systems, such as the PERSPECT system, are evaluated by designers to help ascertain unbiased system merits and demerits, and identify avenues of system improvement. However, the quality of user interface can obscure the utility of such systems and thereby influence designers' evaluations. Therefore, one of the evaluators stressed that developers of early prototype systems should attempt to address interface issues to promote the main system features and ease the evaluation task for designers.

10.4 Summary

Existing CAD systems fail to support the effective utilisation of experiential knowledge according to the needs of designers as they typically represent pre-defined viewpoints that reflect knowledge engineers' perspectives of designers' knowledge requirements. A 'customised viewpoint' approach is presented in this thesis as a means whereby generalised experiential knowledge can be generated directly from specific experiences, according to designers' knowledge needs, and subsequently utilised to help develop a design solution. This approach is complimentary to the CAD philosophy of "design assistance", which is demonstrated by the increased preliminary numerical design support provided to a designer by the PERSPECT system. However, before PERSPECT can be called a comprehensive numerical CV tool, a number of improvements are required to support the needs of practising designers.

11 Discussion

The CV approach represents a new way of thinking about computational design support. Its implementation is proposed to improve the effective utilisation of experiential design knowledge and promote "design assistance". The utility of the CV approach, within the realm of preliminary numerical design, has been identified, tested and evaluated via the implementation of the PERSPECT system. This system utilises information of specific past designs and thereby allows designers to explore and extract general experiential knowledge that can be used subsequently to develop a design solution.

This chapter discusses some of the more general implications of the approach rather than its implementation. First of all, Section 11.1 argues for the potential of the CV approach to support design creativity and then Section 11.2 details a number of ways in which its general utility in design can be improved. Finally, Section 11.3 discusses avenues of further research, which can substantially advance the applicability of this approach in design.

11.1 The CV Approach and Design Creativity

It was mentioned earlier (in Chapter 3.2) that designers' creativity can be hindered by past experiences and that the use of related knowledge can confine future designs to the characteristics of previous designs, resulting in a 'stagnated' design domain. This criticism is based on the idea that design experiences can hinder designers' creativity by inhibiting designers' ability to work with new ideas. In other words, past experiences can restrict the generation and development of design concepts to only those which reflect characteristics of past concepts. In addition, the continual reuse of design experiences (bad or good) can be criticised for producing a design domain that is void of fresh and innovative ideas. In such a 'stagnant' design domain, the boundaries of the domain are restricted by the capabilities, characteristics, performances, etc., of past design experiences. Consequently, past design experience can be criticised for being a source of creativity barriers, rather than a source of potentially useful experiential knowledge, for developing a design solution.

This thesis refutes this criticism that the use of experiential knowledge inhibits creativity, therefore preventing the advancement of a design domain. It is argued here that the CV approach can be used to encourage design creativity in two ways:

- *Encourage cross fertilisation*

  A feature of the CV approach is that of supporting the coupling of design and learning activities. For example, the exploration of specific past design information to generate general experiential knowledge that can be used subsequently to synthesise a design model. Although the CV approach depends on specific information (e.g. past design information), a process of abstraction and generalisation helps to make this information more generalised and therefore more widely applicable. In addition, this approach helps to identify the most applicable generalised knowledge which can be used to develop a design solution. Consequently, the CV approach not only supports the generation of general knowledge, it supports the application of appropriate general knowledge.

  A limiting factor of the general knowledge generated by the CV approach lies in the scope of specific information available for abstraction and generalisation. For example, if only specific bulker ship type designs are available, it would be inappropriate to use associated general knowledge to design tanker type designs. However, if information of both design types was available, it may be possible for designers to identify general 'cross type' knowledge by comparing tankers with bulkers and use this knowledge to contribute to 'cross type' fertilisation. Similarly, if information is included about other domains, designers can explore and possibly identify useful knowledge that can contribute to 'cross domain' fertilisation. Therefore, using the CV approach, designers can generate viewpoints that reflect their chosen perspective of how they wish to view trends across types or domains.

In other words, dependent on the scope of available specific information, the CV approach can encourage cross fertilisation between types or domains and thereby can promote the utilisation of knowledge from analogous designs.

- *Motivate innovation*

  In Chapter 4.1, trends were described as valuable implicit knowledge. It was highlighted that designers can use trends to guide the development of new design solutions so that they can exhibit current design practices/styles or avoid those that are unfavourable.

  The CV approach can be used to help designers explore, extract and utilise trends. However, in addition, these identified trends can be used to motivate creativity. Generated time based trends are proposed to notify designers of 'stagnating' design domains. For example, Figure 48 illustrates the time based trend of attained aircraft speed for propeller and turbine driven designs, and shows the gradual stagnation of propeller driven designs (indicated by the levelling out of the associated trend curve).

  The levelling out of a trend can indicate to a designer that a degree of innovation is required to advance the domain and resume the general domain trend. Therefore, trends that seem to be indicating stagnation can motivate designers to push further the existing barriers of a domain. Consequently, the innovative introduction of turbines to aircraft design enabled the domain to break the speed of sound. This feature of the CV approach has not been tested in this thesis. However, it is proposed as one way in which the CV approach, through the use of trends that 'drive', rather than promote or discourage, existing design practices, may support design creativity.

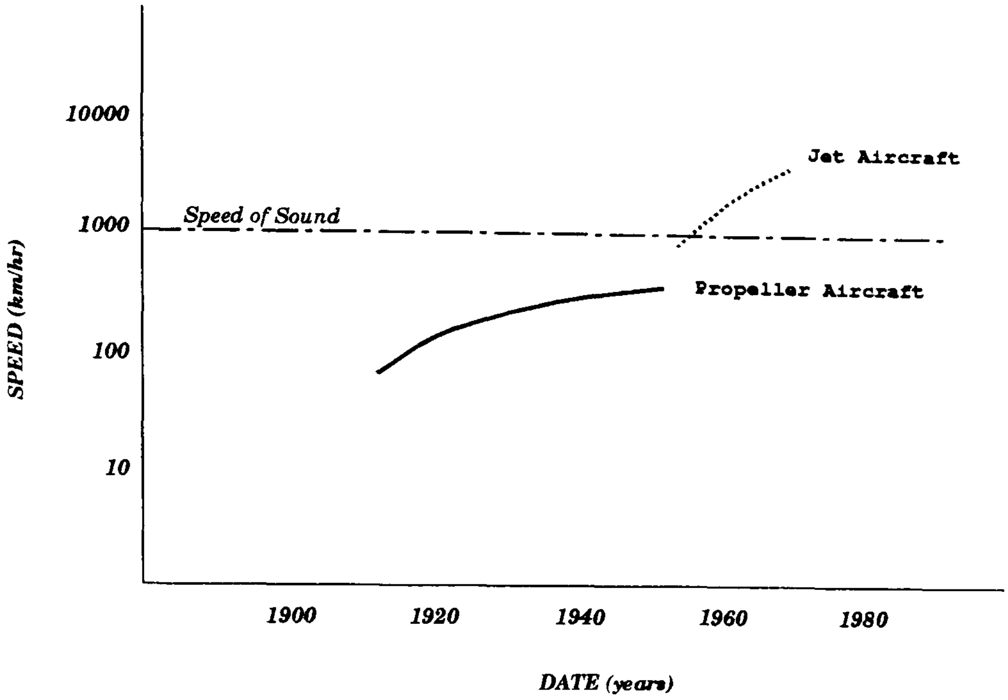


Figure 48: Time based trend of aircraft speed, adapted from [149]

The CV approach is proposed to promote design creativity by using design experiences. In this approach, design experiences are used to (a) encourage the extraction and use of general experiential knowledge across design types or design domains, or (b) help identify design trends that indicate a stagnant design domain and thus motivate the need for creativity and the advancement of a design domain. Thus the criticism of experiential knowledge inhibiting creativity and preventing a design domain's advancement is no longer valid when used with the CV approach.

11.2 Improvements to the CV Approach

This thesis has presented, implemented and demonstrated the CV approach as one that supports the extraction and use of experiential knowledge, based on designers' knowledge needs, using a source of experiential knowledge. However, this approach can be improved upon.

The list below details a number of such improvements that, if supported, will improve the approach's capability of assisting designers.

- introduce heuristics
- modify viewpoints
- acknowledge quality and applicability of experiential knowledge source

- *Introduce heuristics*

  So far the CV approach has been presented as playing a *passive* role in supporting designers; that is, where the designer controls the generation and utilisation of experiential knowledge. Although this supports useful design tasks, as demonstrated in Chapter 10, a more *active* role of this approach can better support these tasks. Chapter 10.3.1 suggested the representation and utilisation of regression heuristics, to help guide the exploration of empirical equations, as one way to enhance the utility of the numerical implementation of the CV approach. In effect, these heuristics would be acting as supplementary knowledge that increases the richness of represented experiential knowledge. The resulting approach would be seen more as an *advisor* to a designer that suggests, for example, interesting regions of a domain to explore or appropriate decompositions of a domain. In this way, the approach would *actively* support the generation of customised viewpoints of experiential knowledge.

- *Modify viewpoints*

  An ability that has not been supported in this thesis is that of modifying customised viewpoints. This could be achieved by (a) automatically altering the knowledge associated with customised viewpoints to reflect an updated source of experiential knowledge or (b) supporting the manipulation of viewpoints as directed by designers. The first sub-task ensures that previously generated viewpoints are incrementally updated with information associated with new designs. Consequently, the development of a design domain would be accommodated by the experiential knowledge associated with generated viewpoints. Alternatively, the second sub-task allows designers to manipulate customised viewpoints and thereby potentially improve the applicability of generated viewpoints. For example, designers could interactively alter the structure of an automatically generated customised viewpoint and the resultant experiential knowledge held within the restructured viewpoint would be automatically re-generated.

The CV approach should accommodate both approaches of achieving this task, thereby automatically maintaining viewpoints of experiential knowledge that are up to date and encouraging their customisation that better reflect designers' knowledge needs.

- *Acknowledge quality and applicability of experiential knowledge source*

  The CV approach is dependent on a source of experiential knowledge (e.g. a set of past designs) from which customised viewpoints of experiential knowledge are generated and used subsequently to develop new design solutions. This source (i.e. specific experiences) may consist of good or bad design experiences that can be equally as useful to designers. For example, for identifying favourable or undesirable design trends. Alternatively, a particular source may contain good and bad experiences according to some criteria, which used together may be help to generate experiential knowledge that is more accurate than using good or bad experiences separately.

  At present the quality and applicability of the source to be used with the CV approach is determined by its users. This is undesirable. The CV approach should be able to assess the quality of an experiential knowledge source according to criteria set by a designer's perspective. Such knowledge should be represented explicitly and thereby assist designers to determine the applicability of represented experiences. In other words, the CV approach should ensure the applicability of a source of experiential knowledge, rather than leave designers with this responsibility.

11.3 Future Work

This thesis has focused upon the utility of the CV approach in preliminary numerical design. Chapter 10.3 presented the advantages and disadvantages of the PERSPECT system and within the disadvantages discussed a number of required improvements to the system, namely:

- confidence building for generated customised viewpoints,
- transfer of more complex empirical equations from S-PLUS to DESIGNER,
- utilisation of heuristics to help render domain models,
- utilisation of more complex goal types to suggest initial estimates for a new design's attribute values,
- selection of the most appropriate form of experiential knowledge,
- representation of richer experiential knowledge in viewpoints, e.g. attribute value ranges, and
- utilisation of empirical equation applicability.

However, the ideas behind this approach equally apply to all of the design phases, concerns the utilisation of experiential knowledge from viewpoints other than numerical ones and originating from sources other than past designs. Therefore, in addition to these required improvements, the following list summarises main areas of future work, which are proposed to substantially advance the applicability of the CV approach and hence its utility in design.

- multiple roles of experiential knowledge
- multiple types of focus
- multiple sources of experiential knowledge
- single and nested viewpoints
- acquisition of experiential knowledge
- integrating CV approach with PDV approach

- **Multiple roles of experiential knowledge**

  The aim of this thesis has been to improve the effective utilisation of experiential knowledge in design. In Chapter 3 a number of roles that experiential knowledge plays in design have been presented; roles that relate to different design process types, phases and activities. The 'customised viewpoint' approach, presented in this thesis, specifically addresses the role of experiential knowledge during the design activities carried out in preliminary design (i.e. to build (synthesise) a design domain model to be subsequently utilised to synthesise a new design by exploring and evaluating experiential knowledge). However, the applicability of this new approach to support the remaining identified experiential knowledge roles needs to be investigated in order to make a step towards more comprehensive "design assistance" that would provide designers with assistance through all design types, phases and activities.

- **Multiple types of focus**

  This thesis has focused upon a well-defined problem area with which to demonstrate and evaluate the utility of the CV approach. The resulting implemented system, PERSPECT, supports numerical design, and is applicable in any design domain that utilises empirical equations to develop a design solution and where past designs can be described by a list of numerical attribute-value pairs. However, the CV approach is proposed to apply to many types of focus (see Chapter 4.3.1) not just numerical, e.g. geometrical, spatial, functional, structural, etc. So what can the CV approach contribute to these other types of focus?

  In a design study carried out by Muller [150], a student design team identified four geometrical types of four wheeled go-carts (i.e. the sprinter, crosser, trimmer and easy-rider) from a number of generated conceptual design alternatives. Although the students used conceptual design alternatives, they could have easily been using existing designs. The same basic process of identifying and extracting abstractions (i.e. general geometric design types) from more specific information (i.e. specific conceptual design alternatives) would have been carried out. As a design team, the students came to a consensus of four design types. Similarly, in Chapter 4.3.1, Figure 9 presents an example of abstracting and generalising cars from a geometrical aspect. The figure illustrates four geometries of existing cars from which two geometrical concepts can be generated. How was this achieved? Supporting the extraction and use of geometrical general knowledge represents challenging problems. What characteristics of geometrical designs do designers focus on? How do they recognise similarities between designs and how are those similarities characterised? The success of the CV approach in the geometrical focus is dependent on researchers answering these questions, along with issues concerning the representation of abstract geometry, the definition of designers' geometrical perspectives, and the abstraction of geometrical information. These problems are currently being investigated [151].

Within the realm of spatial design, there exist few published examples of abstractions and generalisations. However, it is believed that the CV approach has relevance within this type of focus. Consequently, the applicability of this approach will be discussed using a small published example and a simulated one. Figure 49 illustrates an example adapted from Guena & Zreik [152]. In this example, two existing designs have been abstracted and generalised into a more general spatial layout consisting of three space types. These space types map on to spaces with different functional properties, i.e. offices for work, corridor for horizontal circulation of people, and stairwell for vertical circulation of people. In addition, attributes of the overall layout (i.e. length and breadth) have been generalised. This type of extraction of general knowledge is slightly more apparent and easier to explain than the previous example in geometrical design. If this type of spatial abstraction was computationally feasible, the CV approach would be used to specify the designers' perspective as the functionality of spaces and the overall length and breadth attributes of existing designs.

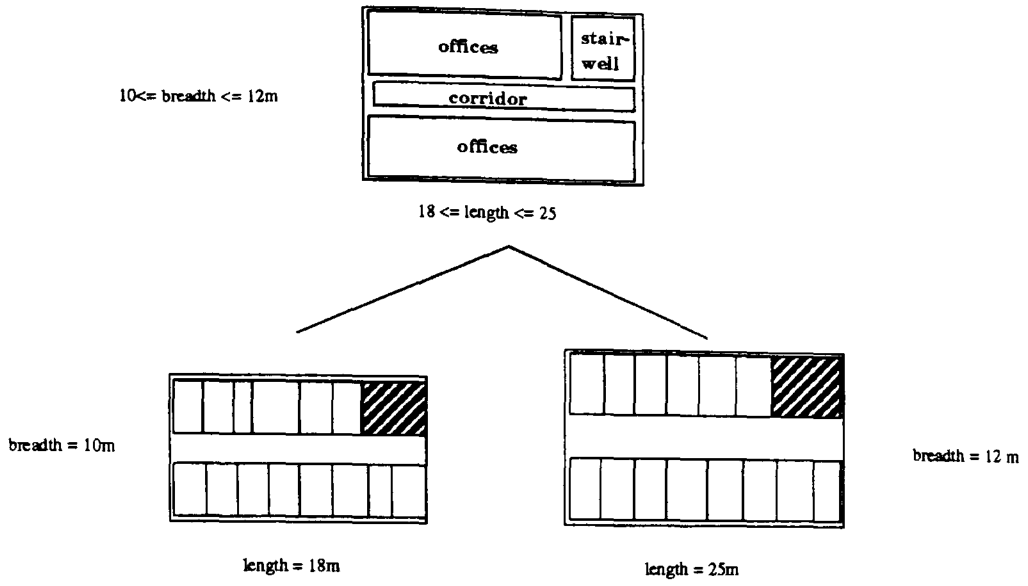
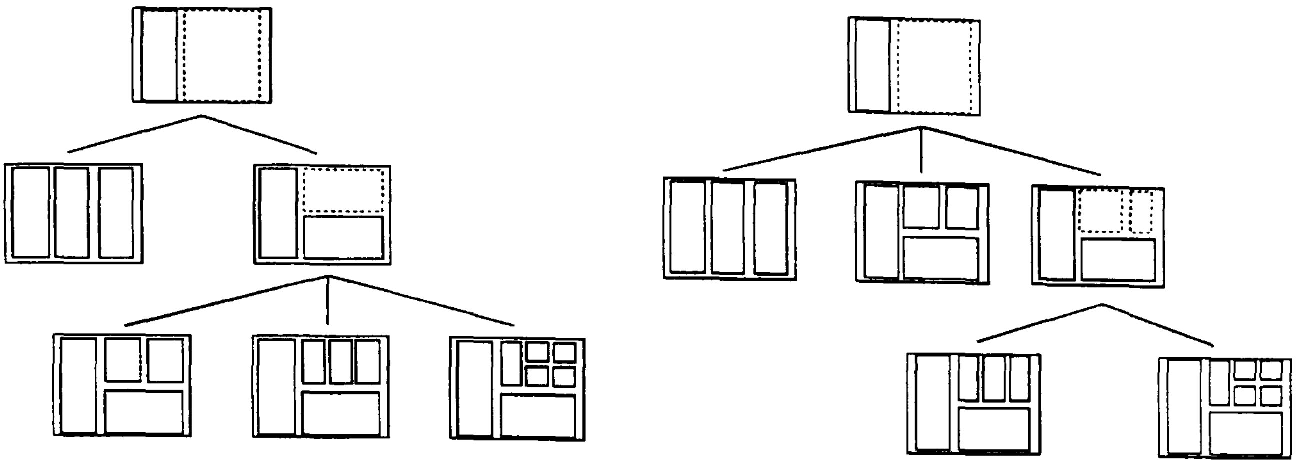


Figure 49: Published example of spatial abstraction and generalisation, adapted from [152]

Another example of spatial abstraction is illustrated in Figure 50. This is a hypothetical example, however it demonstrates that such abstraction is dependent on designers' perspectives. For example, rather than focusing on the functionality of spaces, as illustrated in the previous spatial example, designs can be abstracted and generalised by focusing on regions. Figure 50 illustrates two possible abstract hierarchies generated from four hypothetical layout designs. These hierarchies were generated by focusing on the right hand-side spaces of each layout. Both hierarchies are equally valid viewpoints of the presented layouts. However, they reflect different perspectives governing how best to abstract the specific layouts. Assuming that this type of spatial knowledge can be represented, it is proposed that the CV approach should be able to handle such differences in perspectives when abstracting and therefore present to users a hierarchy that reflects their particular perspectives.



(a) (b)

Figure 50: Two possible abstract hierarchies of four hypothetical spatial layouts

- **Multiple sources of experiential knowledge**

  One of the general requirements for the effective experiential knowledge utilisation is supporting multiple sources of experiential knowledge. This thesis has focused on experiential knowledge originating from the past designs. However, other sources of experiential knowledge exist, e.g. design episodes that help to formulate knowledge of 'how' to design. Effectively utilising knowledge of design episodes would involve generating generalised knowledge about 'how' to design from either fragments of or complete records of previous design processes. It is important to emphasise that the use of such design process knowledge is not to promote a CV tool capable of automating the design process, but rather one that *assists* the designer by suggesting possible courses of action during a new design process.

- **Single and nested viewpoints**

  This thesis has identified and presented the existence of two types of viewpoints (i.e. single and nested) that designer's can generate and use to direct their attention. A process and representation formalism have been developed in which these viewpoints can be computationally generated and utilised. However, now that such a facility is available, further work is required to investigate the practical implications of single and nested viewpoints in design. This requires collaboration with practising designers so that further uses of these types of viewpoints can be explored.

- **Acquisition of experiential knowledge**

  A CV tool is required to support the coupling of design and learning activities. The PERSPECT system couples exploratory, synthesis and evaluation design activities with acquisition, generation and modification learning activities. However, it does not support the acquisition of knowledge originating externally to the system. This is presently an active research area in Int.CAD, which has been given the term *knowledge capture* [153] and is considered here a relevant area of research with which to expand the 'customised viewpoint' approach.

- **Integrating the CV Approach with the PDV Approach**

  Design, like other activities, depends on effective communication. One way in which effective communication can be achieved is through a consensus of domain conceptualisations, i.e. agreement between designers on how a design domain can be decomposed into abstract chunks of knowledge. Such a consensus encourages the reuse of, for example, design concepts.

  Current approaches of design support capitalise on design knowledge consensus; however, in doing so they promote the *regurgitation* of knowledge and do not accommodate the *application* of designers' customised conceptualisations of a design domain. In Chapter 6, the CV approach was presented as one that overcomes the traditional, more restrictive approach to computationally utilising experiential knowledge in design where a knowledge engineer compiles pre-defined knowledge, i.e. the "pre-defined viewpoint" (PDV) approach. The idea of the CV approach is that the knowledge presented to designers should reflect their own perspectives and not preconceived ones.

  Design consensus is a very important aspect and one that should continue to be supported and developed. However, this should not happen at the expense of exploring and representing more customised knowledge. It is therefore proposed that the PDV approach should be integrated with the more flexible CV approach in-

introduced in this thesis. Such a development introduces new issues to the problem of supporting the use of experiential design knowledge. For example, what is the link between pre-defined and customised viewpoints, how does a designer choose between a pre-defined or a customised viewpoint, how does a system manage the potential explosion of viewpoints, should access to customised viewpoints be restricted to designers who initiate them? These are only a few issues that need to be addressed; others will evolve as the realisation of such integration is attempted.

12 Conclusions

As a result of the work carried out and reported in this thesis a number of conclusions can be made.

* Experiential knowledge plays a crucial role in design. This role has been investigated by focusing upon the utility of experiential knowledge in **types**, **phases** and **activities** of design.

- **Design types**: No matter the type of design (i.e. original, adaptive or variant) the designer is engaged in, experiential knowledge is a key resource that designers utilise (a) for the ideation and successful completion of a design solution, (b) to determine the degree of originality/innovativeness of a new design and (c) to identify shifts in the originality of designs.
- **Design phases**: Throughout the design phases (i.e. marketing, problem specification, conceptual, preliminary and detailed design) experiential knowledge manifests itself as a resource from which a design need can be identified and a design idea generated, developed and refined.
- **Design activities**: Design activities such as exploration, synthesis and evaluation all utilise experiential knowledge to help identify, extract, utilise, assess and make decisions about knowledge.

* Past designs and experiments provide sources of explicit information from which implicit knowledge (i.e. trends) can be identified and extracted explicitly into potentially useful abstract and general forms. Therefore, experiential knowledge provides a valuable resource that encompasses:

- a great expanse of knowledge originating from a number of sources, e.g. past design cases,
- implicit knowledge inherent in explicit information, i.e. design trends, and
- multiple forms of knowledge (e.g. heuristics, empirical equations) that explicate implicit knowledge into terms that are understandable and usable.

Designers can manage the utilisation of this variety (i.e. specific to abstract) and complexity (i.e. implicit from explicit) of experiential knowledge by structuring knowledge into viewpoints, which is inherently conditional on designers' needs. This thesis presents evidence for the existence of two particular types of viewpoints in design not previously acknowledged, i.e. **single** and **nested** viewpoints.

Consequently, the general requirements for supporting the effective utilisation of experiential knowledge in design have been identified as:

- the utilisation of multiple sources and forms of experiential knowledge,
- the coupling of design and learning activities, i.e. exploration, synthesis and evaluation, and acquisition, generation and modification, and
- the utilisation of multiple viewpoints of experiential knowledge that reflect the needs of designers.

* Current CAD approaches are primarily directed at the effective and efficient representation of the richness of experiential knowledge. Most systems embrace predefined viewpoints reflecting only specific perspectives of experiential knowledge, i.e. none support the means of abstracting and generalising knowledge to suit particular needs of designers. In addition, few systems support the comprehensive coupling of design and learning activities with which generalised experiential knowledge can be acquired, developed and directly fed back into design for synthesis or evaluation purposes. Systems capable of learning generally support limited automatic generalisation of explicit information to represent implicit knowledge explicitly.

* This thesis presents a new approach proposed to compliment the "design assistant" philosophy of CAD by addressing the effective utilisation of experiential knowledge. The **'customised viewpoint'** approach is a means whereby:

- experiential knowledge can be automatically updated and evolved to reflect the most recent design trends,
- viewpoints can be automatically generated from a source of experiential knowledge, according to designers' own particular perspectives, and
- the utilisation of applicable knowledge contained in an appropriate viewpoint can be computationally supported.

* No single existing system satisfies these requirements. However this thesis proposes that the integration of applicable technology (i.e. regression and concept formation) with an existing numerical design system can make a step towards the realisation of a 'customised viewpoint' tool and therefore improve the "design assistance" available to a designer.

Three existing systems have been identified as playing complimentary roles for the realisation of the 'customised viewpoint' approach: DESIGNER is a CAD system that supports preliminary numerical design; S-PLUS is a commercial package that provides comprehensive data analysis techniques such as regression; and ECOBWEB is a concept formation learning system that classifies examples into generated conceptual hierarchies.

However, an analysis of these systems has identified a number of limitations. First, DESIGNER is unable to utilise experiential knowledge other than empirical equations, and its design utility is dependent on a pre-defined model of a design domain. Second, ECOBWEB is unable to support the generation, representation and utilisation of customised single and nested viewpoints that contains experiential knowledge of empirical equations. Third, the means by which S-PLUS supports the generation of empirical equations according to designers' knowledge requirements is cumbersome.

Thus, integrating and extending the functionality of these three systems has provided the foundation for the realisation of a numerical 'customised viewpoint' tool.

* The PERSPECT system is the realisation of a numerical 'customised viewpoint' tool that provides the following features.

- **Integration of multiple forms of experiential knowledge**
  This is achieved by the transfer of information from one system to another: from ECOBWEB to S-PLUS, from S-PLUS to DESIGNER and from DESIGNER to ECOBWEB.
- **Definition of perspectives**
  A means of defining designers' unpredictable or particular knowledge needs, to govern the generation of appropriate viewpoints of experiential knowledge from a single source, i.e. past designs.
- **Generation, representation and utilisation of customised single and nested viewpoints of experiential knowledge**
  To capitalise on existing functionality, ECOBWEB was extended to support the representation and processing capabilities required to incorporate the generation, representation and utilisation of both customised single and nested viewpoints. The resulting data structure is called a **rationalisation**.

  - **Rendering and abstraction of domain models**
    Numerical domain models can be rendered using S-PLUS and ECOBWEB. The abstraction of a numerical domain model is supported in three ways: (a) the removal of empirical equations, with or without the regeneration of equations, (b) the abstraction of empirical equations and (c) the removal and abstraction of empirical equations.
  - **The coupling of design and learning activities**
    PERSPECT can be used to *explore* a set of past designs to render generalised knowledge by a process of *generation.* This knowledge can then be used to build (i.e. *synthesise*) a domain model which in turn can be used to help *synthesise* and *evaluate* a design solution. Alternatively, an existing domain model can be *evaluated* by checking its knowledge content against an available set of past designs and, if desired, *modified* to reflect knowledge more characteristic of these designs. While *synthesising* a design solution, additional experiential knowledge can be *explored* and used to supplement the knowledge in the domain model. Finally, PERSPECT supports the automatic *acquisition* of a new design to a set of past designs.

* The PERSPECT system is used to illustrate (a) the utility of a 'customised viewpoint' tool in numerical design and (b) the increased "design assistance" available to a designer. Consequently, PERSPECT supports two design tasks:
  - **Domain Model Preparation**
    PERSPECT supports the preparation of a numerical model of a design domain by (a) rendering a new domain model or (b) checking and updating an existing model in the light of new design experiences.
  - **Design Model Initiation**
    PERSPECT supports the initial assignment and calculation of suitable estimates of attribute values that describe a new design by (a) supporting the opportunistic utilisation of forms of experiential knowledge via the extraction of implicit knowledge in the form of explicit empirical equations and generalisations (b) reducing design complexity via the abstraction of an existing domain model.

* The system has been demonstrated to designers, whose expertise lie in the extraction and use of experiential design knowledge originating from past designs and experiments. An analysis of their comments revealed that, although PERSPECT supports valuable additional assistance, additional work is required to improve PERSPECT's ability to support numerical design and user interface. The following list details the main advantages and disadvantages of PERSPECT's capabilities.

  - **Domain Model Preparation**
    - *Advantages*
      - Equation checking in accordance with that advocated in design literature
      - Generation and subsequent use of equations from within a design environment
      - Reduction of domain model complexity
    - *Disadvantages*
      - Limited confidence building
      - Translation of simple empirical equations only
      - Little additional assistance in rendering domain models
  - **Design Model Initiation**
    - *Advantages*
      - Opportunistic utilisation of two forms of experiential knowledge
      - Coupling of design and learning activities
      - Nesting as a means to help guide the search for applicable experiential knowledge
      - Representation and utilisation of empirical sub-equations
    - *Disadvantages*
      - Limited goal utilisation
      - No assistance for choosing between forms of experiential knowledge
      - Limited description of equation applicability
      - Loss of equation applicability

* This thesis proposes and discusses (although does not demonstrate) how the CV approach can be used to support design creativity in two ways: by encouraging cross fertilisation and motivating innovation. It also discusses three aspects which, if supported, can improve the CV approach. These are :

  - utilising heuristics,
  - modifying viewpoints, and
  - acknowledging quality and applicability of experiential knowledge sources.

Finally, a number of areas of future work are presented to improve the effective utilisation of experiential design knowledge and thereby make a further step towards the realisation of a comprehensive "customised viewpoint" tool that exhibits the ideal of "design assistance". These areas of future work are:

- supporting multiple roles of experiential knowledge,
- supporting the generation of viewpoints from multiple types of focus and multiple sources of experiential knowledge,
- investigating the practical implications of single and nested viewpoints,
- supporting the acquisition of experiential knowledge, and
- supporting the integration of the CV approach with the PDV approach.

References

[1] V. Papanek. *Design for the Real World.* Thames and Hudson, London, UK, second edition, 1984.

[2] G. Sturt. *The Wheelwright's Shop.* Cambridge University Press, 1923.

[3] A. Freidmann, J.F. Pile, and F. Wilson. *Interior Design: An introduction to architectural interiors.* Elsevier, New York, USA, second edition, 1970.

[4] K.J. Rawson and E.C. Tupper. *Basic Ship Theory*, volume 1. Longman Inc., third edition, 1983.

[5] B. Colvin. *Land and Landscape: Evolution, design and control.* J. Murray, London, second edition, 1973.

[6] A.D. Bradshaw. Ecological principles in landscape. In A.D. Bradshaw, D.A. Goode, and E.H.P. Thorp, editors, *Ecology and Design in Landscape*, pages 15–36, Oxford, 1983. Twenty-fourth Symposium of the British Ecological Society, Manchester, Blackwell Scientific Publications.

[7] W.H. Mayall. *Principles in Design.* Design Council, London, UK, first edition, 1979.

[8] M.F. Ashby and Jones D.R. *Engineering Materials: An introduction to their properties and applications.* Pergamon Press, Oxford, UK, first edition, 1980.

[9] R.L. Motard. Design: Chemical engineering. In W.R. Spillers, editor, *Basic Questions of Design Theory*, pages 143–146. North-Holland, 1974.

[10] R. Easterby. Machine graphics. In C.H. Flurscheim, editor, *Industrial Design in Engineering: A marriage of techniques*, chapter 7, pages 159–173. The Design Council and Springer-Verlag, 1983.

[11] P. Murdoch. Psychological factors in man/machine interface design. In *Industrial Design in Engineering : a marriage of techniques*, chapter 11, pages 213–226. The Design Council and Springer-Verlag, 1983.

[12] S. Pugh. *Total Design: Integrated methods for successful product engineering.* Addison-Wesley Publishing Company, Wokingham, England, UK, 1990.

[13] K. Lynch. *Site Planning.* MIT, USA, 1962.

[14] M.L. Maher and F. Zhao. Using experience to plan the synthesis of new designs. In J.S. Gero, editor, *Expert Systems in Computer-Aided Design*, pages 349–369, Amsterdam, The Netherlands, 1987. Proceedings of the IFIP WG 5.2 Working Conference on Expert Systems in Computer-Aided Design, Sydney (Jenolan Caves), Australia, February, 1987, Elsevier Science Pubs. B.V. (North-Holland).

[15] R. Scott and N. Macleod. *Process Design Case Studies*, pages 1–3. Institution of Chemical Engineers, Warickshire, 1992.

[16] K.J. MacCallum, A. Duffy, and S. Green. An intelligent concept design assistant. In H. Yoshikawa and E.A. Warman, editors, *Design Theory for CAD*, pages 301–317, Amsterdam, The Netherlands, October 1987. Proceedings of the IFIP WG 5.2 Working Conference on Design Theory for CAD, Tokyo, Japan, October 1985, Elsevier Science Pubs. B.V. (North Holland).

[17] R. Coyne. *Logic Models of Design*, chapter 2, pages 9–28. Pitman Publishing, London, 1988.

[18] S. Finger and J.R. Dixon. A review of research in mechanical engineering design. part 1: descriptive, prescriptive and computer based models of design processes. *Research in Engineering Design*, 1:51–71, 1989.

[19] T. Robertson. The process of innovation and the diffusion of innovation. *Journal of Marketing*, 31:14–19, January 1967.

[20] P. Culverhouse. Four design routes in electronics engineering product development. *Journal of Design and Manufacturing*, June 1993.

[21] G. Pahl and W. Beitz. *Engineering Design: A systematic approach.* The Design Council, London, 1984.

[22] R.M. Henderson and K.B. Clark. Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. *Administrative Science Quarterly*, 35:9–30, 1990.

[23] B. Chandrasekaran. Towards a taxonomy of problem-solving types. *AI Magazine*, 4(1):9–17, 1983.

[24] D.C. Brown and B. Chandrasekaran. Expert systems for a class of mechanical design activity. In J.S Gero, editor, *Knowledge Engineering in Computer-Aided Design*, pages 259–282, Amsterdam, The Netherlands, 1985. Proceedings of the IFIP WG5.2 Working Conference on Knowledge Engineering in Computer-Aided Design, Budapest, Hungary, 1984, Elsevier Science Pubs. B.V. (North-Holland).

[25] J.S. Gero. Design prototypes: A knowledge representation schema for design. *AI Magazine*, 11(4):26–36, Winter 1990.

[26] M.J. French. *Conceptual Design for Engineers.* The Design Council, London, 1985.

[27] D.G. Ullman, T.G. Dietterich, and L.A. Stauffer. A model of the mechanical design process based on empirical data. In *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing*, volume 2, pages 33–52. Academic Press Ltd., 1988.

[28] M. Asimov. *Introduction to Design.* Prentice Hall, Englewood Cliffs, NJ, USA, 1962.

[29] H. Kamil, A.K. Vaish, and L. Berke. An expert system for integrated design of aerospace structures. In J.S Gero, editor, *Artificial Intelligence in Design*, pages 41–53. Proceedings of the Fourth International Conference on the Applications of Artificial Intelligence in Engineering, Springer-Verlag, July 1989.

[30] R. Bakerjian (ed.). *Tool and Manufacturing Engineers Handbook: Design for manufacturability*, volume 6, chapter 1, pages 1/1–1/17. McGraw-Hill Book Co. in cooperation with the Society of Manufacturing Engineers (SME), Michigan, USA, fourth edition, 1992.

[31] R. Bakerjian (ed.). *Tool and Manufacturing Engineers Handbook: Design for manufacturability*, volume 6, chapter 8, pages 8/1–8/16. McGraw-Hill Book Co. in cooperation with the Society of Manufacturing Engineers (SME), Michigan, USA, fourth edition, 1992.

[32] R. Bartos. Ads that irritate may erode trust in adverstised brand. *Business Review*, 59:138–140, 1981.

[33] Packaging design seen as cost-effective marketing strategy. Marketing News, February 1981.

[34] A.H.B. Duffy. *Computer Modelling of Early Stage Numerical Ship Design Knowledge and Expertise.* PhD thesis, Department of Ship and Marine Technology, University of Strathclyde, Glasgow, UK, February 1986.

[35] R.D. Hisrich and M.P. Peters. *Marketing Decisions for New and Mature Products.* Merrill Publishing Company, Columbus, Ohio, USA.

[36] E. Krick. *An Introduction to Engineering.* Wiley, 1969.

[37] Y. Reich. *Building and Improving Design Systems: A machine learning approach.* PhD thesis, Department of Civil Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, April 1991.

[38] T. Smithers, A. Conkie, J. Doheny, B. Logan, and K. Millington. Design as intelligent behaviour: An AI in design research programme. DAI Research Paper 426, Department of Artificial Intelligence, Univeristy of Edinburgh, Edinburgh, UK, May 1989.

[39] R. Oxman. Prior knowledge in design: A dynamic knowledge-based model of design and creativity. *Design Studies*, 11(1):17–28, January 1990.

[40] D.G.M. Watson. Estimating preliminary dimensions in ship design. Transactions IESS, volume 105, pp. 110-161, January 1962.

[41] A.W. Gilfillan. The economic design of bulk cargo carriers. In *Transactions of the Royal Institution of Naval Architects*, volume 3, pages 113–140. 1969.

[42] D.G.M. Watson and A.W. Gilfillan. Some ship design methods. *Naval Architect*, pages 279–324, July 1977.

[43] K.J. MacCallum. Creative ship deign. In International Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design *ICCAS*, 1982.

[44] K.J. MacCallum and A. Duffy. Approximate calculations in preliminary design. pages 219–225. The International Conference on Computer Applications in the Automiation of Shipyard Operation and Ship Design, Elsevier Science Publishers B.V. (North-Holland), 1985.

[45] P. Fazio, C. Bedard, and K. Gowri. Building envelope analysis and design system. In J.S. Gero, editor, *Artificial Intelligence in Design*, pages 133–142, Berlin, Heidelberg, 1989. Computational Mechanics Publications, Springer-Verlag.

[46] C.R. Chaplin. Creativity in engineering. The Fellowship of Engineering, London, UK, November 1989.

[47] N. Cross. *Engineering Design Methods.* John Wiley and Sons, 1989.

[48] M.L. Maher. Process models for design synthesis. *AI Magazine*, 11(4):49–58, Winter 1990.

[49] R. Moneo. On topology. *Oppositions*, 13:23–45, 1978.

[50] S. Pugh. Introduction to total design. Design course notes: Design Division, Department of Design, Manufacture and Engineering Management, University of Strathclyde, Glasgow, UK, 1990.

[51] K.J. MacCallum. Does Intelligent CAD exist? *Artificial Intelligence in Engineering*, 5(2):55–64, 1990.

[52] D.M. Himmelblau. Design in chemical engineering. In W.R. Spillers, editor, *Basic Questions of Design Theory*, pages 121–126. North-Holland Publications, 1974.

[53] P. Galle and L.B. Kovacs. Introspective observations of sketch design. *Design Studies*, 13(3):229–272, June 1992.

[54] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobsen, I. Fiksdahl-King, and S. Angel. *A Pattern Language.* Oxford University Press, New York, 1977.

[55] P.R. Taylor. Cable-stayed bridges and their potential in Canada. *Engineering Journal*, 52(11), 1969.

[56] U. Pighini, G. di Francesco, D.Z. Yuan, A.V. Schettino, and A. Rivalta. The determination of optimal dimensions for a city car using methodical design with prior technical analysis. *Design Studies*, 4(4):233–243, October 1983.

[57] S. McLaughlin and J.S. Gero. Requirements of a reasoning system that supports creative design activity. *Knowledge-Based Systems*, 2(1):62–71, 1989.

[58] A. Persidis and A. Duffy. Learning in engineering design. In H. Yoshikawa, F. Arbab, and T. Tomiyama, editors, *Intelligent CAD III*, pages 251–272, Amsterdam, The Netherlands, 1991. Selected and Reviewed Papers and Reports from the IFIP TC/WG5.2 Third Intelligent Workshop on Computer Aided Design, Osaka, Japan, September 1989, Elsevier Science Publishers B.V. (North-Holland).

[59] Professional engineering journal. The magazine of the Institution of Mechanical Engineers, Mechanical Engineering Publications Ltd.

[60] W.T. Chan and B.C. Paulson. Exploratory design using constraints. *AI EDAM*, 1(1):59–71, 1987.

[61] K.J. MacCallum. Ship design: course notes. Department of Ship and Marine Technology, University of Strathclyde, Glasgow, UK, 1989.

[62] F. de Siervo and F. de Leva. Modern trends in selecting and design Francis turbines. *Water Power and Dam Construction*, 28(8):28–35, August 1976.

[63] F. de Siervo and F. de Leva. Modern trends in selecting and designing Kaplan turbines. *Water Power and Dam Construction*, 29(12):51- 56, December 1977.

[64] A. Lugaresi and A. Massa. Designing Francis turbines: trends in the last decade. *Water Power and Dam Construction*, 39(11):23–28, November 1987.

[65] A. Lugaresi and A. Massa. Designing Kaplan turbines: trends in the last decade. *Water Power and Dam Construction*, 40(5):12–17, May 1988.

[66] P. Galle, J.P. Hansen, S.E. Jansen, M. Rubow, and E.C. Wright. Boligmonstre. monstersamling 11 ("housing patterns. pattern collection 11"). Pattern theory, Royal Danish Academy of Fine Arts, School of Architecture, Department 3D, Copenhagen, Denmark, 1975.

[67] R.A. Willem. Design and science. *Design Studies*, 11(1):43–47, January 1990.

[68] A. Persidis. *Modelling of Abstraction for Computer Aided Design.* PhD thesis, CAD Centre, Department of Design, Manufacture and Engineering Management, University of Strathclyde, Glasgow, UK, 1989.

[69] R. Schank. *Dynamic Memory: A theory of remindings and learning in computers and people.* Cambridge University Press, Cambridge, UK, 1982.

[70] P.J Veerkamp. Multiple worlds in an intelligent CAD system. In H. Yoshikawa and D. Gossard, editors, *Intelligent CAD I*, pages 77–88, Amsterdam, 1989. Proceedings of the IFIP TC 5/WG 5.2 Workshop on Intelligent CAD, Boston, MA, USA, October 1987, North-Holland.

[71] N.P Suh. *The Principles of Design.* Oxford University Press, Oxford,UK, 1990.

[72] S.P. Hoover, J.R. Rinderle, and S. Finger. Models and abstractions in design. *Design Studies*, 12(4):237–245, October 1991.

[73] H. Schmekel. *A system for conceptual design based on general and systematic principles of design.* PhD thesis, Computer Systems for Design and Manufacturing, Department of Manufacturing Systems, The Royal Institute of Technology, Stockholm, Sweden, 1992.

[74] A.H.B. Duffy and S.M. Kerr. Customised perspective of past designs from automated group rationalisation. *International Journal of Artificial Intelligence in Engineering*, 8(3), December 1993.

[75] F. Schweiger and J. Gregori. Developments in the design of Kaplan turbines. *Water Power and Dam Construction*, 39(11):16–20, November 1987.

[76] F. de Siervo and A. Lugaresi. Modern trends in selecting and design reversible Francis pump-turbines. *Water Power and Dam Construction*, 32(5):33–42, May 1980.

[77] Schweiger F and J. Gregori. Developments in pump-tubines. *Water Power and Dam Construction*, 34(10):22–25, October 1982.

[78] F. Schweiger and J. Gregori. Developments in the design of Bulb turbines. *Water Power and Dam Construction*, 40(9):12–15, September 1988.

[79] K. Svanberg. Optimization of geometry in truss design. In *Computer Methods in Applied Mechanics and Engineering*, pages 63–80. 1981.

[80] R. MacCafferty. Design optimisation - a modular interactive approach. *CAD'84*, pages 316–324, April 1984.

[81] C.W.S To. Vapsimm: A computer program for vibration analysis of piping systems incorporating a moving medium. In R.A. Adey, editor, *Engineering Software IV*, Berlin, June 1985. Proceedings of the Fourth International Conference, Kensington Exhibition Centre, London, UK, A Computation Mechanics Publication, Springer-Verlag.

[82] C.A. Brebbia, D.J. Danson, J.M.W. Baynham, and Adey R.A. Beasy: Boundary element analysis system. In R.A. Adey, editor, *Engineering Software IV*, pages 15141–15157, Berlin, June 1985. Proceedings of the Fourth International Conference, Kensington Exhibition Centre, London, UK, A Computation Mechanics Publication, Springer-Verlag.

[83] H. Sardar and A. Saleh. Computer aided design of shell structures. In R.A. Adey, editor, *Engineering Software IV*, pages 1019–1031, Berlin, June 1985. Proceedings of the Fourth International Conference, Kensington Exhibition Centre, London, UK, A Computational Mechanics Publication, Springer-Verlag.

[84] R. Fletcher. *Optimisation.* Academic Press, London, UK, 1969.

[85] O.C. Zienkiewicz. *The Finite Element Method.* McGraw-Hill, London, UK, third edition, 1977.

[86] K.J. MacCallum, B. Yu, A. Frederiksen, and D. McGregor. A system for supporting design configuration. In J.SS. Gero, editor, *Artificial Intelligence in Design '92*, pages 3–19, Dordrecht, The Netherlands, 1992. Second International Conference on Artificial Intelligence in Design, June 1992, Pittsburgh, USA, Kluwer Academic Publishers.

[87] E.A Warman. Computer aided design: Intersection of ideas. In J.C. Latombe, editor, *Artificial Intelligence and Patter Recognition in Computer Aided Design*, pages 1–18, Amsterdam, 1978. North Holland.

[88] P. Pu and N.I. Badler. Design knowledge capture and causal simulation. In H. Yoshikawa and D. Gossard, editors, *Intelligent CAD I*, pages 201–212, Amsterdam, The Netherlands, 1988. Proceedings of the IFIP TC 5/WG 5.2 Workshop on Intelligent CAD, Boston, MA, USA, October 1987, North-Holland.

[89] T. Tomiyama and P.J.W. ten Hagen. Organisation of design knowledge in an intelligent CAD environment. In J.S. Gero, editor, Amsterdam, The Netherlands, *Expert Systems in Computer-Aided Design.* Proceedings of the IFIP WG5.2 Working Conference on Expert Systems in Computer Aided Design, Elsevier North-Holland, 1987.

[90] K.J. MacCallum and S. Green. Thesys: Implementation of a knowledge-based design system with multiple viewpoints. In V. Akman, P.J.W. ten Hagen, and P.J. Veerkamp, editors, *Intelligent CAD Systems II: implementation Issues*, Eurographic Seminars, pages 228–244, Berlin, 1989. Springer-Verlag.

[91] D. Xue, T. Ishida, T. Tomiyama, and H. Yoshikawa. Implementation and application of a design knowledge representation. In *Preprints of the Third IFIP WG5.2 Wokshop on Intelligent CAD, Osaka, Japan*, 1989.

[92] R. Ganeshan, Finger S., and J. Garrett. Representing and reasoning with design intent. In J.S. Gero, editor, *Artificial Intelligence in Design.* Butterworth-Hienemann, 1991.

[93] G.T. Nguyen and D. Rieu. Representing design objects. In J.S. Gero, editor, *Artificial Intelligence in Design.* Butterworth-Heinemann Ltd, 1991.

[94] R. Elmasri and S.B. Navathe. *Fundamentals of Database Systems.* The Benjamin/Cummings Publishing Company Inc, California, USA, 1989.

[95] A. Yazaki and H. Sumoyoshi. Information retreival system and its application for the basic-design of the ship. Preprints of the International Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design (ICCAS), Session 2, Paper 1, August 1973.

[96] H. Matsuka. Interactive ship design systems. Preprints of the International Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design, Session 6, Paper 8., August 1973.

[97] A. Barr and E.A. Feigenbaum. *The Handbook of Artificial Intelligence*, volume 1. Addison-Wesley Publishing Company Inc., Reading,UK, 1986.

[98] S. Addanki and E. Davis. A representation for complex physical domains. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 443–446, August 1985.

[99] S.S. Murthy and S. Addanki. Prompt: An innovative design tool. In J.S.Gero, editor, *Expert Systems in Computer Aided Design*, pages 323–347, Amsterdam, The Netherlands, 1987. Proceedings of the IFIP WG5.2 Working Conference on Expert Systems in Computer-Aided Design, Sydney (Jenolan Caves), Australia, February, 1987, Elsevier Science Publishing B.V. (North Holland).

[100] R. Oxman and J.S. Gero. Designing by prototype refinement in architecture. In J.S. Gero, editor, *Artificial Intelligence in Engineering: Design*, pages 395–412. Computational Mechanics Publications, Southampton,UK, 1988.

[101] J.S. Gero, M.L. Maher, and W. Zhang. Chunking structural design knowledge as prototypes. In J.S. Gero, editor, *Artificial Intelligence in Engineering; Design*, pages 3–21, Amsterdam, The Netherlands, Elsevier Science Publishing B.V. (North-Holland), 1988.

[102] M.G. Dyer, M. Flowers, and J. Hodges. Edison: An engineering design invention system operating naively. In D. Sriram and R. Adey, editors, *Application of Artificial Intelligence in Engineering Problems*, volume 1, pages 327–341, Berlin, April 1986. First International Conference on the Application of Artificial Intelligence in Engineering Problems, Southampton University, UK, Springer-Verlag.

[103] D. Navinchandra. Cased-based reasoning in cyclops: A design problem solver. pages 286–301, San Mateo, CA, 1988. Proceedings of a workshop on case-based reasoning workshop, Florida, USA, May 1988, Morgan Kaufmann.

[104] M.L. Maher and D.M. Zhang. Cadsyn: using case and decomposition knowledge for design synthesis. In J.S. Gero, editor, *Artificial Intelligence in Design '91*, pages 137–150, Oxford, UK, 1991. Papers from the First International Conference on Artificial Intelligence in Design, Edinburgh, UK, June 1991, Butterworth-Heinemann Ltd.

[105] J. Wang and H.C. Howard. A design-dependent apporach to integrated structural design. In J.S. Gero, editor, *Artificial intelligence in Design '91*, pages 151–170, Oxford, UK, 1991. Papers from the First International Conference on Artificial Intelligence in Design, Edinburgh, UK, June 1991, Butterworth-Heinemann Ltd.

[106] K. Hua, I. Smith, B. Faltings, S. Shih, and G. Schmitt. Adaptation of spatial design cases. In J.S. Gero, editor, *Artificial Intelligence in design '92*, pages 559–575, Dordrecht, 1992. Papers from the Second International Conference on Artificial Intelligence in Design, Pittsburgh, June 1992, Kluwer Academic Publishers.

[107] J. Barber, S. Bhatta, A. Goel, M. Jacobson, M. Pearce, L. Penberthy, M. Shankar, R. Simpson, and E. Strouliai. Askjef: integration of case-based reasoning and mutimedia technologies for interface design supportt. In J.S.Gero, editor, *Artificial Intelligence in Design '92*, pages 457–475, Dordrecht, 1992. Papers from the Second International Conference on Artificial intelligence in Design, Pittsburgh, June 1992, Kluwer Academic Publishers.

[108] R. Oxman and R. Oxman. Precedents: Memory structure in design case libraries. In U. Flemming and S. Van Wyk, editors, *CAAD futures '93*, chapter 4, pages 273–287. Elsevier Science Publishers B.V., 1993.

[109] X. Liu and M. Gan. A preliminary structural design expert system (sred-1) based on neural networks. In J.S. Gero, editor, *Artificial Intelligence in Design '91*, pages 785–799, Oxford, UK, 1991. Papers from the First International Conference on Artificial Intelligence in Design, Edinburgh, UK, June 1991, Butterworth-Heinemann Ltd.

[110] R.D. Coyne and S. Newton. A tutorial on neural networks and expert systems for design. In J.S. Gero and F. Sudweeks, editors, *Expert Systems in Engineering, Architecture and Construction*, pages 321–337, University of Sydney, Sydney, Australia, 1989.

[111] R.D. Coyne, S. Newton, and F. Sudweeks. Modelling the emergence of schemas in design reasoning. In *Preprints of Modlling Creativity and Knowledge-Based Creative Design*, pages 173–205, Design Computing Unit, Department of Architecture and Design Science, University of Sydney, Sydney, Australia, 1989.

[112] S. Slade. Case-based reaoning: A research paradigm. *AI Magazine*, 12(1):42–55, Spring 1991.

[113] R.F. Harrison. Neurocomputing - an introduction. In D.A. Linkens and R.I. Nicolcon, editors, *Trends in Information Technology*, pages 181- 197. Peter Peregrins Ltd on behalf of the Institution of Electrical Engineers, London, UK, 1990.

[114] T. Yagiu. *Modelling design objects and processes.* Computer Graphics - Systems and Applications, Springer-Verlag, Berlin, 1991.

[115] D. Hartzband. Enhancing knowledge representation in engineering databases. *IEEE Computer*, pages 39–48, Fall 1985.

[116] D.H. Fishman. *The DBMS-Expert system connnection*, pages 87–101. Ablex Publishing Corp., Norwood N.J., 1986.

[117] C. Zaniolo, H. Ait-kaci, D. Beech, S. Cameratera, L. Kerschberg, and D. Maier. Object oriented database systems and knowledge systems. In *Proceedings of the First International Workshop on Expert Database Systems*, 1986.

[118] G.P. Zarri. Construction and utilising large fact databases using artificial intelligence techniques. In *Proceedings of the First International Workshop on Expert Database Systems*, pages 141–159, 1986.

[119] A. Al-Zobaidie and J.B. Grimson. Expert systems and database systems: How can they serve each other? *Expert Systems*, pages 30–37, February 1987.

[120] K.R. Dittrich and U. Dayal. International workshop on object oriented database systems. IEEE, 1986.

[121] B.B. Jackson. *Multivariate Data Analysis: An Introduction.* Richard D. Irwin, Inc., Homewood, Illinois, 1983.

[122] A.I. Comrey. *A First Course in Factor Analysis.* Academic Press, New York, 1973.

[123] B. Everitt. *Cluster Analysis.* Halsted Press, New York, second edition, 1980.

[124] C.C. Chatfield and A.J. Collins. *Introduction to Multivariate Analysis.* Chapman and Hall, 1980.

[125] P.S. Katsoulis. Optimising block coefficient by an exponential formula. *Shipping World and Shipbuilder*, February 1975.

[126] H. Yoshikawa and T. Koyama. Artificial intelligence in design. In *Proceedings of Theory and Practice of Marine Design*, volume 1, pages 23–29. IMSDC, 1982.

[127] A. Williams. Analysis and storing of ship propulsion data. Preprints of International Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design (ICCAS), Session 2, Paper 2, August 1973.

[128] K. Aihara and N. Sugawara. An interactive real time design procedure in ship initial design. Preprints of the International Conference on Computer Applicaitons in the Automation of Shipyard Operation and Ship Design, August 1973.

[129] J.G. Carbonnel, R.S. Michalski, and T.M. Mitchell. Machine learning: A historical and methodological analysis. *AI Magazine*, 4(3):69–79, Fall 1983.

[130] S. Kocabas. A review of learning. *The Knowledge Engineering Review*, 6(3):195–222, 1991.

[131] S. McLaughlin and J.S. Gero. Acquiring expert knowledge from characterised designs. *AI EDAM*, 1(2):73–87, 1987.

[132] A. Radford, P. Hung, and J.S. Gero. New rules of thumb from computer-aided strucutral design: Acquiring knowledge for expert systems. In J. Wexler, editor, *CAD'84*, pages 558–566, Guilford, 1984. Butterworths.

[133] C.A. Mackenzie and J.S. Gero. Learning design rules from decisions and performances. *Artificial Intelligence in Engineering*, 2(1):2–10, 1987.

[134] S.Z. Kamal, J. Mistree, F. amd Sorab, and W.E. VanArsdale. The development of an inductive learning system for design using experimental information. In J.S. Gero, editor, *Artificial Intelligence in Design*, pages 521–538. Proceedings of the Fourth International Conference on the Applications of Artificial Intelligence in Engineering, Cambridge, UK, Springer-Verlag, July 1989.

[135] T.G. Dietterich and R.S. Michalski. A comparative review of selected methods for learning from examples. In R.S. et al Michalski, editor, *Machine Learning: An Artificial Intelligence Approach*, pages 41–81. Morgan Kaufman, 1983.

[136] R.S. Michalski. A theory and methodology of inductive learning. In R.S. et al Michalski, editor, *Machine Learning: An artificial intelligence approach.* Morgan Kaufman, 1983.

[137] Y. Reich and S.J. Fenves. The formation and use of abstract concepts in design. In D. Fisher and M. Pazzani, editors, *Concept Formation: Knowledge and experience in unsupervised learning*, chapter 12, pages 2–31. Morgan Kauffman, 1991.

[138] N. Ivezic and J.H. Garret. A neural network apporach for acquiring and using synthesis knowledge. Technical report, Department of Civil Engineering, Carnegie Mellon University, Pittsburgh, PA, USA, 1992.

[139] R.B. Rao and R.E. Lu, S. C-Y amd Stepp. Knowledge-based equation discovery in engineering domains. In L.A. Birnbaum and G.C. Collins, editors, *Proceedings of the Eighth International Workshop on Machine Learning*, pages 630–634, CA, USA, 1991. ML 91.

[140] F. Arbab. Design object modelling. In H. Yoshikawa and D. Gossard, editors, *Intelligent CAD I*, pages 3–12. Proceedings of the IFIP TC 5/WG5.2 Workshop on Intelligent CAD, Boston, MA, USA, October 1987, North-Holland, 1989.

[141] StatSci Europe. S-plus: User's manual vol. 1. and vol 2. Statistical Sciences UK Ltd, 52 Sandfield Road, Headington, Oxford, UK, October 1991.

[142] CAD Centre. The DESIGNER system: System notes. CAD Centre, Department of Design, Manufacture and Engineering Management, University of Strathclyde, Glasgow, UK, June 1989.

[143] R.A. Becker, J.M. Chambers, and A.R. Wilks. *The New S Language: A programming environment for data analysis and graphics.* Wadsworth and Brooks/Cole Advanced Books and Software, Pacific Grove, California, USA, 1988.

[144] D.H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(7):139–172, 1987.

[145] P. McBride. The LISPVIEW Grapher. Technical Report M/S MTV12-40, SunPro, A Sun Microsystems, Inc. Business, 2550 Garcia Ave. Mountain View, California, USA, 1991.

[146] Wolfram J. Ship design course notes: Department of Ship and Marine Technology, University of Strathclyde, Glasgow, UK, 1987.

[147] A.H.B. Duffy and K.J. MacCallum. Computer representation of numerical expertise for preliminary design. *Marine Technology*, 26(4):289–302, 1989.

[148] J.R. Scott. Some aspects of multiple regression analysis. Vickers Ltd, St Albans Tank, 1967.

[149] V. Hubka and W.E. Eder. *Theory of Technical Systems: A total concept theory for engineering design.* Springer-Verlag, Berlin Heidelberg, revised english edition, 1988.

[150] W. Muller. Design discipline and the significance of visuo-spatial thinking. *Design Studies*, 10(1):12–23, January 1989.

[151] K.J. MacCallum and A.H.B. Duffy. Rapid capture and use of geometrical concepts in design. Project Proposal, 1992. CAD Centre, Department of Design, Manufacture and Engineering Management, University of Strathclyde, Glasgow, UK.

[152] F. Guena and K. Zreik. Analogy, exporation and generalisation: three activities for knowledge-based architectural design systems. In U. Flemming and S. Van Wyk, editors, *CAAD Futures'93*, pages 255–271. Elsevier Science Publishers B.V. (North-Holland), 1993.

[153] B.A. Babin and R. Loganantharaj. Designer's workbench: A tool to assist in the capture and utilisation of design knowledge. In J.S. Gero, editor, *Artificial Intelligence in Design*, pages 249–267, Oxford, UK, 1991. Papers from the First International Conference in Artificial Intelligence in Design, Edinburgh, UK, June 1991, Butterworth-Heineman Ltd.

Appendices

A - Listing of Domain Description File

```
(add '(beam depth engcost engl length engwt comp rpm vs))
(unitsin "metres" beam)
(unitsin "metres" depth)
(meaningin "main engine cost" engcost)
(unitsin "pounds" engcost)
(meaningin "length of engine" engl)
(unitsin "metres" engl)
(unitsin "metres" length)
(meaningin "weight of engine" engwt)
(unitsin "tonnes" engwt)
(meaningin "number of crew" comp)
(unitsin "revs/min" rpm)
(meaningin "cruise speed" vs)
(unitsin "knots" vs)

(add 'consrate)
(meaningin "fuel consumption rate" consrate)
(unitsin "kg/kw.hr" consrate)
(update '(consrate 0.232))

(add 'ddratio)
(meaningin "dwt/disp ratio" ddratio)
(update '(ddratio 0.73))

(add 'dist)
(meaningin "distance travelled during voyage" dist)
(unitsin "nt.miles" dist)
(update '(dist 22000))

(add 'docktime)
(meaningin "time spent in dock refitting etc" docktime)
(unitsin "days" docktime)
(update '(docktime 10))

(add 'endurance)
(unitsin "nt.miles" endurance)

(add 'eta)
(meaningin "machinery efficiency" eta)
(update '(eta 0.97))

(add 'irate)
(meaningin "interest rate" irate)
(update '(irate 0.12))

(add 'lstime)
(meaningin "time lost due to manoevring etc" lstime)
(unitsin "hrs" lstime)
(update '(lstime 24))

(add 'lyears)
(meaningin "excepted life span of ship" lyears)
(unitsin "years" lyears)
(update '(lyears 15))

(add 'nport)
(meaningin "number of ports per voyage" nport )
(update '(nport 2))

(add 'pbunk)
(meaningin "cost of bunkering" pbunk)
(unitsin "pounds/tonne" pbunk)
(update '(pbunk 236))
```

```
(add 'pdies)
(meaningin "price of diesel oil" pdies)
(unitsin "pounds/tonne" pdies)
(update '(pdies 325))

(add 'plub)
(meaningin "price of lubricant oil" plub)
(unitsin "pounds/tonne" plub)
(update '(plub 236))

(add 'prtime)
(meaningin "time spent in port" prtime)
(unitsin "hours" prtime)
(update '(prtime 18))

(add 'rtbunk)
(meaningin "rate of bunkering" rtbunk)
(unitsin "kg/kw.hr" rtbunk)
(update '(rtbunk 0.213))

(add 'rtdies)
(meaningin "rate of diesel consumption" rtdies)
(unitsin "kg/kw.hr" rtdies)
(update '(rtdies 0.213))

(add 'scon)
(meaningin "service condition factor" scon)
(update '(scon 1.314))

(add 'w)
(meaningin "actual to standard sheer ratio" w)
(update '(w 0.87))

(add 'fbd)
(meaningin "actual fbd of the vessel" fbd)
(unitsin "metres" fbd)
(define 'fb '(depth draught) '(- depth draught))
(use fb fbd)

(add 'disp)
(meaningin "displacement of the ship" disp)
(unitsin "tonnes" disp)
(define 'dsp '(cb length beam draught) '(* cb length beam draught 1.025))
(use dsp disp)

(add 'cb)
(meaningin "block coefficient" cb)
(define 'block '(vs length)
               '(- 0.968 (* 0.269 (/ vs (expt (/ length 0.3048) 0.5)))))
(setf (relation-unreliability block) 2)
(use block cb)

(add 'cin)
(meaningin "inertia coefficient of waterplane" cin)
(define 'i '(cw) '(- (* 0.1385 cw) 0.0552))
(use i cin)

(add 'cp)
(meaningin "prismatic coefficient" cp)
(define 'pris '(cb) '(+ 0.04 (* 0.96 cb)))
(use pris cp)

(add 'cw)
(meaningin "waterplane coefficient" cw)
(define 'cwss '(cp)
              '(cond (  (< cp 0.85) (+ (* 0.878 cp) 0.1733))
                     (t (+ (* 0.6 cp) 0.4))))
(setf (relation-unreliability cwss) 3)
(use cwss cw)
```

```
(add 'draught)
(meaningin "estimate of draught with fixed ddratio" draught)
(unitsin "metres" draught)
(define 'drt '(dwt ddratio length beam cb)
             '(/ dwt (* ddratio 1.025 length beam cb)))
(setf (relation-unreliability drt) 5)
(use drt draught)
(userel independent draught)

(add 'bfbd)
(meaningin "basic freeboard" bfbd)
(unitsin "m-mtres" bfbd)
(define 'bas '(cbfbd tabfbd)
             '(cond (   (> cbfbd 0.68) (* tabfbd (/ (+ cbfbd 0.68) 1.36)))
                    (t tabfbd)))
(use bas bfbd)

(add 'cbfbd)
(meaningin "block coefficient at t=0.85d" cbfbd)
(unitsin "m-mtres" cbfbd)
(define 'cbd '(cb depth draught)
             '(+ cb
                 (* (- (* 0.85 (/ depth draught)) 1) 0.086)
                 (* (- 0.7 cb) 0.0475)))

(use cbd cbfbd)

(add 'dcor)
(meaningin "freeboard depth correction" dcor)
(unitsin "m-metres" dcor)
(define 'dall '(depth length)
              '(let ((r 0)
                    )
                  (if (<= (+ depth (* 0.0001 length)) (/ length 15))
                      (setf r 0)
                      (setf r 250))
                  (if (< length 120)
                      (setf r (/ length 0.48)))
                  (* (- (+ depth (* 0.0001 length)) (/ length 15)) r)))
(use dall dcor)

(add 'reqfbd)
(meaningin "required freeboard" reqfbd)
(unitsin "metres" reqfbd)
(define 'freeb '(bfbd dcor supcor shcor length)
               '(- (/ (- (+ bfbd dcor shcor) supcor) 1000) (/ length 10000)))
(use freeb reqfbd)

(add 'shcor)
(meaningin "freeboard sheer correction" shcor)
(unitsin "m-metres" shcor)
(define 'sh '(length w) '(* 8.9375 (+ (/ length 3) 10) (- 1 w)))
(use sh shcor)

(add 'supcor)
(meaningin "freeboard superstructure correction" supcor)
(unitsin "m-metres" supcor)
(define 'sup '(length) '(let ((var 0)
                             )
                           (if (< length 120)
                               (setf var (+ 860 (* 5.68 (- length 85))))
                               (setf var 1070))
                           (* var 0.035)))

(use sup supcor)

(add 'tabfbd)
(meaningin "estimated tabular freeboard" tabfbd)
(unitsin "m-metres" tabfbd)
(define 'tb '(length) '(if (and (>= length 180) (< length 270))
                           (- (* 26.3889 length) (* 0.03432 length) 1036)
```

```
                            (if (>= length 270)
                                (- (+ (* 16.3108 length) 335)
                                   (* 0.0158 (expt length 2)))
                                (if (< length 100)
                                    (+ (- (* 17.5778 length) 562)
                                       (* 2.1 (- 100 length)))
                                    (- (* 17.5778 length) 562)))))
(use tb tabfbd)


(add 'chi)
(meaningin "performance factor for trial condition" chi)
(define 'hi '(length) '(if (< (/ (- 1000 (* length 3.28)) 100) 1)
                            0.85
                            (+ 0.85
                               (* 0.00185
                                  (expt (/ (- 1000 (* length 3.28)) 100) 2.5)))))
(use hi chi)

(add 'etad)
(meaningin "propeller efficiency" etad)
(define 'tad '(cb rpm beam draught)
              '(let ((tao (- 1.3 (* 0.55 cb) (* 0.00267 rpm)))
                     (tah (+ 0.385 (* 0.7 cb) (* 0.11 (/ beam draught))))
                    )
                   (if (> cb 0.8)
                       (setf tah (+ 0.945 (* 0.11 (/ beam draught))))
                       (setf tah (+ tah (* 20 (- cb 0.8) (- 1.44 tah)))))
                   (* tao tah 1.01)))
(use tad etad)

(add 'pd)
(meaningin "delivered power" pd)
(unitsin "kw" pd)
(define 'y '(pdb ppb) '(* pdb ppb))
(use y pd)

(add 'pdb)
(meaningin "effective power" pdb)
(unitsin "kw" pdb)
(define 'db '(chi length beam draught cb vt etad)
             '(* (/ (* 76 chi 0.71
                       (expt (* length beam draught cb 1.0137) 0.667)
                       (expt vt 3))
                    (* 75 427.1 etad))
                 0.7457))
(use db pdb)

(add 'ppb)
(meaningin "power correction factor from trial to service" ppb)
(define 'pb '(vs vt) '(expt (/ vs vt) (* 4.167 (/ vs vt))))
(use pb ppb)

(add 'power)
(meaningin "shaft power in service conditions" power)
(unitsin "kw" power)
(define 'pow '(scon pd eta) '(* scon (/ pd eta)))
(use pow power)

(add 'vt)
(meaningin "trial speed" vt)
(unitsin "knots" vt)
(define 'v '(cb length) '(* (- 1.7 (* 1.4 cb)) (expt (/ length 0.3048) 0.5)))
(use v vt)

(add 'bm)
(meaningin "distance from centre of bouyancy and metacentre" bm)
(unitsin "metres" bm)
(define 'buoy '(cin cb beam draught) '(* (/ cin cb)
                                         (/ (expt beam 2) draught)
```

```
                                         0.913))
(setf (relation-unreliability buoy) 5)
(use buoy bm)

(add 'gm)
(meaningin "metacentre height" gm)
(unitsin "metres" gm)
(define 'stab '(kb bm kg) '(- (+ kb bm) kg))
(use stab gm)

(add 'kb)
(meaningin "distance from keel to centre of buoyancy" kb)
(unitsin "metres" kb)
(define 'kbuoy '(draught cb cw) '(* draught (- 0.71 (/ (* 0.21 cb) cw)) 0.913))
(setf (relation-unreliability kbuoy) 6)
(use kbuoy kb)

(add 'kg)
(meaningin "distance from keel to centre of gravity" kg)
(unitsin "metres" kg)
(define 'g '(depth) '(* 0.57 depth 0.913))
(setf (relation-unreliability g) 4)
(use g kg)

(add 'lhold)
(meaningin "length of holds" lhold)
(unitsin "metres" lhold)
(define 'lhd '(length engl) '(- length (+ (* 0.09 length ) engl 12.5)))
(use lhd lhold)

(add 'vol)
(meaningin "bale capacity of holds" vol)
(unitsin "m^3" vol)
(define 'capt '(lhold beam depth cb ddratio)
              '(* 1.227 lhold beam depth (+ (/ cb 2) 0.5) ddratio))
(setf (relation-unreliability capt) 5)
(use capt vol)

(add 'amatl)
(meaningin "area of longl material at midships" amatl)
(unitsin "cm^2" amatl)
(define 'la '(length beam depth draught)

'(* (- (+ (- (+ (* 0.1512
                   (log (/ length 0.3048)))
                (* 0.0862
                   (log (/ beam 0.3048))))
             (* 0.0204
                (log (/ depth 0.3048))))
          (* 0.0121
             (log (/ draught 0.3048))))
       0.0095)
    10000
    (expt 0.3048 2))

)

(use la amatl)

(add 'dwt)
(meaningin "cargo deadweight" dwt)
(unitsin "tonnes" dwt)
(define 'dw '(disp lwt fuelwt supwt) '(- disp lwt fuelwt supwt))
(use dw dwt)

(add 'fuelwt)
(meaningin "fuel weight" fuelwt)
(unitsin "tonnes" fuelwt)
(define 'fwt '(endurance vs consrate power)
             '(/ (* endurance 1.852 consrate power) (* vs 0.515 3600)))
```

```
(setf (relation-unreliability fwt) 10)
(use fwt fuelwt)

(add 'lwt)
(meaningin "lightship weight" lwt)
(unitsin "tonnes" lwt)
(define 'lght '(ws wo wm) '(+ ws wo wm))
(use lght lwt)

(add 'supwt)
(meaningin "weight of supplies, consumption etc" supwt)
(unitsin "tonnes" supwt)
(define 'swt '(comp endurance vs) '(/ (* 1.016 comp endurance 0.015) vs))
(setf (relation-unreliability swt) 14)
(use swt supwt)

(add 'wm)
(meaningin "machinery weight" wm)
(unitsin "tonnes" wm)
(define 'm '(engwt power) '(+ engwt
                              (* (* (+ (/ power
                                          (* 35 0.7457))
                                       200)
                                    1.016)
                                 0.8809)))
(setf (relation-unreliability m) 14)
(use m wm)

(add 'wo)
(meaningin "outfit weight" wo)
(unitsin "tonnes" wo)
(define 'outwt '(length beam)
               '(* 1480 1.016 0.8809 (+ 0.25 (* 0.75
                                                (/ length (* 760 0.3048))
                                                (/ beam (* 104 0.3048))))))
(setf (relation-unreliability outwt) 14)
(use outwt wo)

(add 'ws)
(meaningin "steel weight" ws)
(unitsin "tonnes" ws)
(define 's '(amatl length beam depth cb draught)

'(* (+ (* 11180
          (/ length (* 760 0.3048))
          (/ amatl (* 5835.8 (expt 0.3048 2))))
       (* 11180
          (/ length (* 760 0.3048))
          (/ beam (* 104 0.3048))
          (/ depth (* 59 0.3048))
          (+ 1 (* 0.5 0.82))
          (/ (expt (/ (* length 59) (* 760 depth)) 0.5) (+ 1 (* 0.5 cb))))
       (* 1.125 (expt (/ length 0.3048) 1.65) (/ (+ beam depth (/ draught 2))
                                                 0.3048)))
    (/ (+ (* 0.5 cb) 0.4) 800)
    1.016
    (/ 0.8809 3))
)
(setf (relation-unreliability s) 14)
(use s ws)
```

B - Listing of Goals File

```
(defgl 'dwtgl '(dwt >= 26900) 'dwtgl1)
(importin 80 dwtgl)
(defgl 'endgl '(endurance = 19200) 'endgl1)
(importin 50 endgl)
(defgl 'fbdgl '(fbd > reqfbd) 'fbdgl1)
(defgl 'gmgl '(gm > beam * 0.06) 'gmgl1)
(importin 80 gmgl)
(defgl 'vsgl '(vs = 15) 'vsgl1)
(importin 90 vsgl)
```

C - Partial Listing of Set of Past Designs File

```
example
ex-69
positive
DRAUGHT 12.937
DEPTH 17.8
DISPF 77500.0
VS 14.9
LENGTH 215.4
BEAM 32.2
ENDURANCE 24000.0
FCP 45.5
GT 34844.92
NT 24691.9
DWTF 64219.0
CB 0.838
KG 9.8
GM 3.41
CP 0.838
L/B 6.689
L/D 12.101
B/D 1.809
LWT ?
CW ?
POWER 13680.0
COMP 33
ENGWT 436.0
RPM 112.0
APPROXLHOLD 108.8
VOL ?
ENGL 11.2
end

.
.
.

example
ex-93
positive
DRAUGHT 16.352
DEPTH 23.8
DISPF 153387.0
VS 14.55
LENGTH 260.0
BEAM 43.0
ENDURANCE 29710.0
FCP 58.3
GT 63076.0
NT 51068.0
DWTF 133361.0
CB ?
KG ?
GM ?
CP 0.8213
L/B 6.047
L/D 10.924
B/D 1.807
LWT ?
CW ?
POWER 18400.0
COMP 35
ENGWT 580.0
RPM 106.0
APPROXLHOLD 124.02
```

```
VOL ?
ENGL ?
end

end-examples
```

D - Syntax of Attribute-Info File

```
(defparameter *design-description*  '(<attribute-name>...<attribute-name>))
(defparameter *specifications* '(<attribute-name>...<attribute-name>))

(setf (gethash '<attribute-name> *property-types*) 'continuous)
.
.
.
(setf (gethash '<attribute-name> *property-types*) 'continuous)

(setf (gethash '<attribute-name> *expected-interval*) <max-min-distance>)
.
.
.
(setf (gethash '<attribute-name> *expected-interval*) <max-min-distance>)
```

E - Example Listing of Behaviour-Definition File

```
(setq *continuous-classes* 10)
(setq *prediction-method* 'leaf)
(setq *P.v-cl* 0.75)
(setq *P.cl-v* 0.75)
(setq *continuous-ranges* 'static)
(setq *prediction-groups* 'nil)
```

F - Listing of Set of Attribute-Types File

```
attribute-types
DRAUGHT     continuous
DEPTH       continuous
DISPF       continuous
VS          continuous
LENGTH      continuous
BEAM        continuous
ENDURANCE   continuous
FCP         continuous
GT          continuous
NT          continuous
DWTf        continuous
CB          continuous
KG          continuous
GM          continuous
CP          continuous
L/B         continuous
L/D         continuous
B/D         continuous
LWT         continuous
CW          continuous
POWER       continuous
COMP        continuous
ENGWT       continuous
RPM         continuous
APPROXLHOLD continuous
VOL         continuous
ENGL        continuous
end
```