

**Patina: a Method Oriented
Design Environment
for Parametric Analysis**

Simon Shurville

Submitted as partial fulfilment of the
requirements for a PhD at the
University of Brighton
in November 1998

90 0660882 3



701 SHU

To Shirley for all her encouragement and support

Acknowledgements

Many thanks to my supervisors Dr. John Downie, Dr. Lyn Pemberton, Dr. Richard Power and Professor Donia Scott for their help and guidance throughout my research.

I also thank the Science and Education Research Council and Integral Solutions Limited for the financial support that made this study possible.

Finally, I would like to thank the following admission-tutors for the opportunity to study as a mature student: Willie Taylor (Southwark College), Professor Aaron Sloman (University of Sussex) and Dr. Lyn Pemberton (University of Brighton).

Summary

The central issue to be addressed in this thesis is the provision of support for design tasks that require problem formation and evaluation and some inventive adaptation of products and design strategies. Hitherto, computer tools have failed to support the full range of design tasks. In particular, they have been focused upon solving previously formulated design tasks in well-defined domains where little inventiveness with materials or design strategies is required (Green, 1992a). A solution is offered in the form of an analysis that yields a new class of system, called Method Oriented Design Environments (MODEs), which provide support for some of these more complicated design tasks. An implementation of such a system is presented in the form of Patina: a MODE to support parametric analysis.

It is argued that the lack of support for design tasks involving problem formulation, evaluation and inventiveness with components and strategies has partly been due to usage of an overly narrow view of the design process as a basis for system development on the part of developers of knowledge aided design. To provide a more complete orientation for these developers, and to expand the theory of knowledge aided design, an alternative model of design tasks is developed in the form of a 'design activity space' by transferring knowledge from the field of design research to that of knowledge aided design.

A mapping is constructed between this new design activity space and Green's model space of tools for knowledge aided design (Green, 1992a). The mapping is first used to analyse the range of utility of some recent alternatives to traditional knowledge based systems for design. It is then used to single out a 'niche' of design tasks that are not supported by traditional systems or their more recent alternatives.

The design tasks which lie in this niche awaiting support from computer tools share the following characteristics: (1) they encompass the activities of analysis, synthesis and evaluation, (2) they require an intermediate degree of innovation with the product, and (3) they require an intermediate degree of innovation in design strategy.

The class of tools that are proposed to offer support to tasks in this niche are named MODEs because their defining characteristic is that the majority of their constituent knowledge is derived from a design method or strategy. Therefore the main item that is being represented to the user of a MODE is such a structured method rather than an evolving artefact. This is radically different from the traditional knowledge based tools, where the item being represented is an artefact in a particular domain, and from a recent proposal for systems that depict an unstructured process (Blessing, 1994).

To demonstrate the feasibility of implementing a MODE, the implementation of a system called Patina, to support designers in applying the technique of parametric analysis, is reported.

Contents

	page number
Summary	III
Chapter One: Introduction	
1.0 Overview of the research	1: 1
1.1 Trigger to the research	1: 2
1.2 The hypotheses	1: 3
1.3 The methodology	1: 4
1.4 The work plan	1: 4
1.4.1 Phase one: problem identification and hypothesis generation	1: 5
1.4.2 Phase two: knowledge identification and translation	1: 5
1.4.3 Phase three: demonstrations of concepts	1: 6
1.5 Structure of the thesis	1: 7
1.5.1 Part one: theoretical research	1: 7
1.5.2 Part two: putting theory into practice	1: 9
1.6 Contributions to knowledge	1: 10
1.7 Summary	1: 11
Chapter Two: Literature review: histories of design, design research and knowledge aided design	
2.0 Overview	2: 1
2.1 A history of design from the Classical period to the end of the Second World War	2: 1
2.2 Post war design research in Great Britain and the West	2: 5
2.3 Artificial intelligence, knowledge based design and knowledge aided design	2: 11
2.4 Prospects for synthesis between design research and knowledge aided design	2: 24
2.5 Summary	2: 26
Chapter Three: Creating a design activity space	
3.0 Overview	3: 1
3.1 Motivation and method	3: 2
3.1.1 Similarities and differences with prior models	3: 2
3.2 Naive diagrams of the structure of design activities	3: 3
3.3 Taxonomies of design, creativity and inventiveness	3: 9
3.4 First and second generation design methods	3: 14
3.4.1 First generation design methods	3: 14
3.4.2 Second generation design methods	3: 22
3.4.3 Implications of first and second generation design methods for knowledge aided design	3: 28
3.5 Synthesising the design activity space	3: 29
3.5.1 The first axis: a generic three stage model of design	3: 29
3.5.2 The second axis: how much innovation do various design activities require?	3: 30

3.5.3	The third axis: first generation and second generation design methods	3: 31
3.5.4	Some caveats	3: 33
3.6	Summary	3: 35

Chapter Four: Mapping the space of design activities onto a set of knowledge aided design tools and a model space of knowledge aided design systems

4.0	Overview	4: 1
4.1	Green's tools and technologies for knowledge aided design	4: 1
4.1.1	Green's tools for knowledge aided design	4: 3
4.1.2	Green's tools for presenting design knowledge	4: 3
4.1.3	Green's tools for collecting design knowledge	4: 5
4.1.4	Adding a design-visualiser to Green's tool set	4: 6
4.2	Green's model space of technologies for knowledge aided design	4: 9
4.2.1	The first dimension of Green's model space	4: 9
4.2.2	The second dimension of Green's model space	4: 10
4.2.3	The third dimension of Green's model space	4: 11
4.2.4	The fourth dimension of Green's model space	4: 12
4.2.5	The fifth dimension of Green's model space	4: 13
4.2.6	The sixth dimension of Green's model space	4: 13
4.2.7	The seventh dimension of Green's model space	4: 14
4.2.8	Implications of Green's model space for developers of knowledge aided design systems	4: 15
4.3	Mapping the space of design activities onto the space of technologies for knowledge aided design	4: 16
4.3.1	An overview of tools for the first axis: a generic three stage model of design	4: 17
4.3.2	Tools for supporting the stage of analysis	4: 17
4.3.3	Tools for supporting the stage of synthesis	4: 18
4.3.4	Tools for supporting the stage of evaluation	4: 20
4.3.5	Tools for supporting variant-design activities	4: 21
4.3.6	Tools for supporting innovative-design activities	4: 23
4.3.7	Tools for supporting strategic-design activities	4: 24
4.3.8	Tools for supporting tame design activities	4: 26
4.3.9	Tools for supporting wicked design activities	4: 28
4.4	Summary	4: 30

Chapter Five: Identifying a niche in the design activity space for MODEs

5.0	Overview	5: 1
5.1	Method	5: 2
5.2	Estimating the range of utility of DODEs	5: 3
5.2.1	The range of utility of DODEs on the axis concerned with innovation	5: 3
5.2.2	The range of utility of DODEs on the axis concerned with analysis, synthesis and evaluation	5: 7
5.2.3	The range of utility of DODEs on the axis concerned with design methodology and evaluation	5: 8
5.3	Could an alternative approach to implementing DODEs with user seeding and end user programming extend their range of utility?	5: 9

5.4	Estimating the range of utility for process based design environments	5: 12
5.4.1	The functional specification of the PROSUS system	5: 13
5.4.2	The viability of the design matrix for various types of design	5: 16
5.5	The identification of a niche in the design activity space	5: 18
5.6	The proposal of MODEs	5: 21
5.6.1	Comparisons and contrasts between MODEs, DODEs and CITIS	5: 22
5.6.2	Comparisons and contrasts between MODEs and the process based approach	5: 24
5.6.3	An opportunity for further research	5: 26
5.7	Chapter summary and conclusions	5: 26

Chapter Six: The development of Patina: triggers and analysis

6.0	Overview	6: 1
6.1	The context of parametric analysis	6: 2
6.2	The purpose of parametric analysis	6: 3
6.3	The content and output of parametric analysis	6: 6
6.4	Placing parametric analysis in the design activity space	6: 13
6.4.1	Parametric analysis and the first dimension of the design activity space	6: 14
6.4.2	Parametric analysis and the second dimension of the design activity space	6: 15
6.4.3	Parametric analysis and the third dimension of the design activity space	6: 17
6.5	Three stages in parametric analysis which might benefit from knowledge aided support	6: 18
6.5.1	Three points of wickedness within parametric analysis	6: 19
6.5.2	An area of complexity within parametric analysis	6: 21
6.6	Summary	6: 23

Chapter Seven: The development of Patina: a MODE for parametric analysis

7.0	Overview	7: 1
7.1	Development method	7: 1
7.1.1	Reasons for implementing a high fidelity feasibility prototype	7: 2
7.1.2	Reasons for following an object oriented methodology	7: 3
7.2	Concept design	7: 4
7.2.1	Suitability of parametric analysis for implementation as a MODE	7: 4
7.2.2	Reasons to base the structure of Patina's interface on the structure of parametric analysis	7: 4
7.2.3	Reasons to add a passive design-informer in the format of a flow chart as a main interface metaphor with embedded design-informers and secretaries	7: 5

7.2.4	Reasons to address the wickedness of designing a parameter set by including an interactive design-informer	7: 7
7.2.5	Reasons to address the wickedness and complexity of analysing a parameter set by including a rug plot	7: 8
7.2.6	Reasons to make design-secretaries and design-documenters available in the product set, the parameter set and the parametric paintings	7: 10
7.2.7	Reasons to include a representation of the PDS document	7: 10
7.2.8	A summary of the conceptual design decisions	7: 11
7.3	Detail design	7: 11
7.3.1	The choice of programming language and development environment	7: 11
7.3.2	The graphical user interface	7: 12
7.3.3	The detail design of the introductory screen	7: 12
7.3.4	The detail design of the combined home page and menu bar	7: 14
7.3.5	The detail design of the file menu	7: 16
7.3.6	The detail design of the dialog window for creating a parametric analysis	7: 16
7.3.7	The detail design of the commands on the knowledge base menu	7: 17
7.3.8	The detail design of the design team window and its children	7: 18
7.3.9	The detail design of the product set window and its children	7: 21
7.3.10	The detail design of the parameter set window and its children	7: 23
7.3.11	The detail design of the parametric easel window	7: 30
7.3.12	The detail design of the PDS document window	7: 33
7.4	An informal evaluation	7: 33
7.5	Discussion	7: 35
7.6	Summary	7: 36
7.6.1.	Suggestions for further work	7: 36

Chapter Eight: Conclusion

8.0	Overview	8: 1
8.1	Outcomes of describing design using knowledge from design research	8: 1
8.2	Outcomes of using the description of design tasks to classify tools and systems for knowledge aided design	8: 3
8.3	Outcomes of identifying a need for a new class of system and of proposing MODEs	8: 8
8.4	Outcomes of implementing Patina ~ a MODE for parametric analysis	8: 11
8.5	Further research	8: 13

List of figures

Fig: 2.1	The proposed interface to Blessing's PROSUS system (from Blessing, 1994, p 282)	2: 19
Fig: 2.2	"Six design sub-processes defined as a main menu" (a card from a Hypercard system, re-drawn from Gilleard and Lee, 1996, p 48)	2: 20
Fig: 2.3	Modified main menu showing Classification of Fire Hazard as the focus of navigation (a card from a Hypercard system, re-drawn from Gilleard and Lee, 1996, p 48)	2: 21
Fig: 2.4	The proposed interface to Candy and Edmond's system to support creative design (from Candy and Edmonds, 1996, p 86)	2: 23
Fig: 3.1	A general purpose model of design (re-drawn from Lawson, 1990) showing the stages of analysis, synthesis and evaluation being linked by edges that represent the flow of work between these stages	3: 4
Fig: 3.2	Each stage in the generic three stage model of design contains a smaller version of the entire graph	3: 6
Fig: 3.3	A diagram re-drawn from Hertz (1992, p 405) which shows some of the additional variables that are introduced by endorsing a more 'realistic' model of the structure of design activities	3: 7
Fig: 3.4	A continuum of design activities	3: 9
Fig: 3.5	A continuum of design activities represented as a histogram	3: 10
Fig: 3.6	Problem-pull and solution-push projects	3: 12
Fig: 3.7	'The designer as a computer', adapted from Jones (1992, p 50)	3: 16
Fig: 3.8	The designer as black-box	3: 17
Fig: 3.9	A hybrid model of the designer following a procedural design method	3: 21
Fig: 3.10	Two meta-cards re-drawn from Crickmay and Jones (1972)	3: 24
Fig: 3.11	The generic three stage model of design	3: 29
Fig: 3.12	The model of the amount of innovation required by four types of design activities	3: 31
Fig: 3.13	First generation and second generation design methods	3: 32
Fig: 3.14	The three dimensional space of design activities	3: 34
Fig: 4.1	A hierarchy showing Green's two types of knowledge aided design tools	4: 2
Fig: 4.2	The first dimension in Green's model space	4: 9
Fig: 4.3	The second dimension in Green's model space	4: 10
Fig: 4.4	The third dimension in Green's model space	4: 11
Fig: 4.5	The fourth dimension in Green's model space	4: 12
Fig: 4.6	The fifth dimension in Green's model space	4: 13
Fig: 4.7	The sixth dimension in Green's model space	4: 14
Fig: 4.8	The seventh dimension in Green's model space	4: 15
Fig: 5.1	The interface to Fischer's Janus-CRACK system (re-drawn from Fischer, 1993, p 244)	5: 5
Fig: 5.2	The interface to Silverman's system for placing antenna on warships (re-drawn from Silverman, 1992, p 110)	5: 5

Fig: 5.3	This diagram illustrates the estimated range of utility of DODEs along two important axes of the design activity space	5: 8
Fig: 5.4	This diagram presents a simplified version of Blessing's design matrix (adapted from Blessing, 1996, p 116)	5: 15
Fig: 5.5	This diagram illustrates the estimated range of utility of process based environments along two important axes of the design activity space	5: 18
Fig: 5.6	This diagram illustrates a niche between the regions in the design activity space that are supported by DODEs and the process based approach	5: 21
Fig: 5.7	This diagram illustrates the suggestion that MODEs should be placed between DODEs and the process based approach	5: 22
Fig: 6.1	A diagram of the product life cycle (re-drawn from Hollins and Hollins, 1991, p 93)	6: 3
Fig: 6.2	A diagram of the total design method highlighting the stage of market research	6: 2
Fig: 6.3	A diagram showing the three iterative stages of parametric analysis	6: 6
Fig: 6.4	A diagram highlighting the prescriptive stage of parametric analysis	6: 7
Fig: 6.5	A diagram highlighting the empirical stage of parametric analysis	6: 10
Fig: 6.6	A diagram highlighting the analytical stage of parametric analysis	6: 10
Fig: 6.7	A diagram of the total design method showing the performance of a parametric analysis, the generation of a product design specification and the stages where it is consulted	6: 11
Fig: 6.8	Three pages from a hypothetical PDS (adapted from Pugh, 1990)	6: 12
Fig: 6.9	The three-dimensional space of design activities	6: 18
Fig: 7.1	Patina's introductory screen	7: 13
Fig: 7.2	Patina's home page and menu bar	7: 15
Fig: 7.3	Patina's file menu	7: 15
Fig: 7.4	Patina create's a parametric analysis dialog window	7: 16
Fig: 7.5	Patina's 'knowledge base' menu	7: 17
Fig: 7.6	Patina's design team window	7: 18
Fig: 7.7	Patina's add a designer profile dialog window	7: 20
Fig: 7.8	Patina's delete a designer profile dialog window	7: 20
Fig: 7.10	The update a product window showing the process of adding an optional note	7: 22
Fig: 7.11	The parameter set window	7: 11
Fig: 7.12a	The note book representing the parameter set	7: 24
Fig: 7.12b	The visualisation of the values and statistics for a particular parameter	7: 24
Fig: 7.13	An interactive design-informer which helps the design team to select their parameters	7: 26
Fig: 7.14	The add a parameter to the parameter set window	7: 27
Fig: 7.15	The update a parameter window	7: 27
Fig: 7.16a	The parametric easel	7: 29
Fig: 7.16b	A detail from the parametric easel showing how four scatter plots are 'knitted' together	7: 29
Fig: 7.16c	Three pull-down menus causing 'parameter 1' and 'parameter 2' to be graphed against 'parameter 1'	7: 31

Fig: 7.16d	Erasing a line on a scatter plot within the parametric easel	7: 31
Fig: 7.17	The special design rationale dialog for writing about multiple parameters	7: 32
Fig: 7.18	The PDS document window	7: 32
		page number

List of Tables

Table 6.1	A set of standard parameters that would be expected in a PDS for an engineering product (from Hollins and Hollins, 1991, p 59)	6: 9
Table 8.1	An indication of the suitability of various tools to design activities	8: 4
Table 8.2	An indication of the suitability of various systems to design activities	8: 7

Chapter One

Introduction

1.0 Overview of the research

The central issue to be addressed in this thesis is the provision of support for design tasks that require problem formation and evaluation and some inventive adaptation of products and design strategies. Hitherto, computer tools have failed to support the full range of design tasks. In particular, they have been focused upon solving previously formulated design tasks in well-defined domains where little inventiveness with materials or design strategies is required (Green, 1992a). A solution is offered in the form of an analysis that yields a new class of system, called Method Oriented Design Environments (MODEs), which provide support for some of these more complicated design tasks. An implementation of such a system is presented in the form of Patina: a MODE to support parametric analysis.

It is argued that the lack of support for design tasks involving problem formulation, evaluation and inventiveness with components and strategies has partly been due to usage of an overly narrow view of the design process as a basis for system development on the part of developers of knowledge aided design. To provide a more complete orientation for these developers, and to expand the theory of knowledge aided design, an alternative model of design tasks is developed in the form of a 'design activity space' by transferring knowledge from the field of design research to that of knowledge aided design.

A mapping is constructed between this new design activity space and Green's model space of tools for knowledge aided design (Green, 1992a). The mapping is first used to analyse the range of utility of some recent alternatives to traditional knowledge based systems for design. It is then used to single out a 'niche' of design tasks that are not supported by traditional systems or their more recent alternatives.

The design tasks which lie in this niche awaiting support from computer tools share the following characteristics: (1) they encompass the activities of analysis, synthesis and evaluation, (2) they require an intermediate degree of innovation with the product, and (3) they require an intermediate degree of innovation in design strategy.

The class of tools that are proposed to offer support to tasks in this niche are named MODEs because their defining characteristic is that the majority of their constituent knowledge is derived from a design method or strategy. Therefore the main item that is being represented to the user of a MODE is such a structured method rather than an evolving artefact. This is radically different from the traditional knowledge based tools,

where the item being represented is an artefact in a particular domain, and from a recent proposal for systems that depict an unstructured process (Blessing, 1994).

To demonstrate the feasibility of implementing a MODE, the implementation of a system called Patina, to support designers in applying the technique of parametric analysis, is reported.

1.1 Trigger to the research

The development of the design activity space, MODEs and Patina was triggered by a goal that Marc Green (1992, 1992a) has set for the field of knowledge aided design. This goal is to develop computer tools that incorporate knowledge about design in order to assist or replace designers throughout their work. When Green proposed this goal he noted that existing tools almost always fell into a 'cluster' of similar knowledge based technologies for 'planning' and 'fault diagnosis' that addressed a subset of the possible range of design activities (see tools and systems mentioned in Pugh, 1989; Blessing, 1994; Rychener, 1988; Coyne *et al*, 1990; Winstanley, 1990; Tong and Sriram, 1992) to the exclusion of other computer technologies, such as hypermedia and computer mediated communication. He argued that it would be beneficial if future systems were to address a wider spectrum of design activities by applying a broader range of component technologies.

Green made a significant step by describing a set of tools for knowledge aided design that encompassed knowledge based systems and such alternatives as hypermedia and computer mediated communication and by setting them in a 'model space'. This is because such a classification made his cluster of existing systems apparent. However, it appears that some classificatory work remains to be done. A full scale system for knowledge aided design will be likely to contain a number of specialised tools which address different aspects of the range of design activities encountered during those projects. For this reason, classes of system, such as Fischer and Silverman's Domain Oriented Design Environments (DODEs) (Fischer, 1992, 1994a, 1994b; Silverman, 1992)¹, can be identified that bear a family resemblance to one another at the levels of component selection and applicability in the range of design activities. So it seems a reasonable and needed goal to seek regions in Green's space at the level of systems rather than their component tools. Such regions could characterise the components and range of utility for existing or envisaged classes of system and map territory for new classes of system. Green suggested that new types of tools can be generated by

¹These are discussed in detail in Chapters Two and Five

analysing unexplored regions of the model space. This strategy provides the point of departure for the research reported here which seeks to add a new class of system to the inventory of knowledge aided design.

1.2 The hypotheses

The goal of expanding available range tools and systems for knowledge aided design in order to assist or replace designers *throughout* their work was pursued by seeking to confirm the following hypotheses:

- H 1** It is possible to construct a design activity space whose axes include strategies sourced from design research for characterising and addressing design projects. This new design activity space should enable new insights to be generated into the classes of knowledge aided design systems that can and should be created.
- H 2** It is possible to develop mappings between this design activity space and Green's model space of knowledge aided design systems. These mappings should enable the approximate range of applicability of individual tools and classes of systems to be indicated. These mappings should also make it possible to re-characterise the accepted range of applicability for some existing classes of system.
- H 3** It is possible to search the design activity space for at least one niche that is unsupported by existing classes of system and to propose a new class of system, composed of tools located in Green's model space, that appears suited to supporting activities in this niche. The use of the design activity space should indicate that the particular class that has been proposed is suited to supporting activities in the niche and that the surrounding classes of system are unsuited to offering such support.
- H 4** It is possible to design and implement an example of this new class of system by locating a design practice in the design activity space and then applying the mapping to Green's model space in order to propose an aggregate of components that will support this practice.

1.3 The methodology

This section describes the methodology that was employed to identify and address the hypotheses presented above.

The methodology sets out to suggest new ways of working in knowledge aided design by transferring and translating relevant knowledge from design history and design methodology. It involved taking a critical stance to one branch of academic literature, thereby highlighting a problem, and focusing a search upon other branches of literature for knowledge that could be advantageously transferred to tackle it. Agre (1997) calls such a methodology 'critical technical practice'. The purpose of the critical component in such a methodology is to explain how a particular problem arises by examining relevant technical literature. This literature is analysed for erroneous assumptions or limited working practices whose identification can trigger the proposal of new ground rules and practices to produce improved technical solutions. Such technical solutions are then demonstrated in a practical part of the methodology through implementation of an example.

The methodology was qualitative rather than quantitative because the data involved were the theories and working practices of three academic subject areas: namely those of developers of knowledge aided design systems, design historians, and design researchers. Moreover, the process which was applied to interpret this data and to draw conclusions from it was largely one of motivated inference. The reason for this is that it is the nature of the problem at hand that few implemented systems existed which could be tested, so the point was to suggest theory-based designs for systems that can be implemented and tested in further rounds of research. In short the problematic part of the methodology is a 'chicken and egg' situation in which systems need to be developed in order to generate and support theories of system design and such theories need to be in place in order to develop appropriate systems.

1.4 The work plan

The work plan was structured into three sequential phases with the critical part of the methodology being carried out in the first and second phases and the technical aspect being carried out in the third phase. Phase one involved problem identification and hypothesis generation. Phase two was an interdisciplinary and mainly theoretical phase of knowledge identification and transfer. Phase three was mainly empirical and involved demonstrating the concepts from the previous phases. In this section each of these phases will be motivated and described in detail.

1.4.1 Phase one: problem identification and hypothesis generation

In the first phase, which consisted of problem identification and hypothesis generation, the existence of the problem of under-exploration of the space of technologies for knowledge aided design was noted. It was proposed that this might be addressed by transferring and translating knowledge from the subject areas of design history and design research.

The problem itself was raised by Green in his article *Conceptions and Misconceptions of Knowledge Aided Design* (1992a). The conviction that it could be addressed by changing the nature of the model of design activities was inspired by the work of Goel and Pirolli (1989, 1992). Although Goel and Pirolli were heavily biased toward the information processing approach popularised by Simon (1969), they introduced design theory to the area of modelling design in the cognitive science / artificial intelligence tradition. It appeared worthwhile to follow their lead and see if a model more firmly based on design theory would provide new insights to help address the research agenda Green had set.

Critically reviewing the literature of knowledge aided design led to an intuition that investigating the range of applicability of DODEs and process based design environments in the light of a model informed by design theory might reveal a niche for a new class of system.

1.4.2 Phase two: knowledge identification and translation

The second phase continued the interdisciplinary approach of the work plan and was quite theoretical. It addressed two goals that had been set in phase one. The first was to identify and transfer knowledge from design history and design research in order to construct a model of design activities and ways to approach them for use in knowledge aided design. The second was to analyse this space to motivate the definition of new classes of systems for knowledge aided design and to implement a demonstration of concept for such systems.

The component stages within this phase of the work plan were: (1) a focused literature search in the areas of design history and design research; (2) the construction of a space of design activities that includes design methodology and other accepted ideas from the design research literature and its mapping to an existing space of knowledge aided design tools; (3) the appraisal of existing classes of knowledge aided design in the light of this new model space, and (4) the proposal of a class of knowledge

aided design systems that could address an area of the design task space for which previous knowledge aided design tools were unsuited.

1.4.3 Phase three: demonstrations of concepts

The goal of the third phase of the work plan was to generate material which could be used to illustrate the outcomes of the theoretical sections of the work plan. It was an empirical phase because it involved performing two demonstrations of the use of the model space that was generated in the second phase.

The first of these demonstrations was performed by describing a particular design activity in terms of the new model and then analysing it for components that could benefit from knowledge aided support.

Parametric analysis was chosen because it is taught in British design courses and used by working designers¹. The method was analysed to locate it along the design activity space. Its location in that space matched the region for which MODEs had been proposed and this indicated that a MODE would be an appropriate class of system with which to support the method. The method was also analysed for 'complex' and 'wicked' elements that could benefit from 'knowledge based' and 'knowledge aided' tools². These activities provided an indication of the kinds of tools that should constitute the system.

The second demonstration was performed by implementing a knowledge aided design system on the basis of this analysis to serve as an example of the class of knowledge aided design systems that were proposed during the second phase of the research. The system was implemented using a methodology known as rapid object oriented prototyping.

Patina was developed as a feasibility prototype of a MODE using Connell and Shafer's *Object Oriented Rapid Prototyping* method (1995). The feasibility prototype whose design is reported here was based upon: (1) a literature review of parametric analysis, (2) an interview with Dr William Hollins, (3) discussions and interviews with various designers and engineers at the University of Brighton and local design companies, (4) the theory developed and cited in the first half of the thesis, and (5) an implementation in SmallTalk. This design therefore represents a first iteration of Connell and Shafer's methodology. The result is an operational MODE which supports each stage in parametric analysis.

¹ See the discussion of the method in Chapter Six.

² See the discussion of glass- and black-box methods and their relationship to these types of tool in Chapters Three and Four.

1.5 Structure of the thesis

The thesis is structured into two parts which reflect the critical and technical aspects of the methodology. The first of these is concerned with advancing the theory of design research/knowledge aided design. The second aims to put the results of this theoretical work into practice by analysing a particular design method and building a system to support it. In this section the content of the chapters of the thesis, which constitute each of these aspects of the research, will be introduced then tied together with a summary of the concluding chapter.

1.5.1 Part one: theoretical research

The first part of the thesis builds upon the histories of design research and knowledge aided design to: (1) highlight a possible cause of the tendency of systems to inhabit Green's cluster (Chapter Two), and (2) propose a path toward a resolution that will enable principled exploration of new regions of the whole model space. This path consists of: constructing a space of design activities (Chapter Three), mapping it to Green's model space of technologies (Chapter Four), analysing the range of applicability of the current and envisaged occupants of Green's space (Chapter Five), and proposing a new class of systems which serves a function in the space of design activities and inhabits a new region of the model space (also, Chapter Five). This part of the thesis thus address hypotheses one, two and three.

In Chapter Two the histories and literature of design, design research and knowledge aided design are surveyed in order to: (1) introduce the interdisciplinary material that will be referred to in the core of the thesis to readers from either of the relevant research communities; (2) yield descriptions of the first and second generation methods which will provide much of the knowledge for MODEs in general and for the system to be presented in the later chapters of the thesis; (3) suggest that the cause for the tendency of knowledge aided design systems to fall into the cluster identified by Green is that developers of such systems have relied on an incomplete model of design which mischaracterises optimisation problems as design projects, and (4) describe a range of classes of knowledge aided design systems which offer alternatives to those in Green's cluster of systems, i.e., domain oriented design environments, process based design environments and environments to support creative design.

The chapter sets the key knowledge and technologies in context. It suggests that a new model of design activities might provide insights that offer: (1) a means to characterise the range of applicability of the existing and proposed inhabitants of

Green's space of knowledge aided design systems, and (2) a means to envisage future classes of system that can populate newly identified gaps between these classes.

In Chapter Three a space that describes design activities and which can be used to describe the purposes that specific knowledge aided design systems should serve is constructed from various examples of design research. The space contains three axes whose contents should be familiar to designers who have been trained in the Anglo American tradition and which should be sufficiently uncomplicated to serve as a starting point for the development of knowledge aided design systems:

- (x) The x axis describes a generic three stage model of design activities. The model separates design activities into analysis, synthesis and evaluation.
- (y) The y axis describes a model of the amount of innovation required by four types of design activities: repeat orders, configurative variant-design and true variant-design, innovative-design, and strategic-design.
- (z) The z axis contains the first-generation and second-generation design methods that were explained in Chapter Two's history of design research.

The chapter also explains why this model is designed to be a relatively uncomplicated pilot for larger and more complicated models that constitute goals for future research in this field.

With a space of design activities in hand the next task is to couple it with a suitable space in which to characterise tools and systems for knowledge aided design. So Chapter Four begins by describing a set of tools for knowledge aided design and then describes and critiques a model space that categorises such tools along seven dimensions. These tools and the model space were both proposed by Green (1992a). The arguments which support this coupling are biased toward analysis rather than empiricism because many of the tools which Green has defined have either not been fully implemented or have not been tested sufficiently to provide such data.

Chapter Five is where the main theoretical results of the research are described and the work of the three previous chapters is tied together to yield a new class of systems for knowledge aided design. The chapter has five goals. (1) To indicate the range of applicability of DODEs and process based environments within the design activity space. This involves arguing that the range of both types of system is more limited than has been claimed. (2) To identify a niche in the design activity space between the regions which should be supported with DODEs and process based environments. This

critical region is, it will be argued, in need of support from an appropriate class of system. (3) To propose MODEs as a class of system which can offer support for this niche and which occupies a region between DODEs and process based environments in Green's model space. (4) To propose envisaged technologies with which to implement MODEs and to compare MODEs with their neighbours in Green's model space. (5) To analyse the range of applicability of MODEs as they are currently envisaged in search of their limitations and further research goals.

1.5.2 Part two: putting theory into practice

At this point in the thesis the theoretical claims have been introduced and it is time to turn to practical matters. These involve: (1) using the spaces and mappings developed in the theoretical work to demonstrate that a particular design practice is suited to support via a MODE, and (2) mappings between design activities and tools developed in the theoretical work to design and implement a system to demonstrate the viability of the concept of a MODE.

The first of these goals is addressed in Chapter Six and the second is addressed in Chapter Seven.

The first section of Chapter Six describes the triggers which led to the implementation of Patina as a MODE for parametric analysis. The most important of these in the context of the argument presented in the thesis is the need to present demonstrations of: (1) the utility of the design activity space, (2) the utility of the coupling between the design activity space and Green's modelling space, and (3) the class of MODEs which were introduced in Chapter Five.

A particular prescriptive design method, known as parametric analysis, is used as an example of a design practice that is located in the region of the design activity space that is to be supported by MODEs.

The core of the chapter begins with descriptions of the organisational contexts in which parametric analysis is likely and less likely to be used, the purpose of performing a parametric analysis, together with the kinds of knowledge which it has been shown to produce and the types of products and services it can survey. In preparation for designing a system to support the method its various stages are described in detail alongside the artefacts and other materials which can be used in the course of an analysis.

Finally, parametric analysis is located along the three dimensions of the design activity space, which is introduced in Chapter Two, and the stages in the method which might benefit from knowledge aided support are highlighted.

Chapter Seven describes the synthesis of components to create Patina, a high-fidelity, feasibility and functional analysis prototype of a MODE to support parametric analysis. The system ameliorates the effects of some wicked and complex elements of the method that were highlighted in Chapter Six. The design and implementation of the program are reported in three key sections of the chapter which describe: (1) Patina's development strategy, (2) the elements of the program's concept design, and (3) the implementation of this concept design as a detail design.

These sections describe how various tools from Green's space of technologies were adapted to fit the demands of parametric analysis referring to the mapping between the model space of design activities and the component space of knowledge aided design tools which was developed in Chapter Four.

Chapter Eight summarises the outcomes of the research and then discusses how they can be interpreted as contributions to knowledge in the area of knowledge aided design. It also suggests relevant avenues for further work and makes some recommendations as to how the research presented here could be used by practitioners of knowledge aided design.

1.6 Contributions to knowledge

Although the thesis is entitled *Patina: a Method Oriented Design Environment for Parametric Analysis*, a good deal of analysis is necessary to reach the point where it is possible to: (1) locate a niche in the design activity space that stands in need of support, (2) propose the class of MODEs as a means to provide that support, and (3) derive the basic design of a MODE by using a mapping between the design activity space and tools in Green's model space. It is intended that the contribution to knowledge of this analytical work is at least as important as that of the design and implementation of Patina itself. To reflect this intention the thesis is implicitly structured into two parts. In the theoretical first part the problem leading to the definition of MODEs is identified and MODEs are proposed as one step toward its broad resolution. In the more concrete second part an example of a MODE is designed and implemented.

This structure begs the question of where the contributions to knowledge are to be found:

- The majority of new analytical and theoretical knowledge, which is based on the historical and technical material discussed in Chapters Two, Three and Four, is presented and tied together in Chapter Five, which proposes the new class of systems in light of the analysis.

- Chapter Six demonstrates the utility of the models and mappings that are developed in the first half of the thesis. This is done by analysing parametric analysis for areas that could be supported and showing that a system to support it will correspond to the concept of a MODE. This result is predictable using the models and the motivation for the proposal of MODEs.
- In Chapter Seven the feasibility of implementing a MODE is demonstrated by implementing such a system according to the principles that emerged during the proposal of this class of system in Chapter Five.
- The conclusion summarises and assesses both the analytical and concrete parts of the work.

A second important concern is the type of knowledge that is contributed to the field of knowledge aided design. Since the primary data in question were the theories and working practices of developers of knowledge aided design systems, design historians, and design researchers and since the work required interpretation, transfer and invention, the type of knowledge that was generated was qualitative rather than quantitative. More specifically the work generates the contributions to theory discussed above, illustrates their value in the design of a system and introduces a tool for data visualisation. Patina itself can be seen as an empirical demonstration that the theory that has been generated can lead to a practical outcome. To clarify this point, it should be noted that Patina's initial evaluation has been an informal usability evaluation. For this reason that evaluation is very briefly summarised here. For this reason no claims are made that Patina has yet contributed a set of quantifiable results. Rather its contribution is in the form of a demonstration of concept that should enable the class of MODEs to be visualised through a working prototype.

1.7 Summary

This introductory chapter has:

- provided an overview of the programme of research reported in the thesis
- presented the hypotheses to be addressed and discussed the methodology and work plan that were followed in order to address them

- described the structure of the thesis and shown where each hypothesis is addressed
- discussed the type of knowledge that the work reported here contributes to the subject area of knowledge aided design.

Chapter Two

Literature review: histories of design, design research and knowledge aided design

2.0 Overview

In this chapter the histories of design, design research and knowledge aided design are examined in order to:

- 1 introduce the interdisciplinary material that will be referred to in the core of the thesis
- 2 yield descriptions of the first and second generation methods which will provide much of the knowledge for MODEs and for the system to be presented in the later chapters
- 3 suggest that systems for knowledge aided design have tended to support previously formulated problems rather than unformulated problems because their developers have relied on an incomplete model of design
- 4 describe a range of alternatives to those found in Green's cluster of knowledge aided design systems, including domain oriented design environments, process based design environments and environments to support creative design.

The chapter contains three main sections, which discuss the histories of design, design research and knowledge aided design. It also contains a section which turns to the future and discusses the need for collaboration between these design researchers and developers of systems for knowledge aided design. A concluding summary highlights the issues raised here that will be examined in later chapters.

2.1 A history of design from the Classical period to the end of the Second World War

The first part of this history of design traces design from classical societies to the end of the second world war. It begins by contrasting the roles that design plays in tribal societies with those it has come to play in the industrialised West. It then investigates the role that design thinking played in Classical philosophical thought. The main body of the

history describes the evolution of Western design from its craft-based origins in the Mediaeval period to an ever more systematic activity throughout the Renaissance and the Industrial Revolution.

Archaeological and anthropological evidence shows that design has been performed as a craft-based activity in the context of human tribal societies throughout history. However, as Barley observes: "...until comparatively recently, most societies lived in a world where there was a finite number of objects and their form was relatively stable. In such worlds the introduction of a new object could entail a social revolution." (Barley, 1992, p 1).

Treating design as an activity that routinely produces innovative products, which equally routinely entail the kind of social revolutions brought about by the American gas station (Jakle and Sculle, 1994) is a relatively recent human activity that has deep roots in the evolution of Western capitalism and its constant requirement for growth (c.f., Karatani, 1995). From a Western-centric perspective, then, it is not surprising that design has been described as an ancient human activity that has recently evolved to become a modern profession (Ghose, 1995). The way in which this evolution has allowed design to move from being a craft-based activity to the systematic domain of professional engineers and designers (Pacy, 1992) is the subject of this section.

The Western tradition of the formalising of the study of design began with the rhetorical interest which Greek philosophers took in the subject (c.f., de Camp, 1963; Landels, 1978; Hodges, 1992). For example, design and technology were subjects of debate for Classical philosophers, such as Aristotle and Plato, who discussed *techne* as a form of applicable scientific knowledge. However, the interest of these philosophers was largely confined to considering the role that *techne* played as a type of knowledge in their epistemological, ethical and political theories. They were not as interested in studying *techne* as an activity in its own right for pragmatic purposes (Mitcham, 1994).

It is problematic for modern commentators to interpret the intentions of the ancient Greek philosophers. However, the following citations from design literature illustrate that Aristotle and Plato are regarded as precursors of some of the products that will be of concern later in the thesis and of the interplay between design and social planning. Mitchell argues that some moral reasoning in Aristotle's *Poetics* led him to envisage a "species of artificially intelligent tools" (1995, p 144) that would replace human slaves. In retrospect such tools could be seen as inspiration for the kind of knowledge based systems for design whose history will be introduced in section 2.3 (c.f., Boden, 1996). Turning to Plato's *Republic*, Onians (1991) argues that Plato's conceptions of design and manufacture in Classical Athens influenced his own design for a state that was to be run on the lines of a master craftsman running a large shop. This could be interpreted as an

early influence of the practice of design on the theory social planning. The Greek philosophers, then, began to conceptualise design and manufacturing skill as a kind of human knowledge that could be discussed in its own right.

The Middle Ages left much evidence of innovative products of design activity (c.f., Hall, 1984), particularly in architecture but also in areas such as woodwork, metalwork, stained glass, stone carving, and embroidery (Gloag, 1947). The art of weapon making was of such practical importance that its practice and research caused citizens of this era to overcome their Mediaeval suspicions of technology. However, in the main, design remained a craft-based activity and there were relatively few innovations in the design process itself. As Gloag observed: “these men were slowly establishing a tradition of design, broadening and deepening skills that had been handed on generation after generation.” (Gloag, 1947, p 7).

One problem that faced Mediaeval designers who wanted to engage in large scale projects was that illiteracy was widespread amongst their work force. This was compounded by limitations on the ways that had been invented to communicate using pictorial representations. Subsequent history shows how during the Renaissance the adoption of conceptual tools such as perspective and two-dimensional views of three-dimensional objects helped designers to improve on the impressive architectural techniques of Classical and Mediaeval societies. Booker (1963) and Ferguson (1992) present histories of engineering and architectural drafting which show how this activity gradually evolved from an ornate illustrative tradition toward objective plans with standardised systems of interpretation. These systems offered a medium for detailed communication between designers, manufacturers and their clients. With such means of standardised communication and envisualisation, designers could work in teams to produce more complicated structures in new materials. One economic and military spur for such development came from the invention of the cannon, which replaced the dominant technology of the siege engine, heralding a pressing need to invent and build new architectural structures (Ferguson, 1992).

Gloag (1942) argues that some of the architects and designers of the Renaissance period, including Wren and Chippendale, applied these technologies in ways which were less reminiscent of craft-based Mediaeval techniques and more reminiscent of industrial design and manufacture in the first half of the Twentieth Century. He also demonstrates that for a number of social, economic and technological reasons traditional English design began to be internationalised during this period (Gloag, 1947). So the Renaissance marks a transitional period where the craft-based traditions of the Mediaeval period were still in force but they had begun to be superseded by more modern approaches to design (Gloag, 1942).

The methods of working and their associated technologies which had begun to be discovered during the Renaissance fuelled the Industrial Revolution. In this period much effort was invested in the study and teaching of manufacturing-, marketing- and sales-expertise (Owen, 1991). Between the height of the Industrial Revolution and the beginnings of the post-modern age many changes affected the status and working conditions of designers. The profession of design also became increasingly regulated (Pacey, 1992). For example, the professional body of British Architects changed status from a club in 1791 to a chartered society in 1837 (Lawson, 1980, 1990). Designers also became increasingly divorced from the process of manufacture. Their familiarity with the day to day running of an efficient shop, which had so impressed Plato, was often replaced by education in abstract theory. This process was often reflected by physical relocation from the workshop to a specialised design office. Designers working in such offices needed to be equipped with ever more formalised systems for externalising design knowledge. This continued the process that had begun in the Renaissance and meant that their designs could be communicated to artisans or passed to automated manufacturing facilities (Ferguson, 1992). Dissent against this outcome of the Industrial Revolution became apparent in the writings of Ruskin and Morris. These writers favoured a return to craft-based techniques. However economic forces dictated that even Morris was eventually forced to adopt the methods of industrial design and manufacture which he had argued against (Open University Arts Foundation Course Team, 1974).

The divorce of design expertise from a particular tradition can be illustrated by the thoughts of Thomas Edison. The prime technological innovation of Edison's era was not a particular product. Rather it was the way of making products in a highly rational and designed fashion, which Biggs (1995) calls the 'engineered factory'. Edison expressed the view that the activities of design and automated manufacture could be organised in similar ways (Mitcham, 1994). He described his ideal workplace as an invention factory. In such a factory products would be invented to order just as diverse products were made to order in the engineered factory. This idea predicted the kind of technological products of knowledge based and knowledge aided systems for design which will be described in section 2.3.

The invention of Edison's invention factory remained an elusive goal during the later half of the Nineteenth Century (c.f., Rolt, 1974) and the first half of the Twentieth Century. Despite this fact the notion of the desirability of the invention factory gained general currency. Mitcham (1994, p 218), for example, underscores how important this idea seemed by quoting Whitehead: "The greatest invention of the nineteenth century was the invention of the method of invention. In order to understand our epoch, we can neglect all the details of change, such as railways, telegraphs, radios, spinning machines

and synthetic dyes. We must concentrate on the method itself; that is the real novelty, which has broken up the foundations of the old civilisation.” (Whitehead, 1925, p 95). This quote illustrates how routine invention had become in the first half of the twentieth century. It also demonstrates how inhabitants of the modern world had already appreciated how important the acceleration of invention had been in shaping that world.

By the close of the Second World War, design expertise was increasingly viewed as a form of capital which could be invested in and researched. As the next section will argue, Western governments began to feel that investing in such capital might produce economic and military gains which would lead to cultural and military dominance (Maguire, 1991).

2.2 Post-war design research in Great Britain and the West

This second part of our history of design traces design research from the close of the Second World War to the later 1990s. This section discusses concepts such as first and second generation design methods and participatory design which will contribute to the analyses of design and knowledge aided design presented in the first half of the thesis.

The close of the Second World War is where the history of design research begins in earnest because, for a mixture of social, political and economic reasons, researchers from the humanities began to receive ample funding to study design and technology during the subsequent economic boom. Consequently, design became an attractive domain for researchers from fields such as anthropology, psychology, sociology and aesthetics, especially in the United States, West Germany, Switzerland, Great Britain and Italy.

As Heims (1992) observes, the mid-century was a time when the social sciences began to receive funding generated by this post-war economic prosperity which rivalled that of the natural sciences. Much of this funding was bestowed by political forces with considerable interest in gaining or maintaining technological dominance in consumer and military culture. More controversially, as Heims also argues, there is another reason why social scientists, especially in the United States, might have chosen to study practical matters, such as design and technology. Since these Western governments saw their societies as being in marked opposition to the Soviet Union, where a social experiment was being run on a grand scale, the study of societies and new ways to organise them was tacitly discouraged by the funding agencies of these governments. For this reason it is arguable that it might be more appropriate to refer to the point in history at which the modern design movement began not as the end of the Second World War but rather as the onset of the Cold War.

In this way design research became established in the post-war era as a respectable way for sociologists and others to appear to pursue quantifiable research while in fact applying softer methods from the qualitative sciences. As this history will show, in Great Britain and the United States of America this particular generation of design research tended to produce rather systematic, almost algorithmic, design methods—Ferguson (1992), for example, documents how the low academic status afforded to subjects such as the study of the role of drawing in design accounts for some of the push toward algorithmic methods borrowed from the natural sciences. In contrast, however, a small minority of accepted methods, such as Gordon's Synectics (1961) relied on less rational and operations-research flavoured techniques such as a communal exploration of a group of designers' wildest fantasies about what it would be like to be an object they were designing.

The United States and Britain were far from the only countries or societies where design research flourished in the post-war era. Moreover, each of these countries and societies tended to explore an individual direction. For example:

- Scandinavian designers laid the foundations for what would become known as participatory design (c.f., Wooley, 1995).
- Although Japan began the post-war era as a country "in which there were no designers" (Hirano, 1995, p 219), the Japanese government consciously forged alliances between public institutions and the private sector to establish a society in which a "heightened corporate, commercial and public awareness of design now exists" (*Op Cite*).
- In Germany a programme of research into formal design methodology was pursued that exhibited many parallels with the Anglo-American tradition. Unfortunately, as Blessing argues (1994), their contribution was largely ignored by Anglo-American historians of design.

For a mixture of ideological and economic reasons the work of some countries and societies, such as the Soviet Union and Asia, was conspicuously absent from citation in the Anglo-American design research movement (Margolin, 1989; Ghose, 1995). However this should not be taken to suggest that research was not taking place: these cases show that the histories of post-war design research and design culture and the disciplines which have investigated them are manifold. This section focuses on the

history of British design research movement for the sake of brevity and because it produced ideas that are especially relevant to the arguments presented later in the thesis.

The defining characteristic of British design research in this century has been the advocacy and study of the move from craft-based design techniques toward structured industrial design (Pye, 1964, 1978; Jones 1970, 1992). Jones provides an interesting illustration of the evolved nature of craft-based design during a commentary on some historical observations of a Wheelwright's shop which were made by Sturt in 1923 (Jones, 1970, 1992; Sturt 1923). Jones quotes Sturt's attempts to discover the real reason why a particular type of wagon wheel is dish shaped. Eventually, after discussing several possible reasons why this might be, Jones notes that there are an unknown number of possible reasons why the wagon wheel is dish shaped. He asserts that: "For us it is enough to notice that each part of the wagon is shaped not for one reason but for many, and that there is a delicate adjustment throughout the whole to get the best out of each bit. We should also notice that the craftsman who reproduces and modifies the form does not know all the reasons for what is done, he knows only the way to do it...Craftsmen do not, and often cannot, draw their works and neither can they give adequate reasons for the decisions they take." (Jones, 1970, 1992, p 18–19).

Despite the fact that industrial design had been gathering momentum in Britain since the Renaissance such craft-based activities were still practised at the beginning of the post-war era. Indeed Gloag's *The English Tradition of Design* (1947) describes a post-war Britain in which genuine crafts-people continued to work within a culture that also included such exponents of large scale industrial design as The London Underground group. Gloag himself was much more than a historian of design. He advocated movement away from such craft-based design towards an industrial design that would not lose the advantages of a craft-based approach nor the British character of the resulting products. The theme of such work can be traced through to the design research movement—especially in the writings of Pye (1964, 1978) and Jones (1970, 1991, 1992).

Post-war British design research also has roots in the establishment of government organisations and educational institutions. These organisations included: The *Design and Industries Association*, which was instituted in 1915, the *British Institute of Industrial Art*, instituted in 1920, and the *Council for Industrial Design*, which was instituted in 1944 (Gloag, 1947). Later post-war educational institutions such as the red brick universities, the polytechnics and the Open University contributed to the dissemination of a design culture that came to supersede and enhance the traditional engineering apprenticeship (Potter, 1980).

The British programme of enquiry into rational and systematic design methods gathered force in the 1960s, with the emergence of work typified by Archer (1965), Alexander (1963) and Jones (1963).

In 1962, the Design Research Society organised the first *Conference on Design Methods* (Jones and Thornley, 1963). As its name implies, the conference introduced the first generation of design researchers to one another and publicised the first generation of prescriptive design methods. Jones provides us with the flavour of the content, purpose and methodology of this movement by describing his own approach in this era: "... I began trying for what I'd now call a human functionalism i.e., making design thought public so that [design methods] are not limited to the experience of the designer and can incorporate scientific knowledge of human abilities and limitations" (Jones, 1977, p 50). In contrast, however, the conference also featured researchers who argued that creative design could not be managed by entirely systematic means. For example Page's review of the papers at the conference asserted: "I think that in science, on the whole, we have the direction of creative work much better planned than in engineering. Anyone who directs programmes of research in science knows that all you can do in practice is take on bright young men, give them the best tools that you can afford, and let them get on with it. They will not do precisely what you want them to do or even what they set out to do." (Page, 1963, p 206).

It is important to note that Jones' human functionalist methods did not attempt to replace human practice with entirely systematic or algorithmic methods. Instead they attempted to structure native human abilities so that designers could learn to manage their own activities as efficiently as possible and also to explain them to colleagues, managers and clients. This reflected a partiality toward organisation design that was widespread throughout this era (Owen, 1991). As Cross (1984) argues, these methods were erroneously criticised by certain designers and theorists for being too systematic and for ignoring the intuitive. While they encouraged designers to perform pre-defined activities within a formal framework, these methods also encouraged logical analysis and creative thought. However, there was some justified content to these criticisms of such procedural design methods. As Jones later observed, it was striking that each phase of a procedural design method seemed to begin with a phase of goal setting which was difficult to do, since it required intuition. Success in this phase was heavily dependent upon a designers personal history and training and upon how they perceived the context they were working in (Jones, 1977). So, if designers were not sufficiently well-trained or were not warned of this difficulty, there was a risk that first generation design methods would break-down in certain contexts of use. Moreover, similar problems were encountered when designers approached a design activity from different educational

backgrounds or cultures yet needed to communicate with one another when they followed a particular method.

Between the late 1960s and the early 1980s, design theorists such as Archer (1979), Broadbent and Ward (1969), Rittel and Webber (1973), Rittel *et al* (1972) and Jones (1977) began to recognise and report these problems. Consequently they began to doubt the merit of the first generation of design methods as solutions to the general problems of design for products that would be used by people. The causes for this move will be explored in the section in Chapter Three on the emergence of second generation design methods. What is important to note here is that during this period the design research movement began to re-explore the need to educate designers to understand problems prior to solving them. In particular it was felt to be important for designers to realise that they were culturally situated and that design is a process with social and political inputs and consequences.

In the 1970s, then, design education became an increasingly important issue to the design research community. As Archer commented in 1979: "Design methodology is alive and well, living in the bosom of its family: design history, design philosophy, design criticism, design epistemology, design modelling, design measurement, design management and design education." (Archer, 1979, p 18). Accordingly post-war educational institutions such as the Open University began to offer open access courses on design with contents that ranged from the applied (Cross, 1975, 1989) to the psychological and philosophical (Crickmay and Jones, 1972). This helped to establish an education context in which procedural design methods could be taught in a way that took account of their potential short-comings. Moreover, this generation of students could also learn from practical experiences of design that were recorded in journals such as *Design Studies*.

One area where the influence of second generation design thinking came to the fore was in architecture. In particular, the ideas of participatory design began to be imported in Britain and America from Scandinavian architectural traditions (Papenek, 1974; Alexander *et al*, 1975; Towers, 1995). These advocated involving the people who would live or work in a building within the team of people who would design it. This was one way to address the social and political problems of goal-setting and evaluation which were so problematic for first generation design methods. However, participatory design incurred visible expenses at the design stage which might otherwise have only become apparent during maintenance. This could cause problems for design professionals because these costs would affect the size of a bid for a project. A second problem was that the lay people who needed to be involved in a project might be professionals in their own right, with their own set of commitments. It could be difficult to ensure that such

professionals were awarded credit for their involvement in a design project that would contribute to their own professional standing (Alexander *et al*, 1975). Types of participants who came to be studied in the design literature included people from pre-industrial societies, women, consumers and children (Pacy, 1992). While participatory design continued to make inroads into architectural schools well into the 1980s, as Wooley (1995) notes, its progress in that decade was tempered by perceived financial realities.

The most individualistic response to the need for new ways to design in this era came from Jones (1991) who turned toward random generation as a way to stimulate creativity and to generate new products and ways of working. While that approach was stimulating it is best seen as an interesting tributary to design research. It was certainly not representative of the direction taken by a movement which was heading toward a more informed version of first generation of design methods.

In the later 1980s and early 1990s, Anglo American design theorists had started to produce mature textbooks on the systematic management of the design process. There seemed to be a realisation that first generation procedural design methods had a valuable role to play in design education as well as in design practice, so long as they were tempered by realistic evaluations of the problems of context and culture. Such works also capitalised on the design culture in which undergraduate designers now found themselves. They concentrated on providing methods that could be applied to areas such as architecture, engineering and the service sector: *Successful Product Design—What to do and When* (Hollins and Pugh, 1990) reports upon product design; Lawson's *How Designers Think: the Design Process Demystified* (1990) is especially strong in the domain of architecture; Jones' *Design Methods: Seeds of Human Futures* (1992) updates his 1970 text to take account of interim developments, Pugh's *Total Design: Integrated Methods for Successful Product Engineering*, (1991) is devoted to engineering, and Hollins and Hollins' *Total Design: Managing the Design Process in the Service Sector*, (1991) applies structured methodologies to the service sector.

The influence of second generation thinking, especially the utility of case studies in design education, can be seen in works such as Petroski's *To Engineer is Human: The Role of Failure in Design* and *Invention by Design: How Engineers Get from Thought to Thing*, (1992, 1996).

If the later 1980s and early 1990s can be seen as a period in which design research consolidated around four decades of post-war achievement, then the contents of *Design Studies* in that decade also reveals that British, American and Australian design researchers increasingly began to explore the new territory of knowledge based design. Thus, in this period the British design research movement started to colonise

departments of computer science and artificial intelligence (Pugh, 1989) and vice versa (Beardon *et al*, 1995).

Some researchers and educators appreciated that there was not only a need for design technology with which to support designers working in the information age but that educational techniques would need to be adjusted to take account of them. For example Owen observed: “design students now operate in a post industrial age of information technology with shorter lead times for design; yet much of the syllabus of design education is devoted to methods that are limited to what could be done with calculators and pencil and paper.” (Owen, 1991, p 28).

Politically, within the larger environment of the European Community, British design competitiveness began to focus on Japan, rather than the United States. This led to a funding climate in the 1980s and early 1990s that has been called techno-orientalism (Morley and Robbins, 1992; Barry, 1996). During this period the Japanese government were investing heavily in publicly-funded research programmes in the subject areas of information technology, robotics and artificial intelligence, such as the fifth generation programme (Moravec, 1988). The European Community responded with centrally-funded European research programmes, such as ESPRIT, in the subject areas of information technology, artificial intelligence and network technologies. These programmes led to a climate in which the development of the Internet and establishment of the World Wide Web, with its opportunities for collaborative design across borders and time zones, became possible as well as politically desirable (Barry, 1996).

These trends bring us to a point where it is appropriate to review the history of artificial intelligence—a technology that has evolved in parallel with the events recounted in this section.

2.3 Artificial intelligence, knowledge based design and knowledge aided design

This concluding part of our history of design traces artificial intelligence from the close of the Second World War to the emergence of systems for design during the 1980s and 1990s. It discusses the component technologies of knowledge based systems and discusses the theories of mind which underpinned them. As the history enters the 1980s, it focuses more closely upon knowledge based and knowledge aided design and shows how particular views of design came to influence the functionality of the first wave of such systems. The section concludes with a review of alternative types of system that have been developed or proposed in the 1990s.

Artificial intelligence, which can be described as the study of how to make machines act more like people, has roots in a wide range of disciplines. These include cognitive

psychology, philosophy, robotics and operations research. However the most important influence on artificial intelligence has always been the computer—a technology which has served as both an essential foundation for its view of mind (Putnam, 1975) and as a chemist's flask (Minsky, 1986) for practical researchers in this subject area. Since computers and computer science are one of the major contributions which the academic, and industrial establishments of United States of America have made to the technology and culture of the second half of the twentieth century it is not surprising that much of the research that will be described in this section has taken place in the United States. This has had implications for the field since it became broadly based on views of the mind that were dominant within the post-war American scientific establishment (c.f., Agre, 1997).

During the 1940s and 1950s some American scientists and engineers who had proven themselves within the area of weapons and computer research in the Second World War began to take an interest in modelling or replicating human intelligence via their new computational technologies. In particular John von Neumann took an interest in this field (Macrae, 1992) which culminated in the posthumous publication of *The Computer and the Brain* (von Neumann, 1958). It is arguable that the role which von Neumann had played in the development of both the atomic bomb and the computer brought a legitimacy to research in this new area—especially for the United States Department of Defence (Norberg and O'Neil, 1995), which has continued to fund speculative and strategic research throughout the post-war era (de Landa, 1991).

In the period between the late 1950s and the early 1990s this speculative and strategic research had some problems maintaining its early promise but it eventually provided key technologies for more applied research in the subject areas of knowledge based systems and expert systems—Stier (1992) notes that what may have been the first expert system (built in 1958) was applied to the domain of designing generators and motors.

This work was based on a blend of cognitive psychology and operations research which can be traced to Herbert Simon's *The Sciences of the Artificial* (Simon, 1969). Herbert Simon's work integrated cognitive psychology and operations research and modelled human activities as heuristic goal-directed search (c.f., Boden, 1988). By the 1970s and 1980s, it had underpinned the development of programs (Newell and Simon, 1972) that formulated answers to specific questions in the form of plans. These plans were generated by heuristically searching for potential paths from well defined and formally represented start states to similar goal states (see papers collected in Allen, Hendler and Tate, 1990). This approach was applied to creating programs that modelled intelligent activity in a variety of domains, such as Chess, and logical theorem proving and design.

During the early 1970s, however, artificial intelligence met with a demonstrable lack of success in applying these ideas to real world problems. This meant that artificial intelligence research was taking place in what Winston called its funding dark ages (Winston and Pendergrass, 1984). However, during the later 1970s and early 1980s a number of commercially successful programs emerged. These succeeded by treating commercially viable areas, such as medical diagnosis, as self contained micro worlds. Such self contained worlds that could be modelled on a computer that could now represent a state of affairs and reason about it.

The commercial success of these programs promoted a sea-change in funding. Industrialists began to perceive artificial intelligence as an area worthy of profit-motivated investment (Winston and Pendergrass, 1984). As a result the 1980s witnessed a well funded programme of development and implementation of expert systems and knowledge based systems that were applied to very specific domains.

Since the fact that such programs were set in domain specific micro worlds was integral to their success, much effort was paid during this period to discovering ways to represent knowledge about such domains. It was felt that this would allow modularity between knowledge and inference within a program so that standard goal-directed search-engines could be re-used to solve problems within different domains (Brachman, 1985).

The other main theme of research in this era was devoted to discovering ways to elicit expertise from experts which could be used as the vital source of knowledge for such systems. As Hayes-Roth *et al* put it: "knowledge acquisition is the bottleneck in the construction of expert systems" (Hayes-Roth *et al*, 1983, p 1).

During the 1980s a combination of academics and industrialists began to identify engineering design as a promising domain for their newly matured knowledge based and expert systems technologies. The kinds of systems that were developed in that era are documented by texts such as Rycheners *Expert Systems for Engineering Design* (Rychener, 1988) amongst others (e.g., Pham, 1988; Winstanley, 1990). Unfortunately these systems generally suffered from a major limitation. Researchers tended to develop very similar programs that mostly tackled well-defined optimisation problems which were erroneously presented to users as design problems. The following quote from researchers in artificial intelligence and design at MIT indicates how design was perceived by such developers: "(design is) the process of specifying a description of an artefact that satisfies constraints arising from a number of sources by using diverse sources of knowledge" (Sriram *et al*, 1989, p 79).

This definition is useful here because it reveals an underlying assumption on the part of such developers which, it will be argued, has contributed toward the narrow range of

utility of the majority of systems for knowledge aided design. If it is axiomatic that design involves satisfying a set of objective or implicit constraints, then it should also be axiomatic that constraints do not simply arise; rather they must be actively sought in activities that are fundamental to design (Rittel and Webber, 1973; Buchanan, 1995; see also papers in Cross, 1984, and the discussion in Chapters Three and Four of this thesis). To assume otherwise leads to models in which constraints are either externally imposed or they are so routine that they contradict the notion of design as an innovative activity. In the first case the model of design is incomplete and in the second it is simply a misnomer.

So this definition indicates a bias on the part of developers of knowledge aided design systems toward the production of a solution that satisfies a set of constraints and away from the identification, formalisation and evaluation of these very constraints. This mischaracterisation of optimisation problems as design problems ignored many observations that had already been made in the design research literature. Specifically, it ignored two key observations of the second generation of design research. First, design is a creative activity which usually has social and economic inputs and consequences. Second, identifying these inputs and evaluating these consequences is very much part of a design activity. For example, in Rychener's review of research in expert systems for design these systems are characterised such that their inputs assume some givens which include "the specification of the desired object or system, giving its features, functions, constraints, budgets etc." (Rychener, 1988, p 12). In such systems the design part of the problem was being addressed by two sets of people rather than the system. First, the system developers would be responsible for how knowledge about a particular problem was represented. Second, the user would be responsible for how the start and goal states for the system were characterised.

A major contributor for this oversight seems to have been that many of these systems were grounded on the reification of a single American-centric strand of the design research literature. For example, Rychener's literature review shows a bias toward the methodologies of cognitive psychology, which have been seen to be a root of expert systems technology, over those of design research. But more importantly it views Simon's *The Sciences of the Artificial* as the "root" of engineering design research and all but ignores the traditions of social psychology and ethnomethodology (Simon, 1969). This was short sighted, because as the previous sections have indicated, design research had already produced a rich literature that included studies that complemented cognitive psychology with the outputs of social sciences including ethnomethodology.

This type of characterisation can be found in more recent models of design, such as that of Chandrasekaran (1992). This model characterised design processes using the

problem solving paradigm inherited from Simon and his contemporaries. The problem solving paradigm led to a design methodology in knowledge based systems which involves eliciting knowledge from domain experts and then transforming this knowledge into an automated or semi-automated program to replace or support these experts or their junior colleagues. Since the domain experts in question are designers working in a particular type of product area it is implicit to this methodology that the resulting systems will be intended for use when designing similar products. While this paradigm and its resulting methodology has given rise to many knowledge based systems that are well suited to solving problems there is an inherent limitation to applying it to design problems. This is because the strengths of this standard problem solving paradigm did not include a portrayal of the activities of problem definition and evaluation which are characteristic of the initial and later stages of each component of structured design. So it is unsurprising that developing a system by working from such a task analysis through a standard development methodology would tend to lead to a product that failed to support designers through the definition and evaluation stages of the design process.

Goel and Pirolli took on the baton of developing a design problem space from such researchers as Akin (1979), Brown and Chandrasekaren (1989), Jefferies et al (1981) and Newell and Simon (1972). On the basis of such research—which was mainly centred on software design—and design theorists such as Archer (1969) and Rittel and Webber (1973), they hypothesised that design exhibited “major invariants across design problem solving situations” (Goel and Pirolli, 1992, p 399) and hence advocated creating characterisations of design that cut across disciplinary boundaries to create a generic design problem space. They initially sought to categorise design from a cognitive perspective, arguing that it is “fundamentally mental, representational, and a signature of human intelligence” (Goel and Pirolli, 1992, p 396), but began to introduce some design theory into their model—which largely consisted of ideas that paraphrased Rittel and Webbers work. They concluded that “the notion of a design problem space is an interesting and explanatory theoretical construct worthy of further study” (Goel and Pirolli, 1992, p 427) in the field of cognitive science. This was a brave move that re-invented design research in areas such as second generation design methods that will be reviewed in Chapters Three and Four. However, their contribution was aimed at cognitive psychology rather than system design/theory and there still appears to be a need for a compact representation of the space of design activities whose key ingredients include various types of recognised design methodology and the notion that design problems often need to be continuously set and evaluated. This need for a representation of the space of design activities constitutes the research goal to be addressed in Chapter Four.

If the thinking behind the era of knowledge based systems and expert systems for design echoed that of Edison's invention factory, then developments in the first half of the 1990s have echoed those of the second generation of design researchers. These ideas have found expression in the texts of Coyne *et al* (1990), Fischer and co-workers (1992, 1994a, 1994b, 1996), Silverman (1992) and Green (1992, 1992a) which were published in the first half of the 1990s.

The 1990s opened with the publication of an Australian text by Coyne *et al* entitled *Knowledge-Based Design Systems* (1990). The publication of this text marks an important point in the history of knowledge aided design for two reasons. First, it summarised the work to date in the development of knowledge based systems and expert systems for design. Moreover it did so in the format of a course for an undergraduate level audience. This illustrated that members of the artificial intelligence community had begun to see knowledge based systems and expert systems for design as a mature subject area in its own right. Second, the work opened with a chapter that began the task of setting the development of knowledge based systems and expert systems for design in the context of design research. Although this was an important move the rest of the chapters were *still* steeped in the lore of existing technologies for building knowledge based systems and expert systems.

So, if *Knowledge-Based Design Systems* addressed the lack of historical context that can be found in works by authors such as Rychener (1988), then it also continued to fall victim to the temptation to apply mature artificial intelligence technologies to a broad range of design activities. However, the introductory chapter *did* introduce its readers to a fine-grained model of design and it *did* prepare the way for researchers such as Fischer, Silverman and Green who started to develop new technologies and hybrids of technologies in response to that model. From the perspective of the later 1990s Coyne *et al*'s text can be seen as a sign post to the transition from knowledge based design to knowledge aided design.

Gerhardt Fischer entered the world of expert systems and knowledge based systems for design from the software engineering community (Fischer, 1992, 1994a, 1994b, *et al* 1996). Fischer treated programmers as designers and cited design researchers, particularly Rittel and Webber (1973), to argue against the idea of trying to automate the writing of code. Fischer underscored the strong interrelationship between problem setting and problem solving and introduced many software engineers and members of the knowledge based systems community to the ideas that (i) one cannot gather information meaningfully unless one has understood the problem, yet one cannot understand the problem without information about it; and (ii) professional practice has at least as much to do with setting a problem as with solving a problem.

Fischer noticed that traditional knowledge based systems and expert systems did not address these parts of the design process and attempted to create software that absorbed some of the difficulties associated with them. Fischer invented the concept of the Domain Oriented Design Environment (DODE), a type of system centred on an iconic user-interface representation of the item being designed and on a technology known as critiquing.

Critiquing is a technology which has become associated with the work of Barry Silverman and in particular with his theories presented in *Critiquing Human Error* (1992). Silverman describes the critics as small knowledge based systems whose role is to help “people to notice, criticise, and reduce human fallibility” (*Op Cite* p 3). Essentially, Critics shadow human decision-making and attempt to bring evidence of faulty decision making to a human’s attention. Silverman observed that critics appear most useful “when inserted into pre-existing automated environments for users at competent/proficient skill levels attempting semi-structured tasks” (*Op Cite* p 38). As the previous sections have shown Silverman’s description of the skill sets that Critics enhance and the activities for which Critics appear useful is very close to a description of the skill set and activities of a professional designer. Silverman has produced empirical evidence to support this claim by developing and evaluating a wide range of critiquing systems for designing items such as military documents and mine-sweepers. As Chapter Three will illustrate, these are very similar to Fischer’s concept of a DODE.

The systems developed by Fischer and Silverman began to break away from some of the traditions of expert systems and knowledge based systems for design. In doing so they illustrated a trend toward new ways of interacting with computers and design problems that was documented in detail by Marc Green (1992, 1992a).

Green argued that the previous generations of knowledge-intensive systems for design had tended to rely on the expert systems and knowledge based models. He suggested that this was short sighted because there was a large and unexplored space of different types of technologies and hybrids of technologies that could be developed.

Green differentiated these knowledge-intensive systems from Computer Aided Drafting systems and argued that it would be a good idea to think in terms of knowledge aided design rather than knowledge based design. He argued that this would not predispose developers to simply re-apply expert systems and knowledge based systems to each of the different types of activities which designers pursue. Green also described, categorised and predicted many different types of tools and technologies for knowledge aided design which have since come into widespread use within the subject area (a selection of these will be described in Chapter Four).

Later articles from the trade press, such as an overview of design management systems aimed at concurrent engineering (c.f., Hars, 1996), show that many of Green's predictions are fast becoming established commercial realities in the later 1990s. This helps to demonstrate that Green's text also played a political role in terms of securing funding for research in knowledge aided design. Green observed that designers working in the new economies of the Pacific Rim worked for a small fraction of the salary of a Western designer. He therefore argued that it was prudent for organisations based in the West to either out-source their design work or to investigate whether knowledge aided technologies could improve the productivity of their home-based designer. Much of Green's text advocates the desirability of involving a designer in the design process who is familiar with the product and market under consideration. Therefore Green's text can be read as suggesting it is worthwhile for Western agencies to invest in knowledge aided design because this would allow them to be able to afford to employ local designers who are familiar with the local culture.

So, by the mid 1990s, an ongoing rivalry for economic wealth and cultural dominance between various nations was set to fund a further round of applied design research and research into artificial intelligence.

The next significant event in the history of knowledge aided design was the move toward systems which strove to integrate significant amounts of prescriptive and/or descriptive design knowledge. These systems indicate that the goal of the subject area is gradually shifting away from creating autonomous domain-specific systems which are not especially grounded in the theory of design. The revised target is for more interactive and domain-independent systems which begin to integrate the outcomes of the history of design research.

Three examples of such research which will bring this history up to date are: Blessing's hypothetical PROcess-based SUport System (PROSUS) (1994, 1996); Gilleard and Lee's Construction Technology Identification System (CTIS) (1996); and Candy and Edmonds prescriptions for the future of Knowledge Support systems for creative design (1996).

Lucienne Blessing's work has furthered the migration of ideas from design research to the subject area of knowledge aided design systems (Stomph-Blessing, 1992; and the first chapter of Blessing, 1994). For example, her paper *Engineering Design and Artificial Intelligence: a Promising Marriage?* (presented at Cross *et al*'s 1992 workshop on *Research in Design Thinking*), significantly strengthened the bridges that had already been built by Coyne *et al*, Fischer and Green. In 1994, Blessing published a thesis entitled *A Process-Based Approach to Computer-Supported Engineering Design*.

This project brought the findings of the previous generations of design researchers to bear upon the specification of her PROSUS system.

PROSUS furthered the work of researchers who had applied the technologies of issue based information systems (see Fischer *et al*, 1991) to the design domain. PROSUS sought to support a Total Design-like (Hollins and Pugh, 1990; Hollins and Hollins, 1991; Pugh, 1991) range of activities by representing an ongoing project as a matrix of issues and activities which would engage designers throughout the life of the project. Such a design matrix would provide a “working area for the designer or design team and enable the system to suggest relevant knowledge, methods, and design histories.” (Blessing, 1994, p iv).

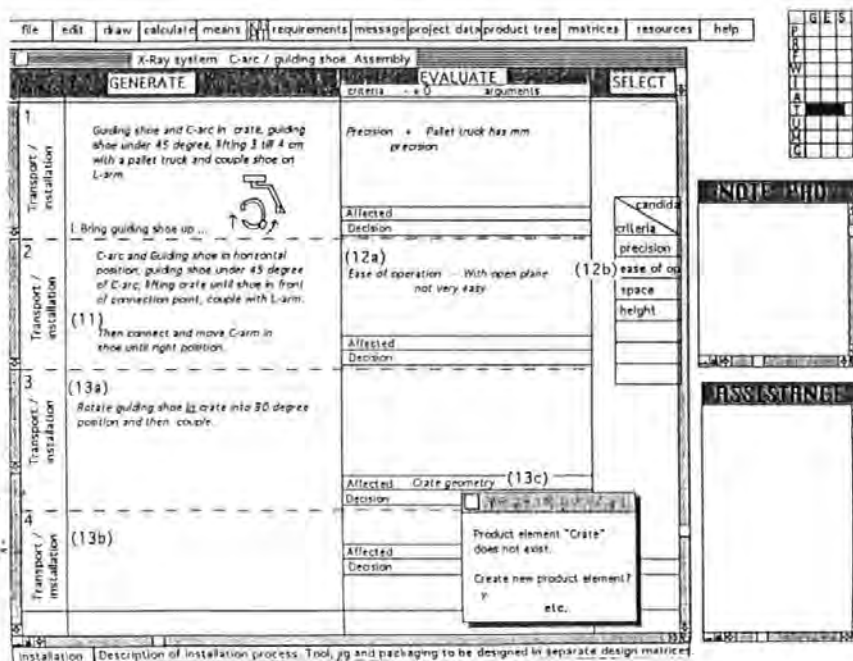


Figure 2.1: The proposed interface to Blessing's PROSUS system (from Blessing, 1994, p 282).

Although PROSUS remains a hypothetical system, whose existence is limited to a sequence of illustrated screen dumps of the user-interface (see figure 2.1), it was an important step forward for knowledge aided design for two reasons. First, its specification acknowledged the importance of supporting: a methodological design activity; the structured documentation of product data; the retrieval of project data for reuse as well as structured knowledge about methods, tools and design history; the provision of context sensitive advice, assistance and guidance; and conduits for communication to facilitate teamwork (Blessing, 1994, p iii).

Second, Blessing based PROSUS on a theoretical foundation that advocated a move away from product-based systems toward process-based systems. That is, an implemented version of Blessing's system would present its users with an interface that

represented the process of designing a product rather than an illustration of the evolving product itself—to clarify, illustrations of part of the product were embedded within Blessing diagrams of the systems interface (see Blessing, 1994, pp 273–301).

The work of Gilleard and Lee (1993, 1994, 1996) is perhaps less theoretically groundbreaking and ambitious than that of Blessing; yet the implementation of their CITIS system offers tantalising glimpses of a process-based future for knowledge aided design.

In 1993 Gilleard and Lee had successfully implemented BSE, a design tool for rule-based building services engineering in the domain of fire sprinkler layout design using a hypermedia authoring system (Gilleard and Lee, 1993). However, upon evaluation of this system with experienced designers, Gilleard and Lee discovered that their subjects both questioned the need for it to be “so prescriptive” (1996, p 43) and argued that the activity was more error prone, and therefore more iterative, than BSE’s design assumed. These comments lead Gilleard and Lee to recognise that the assumptions behind the design of BSE needed to be re-examined.

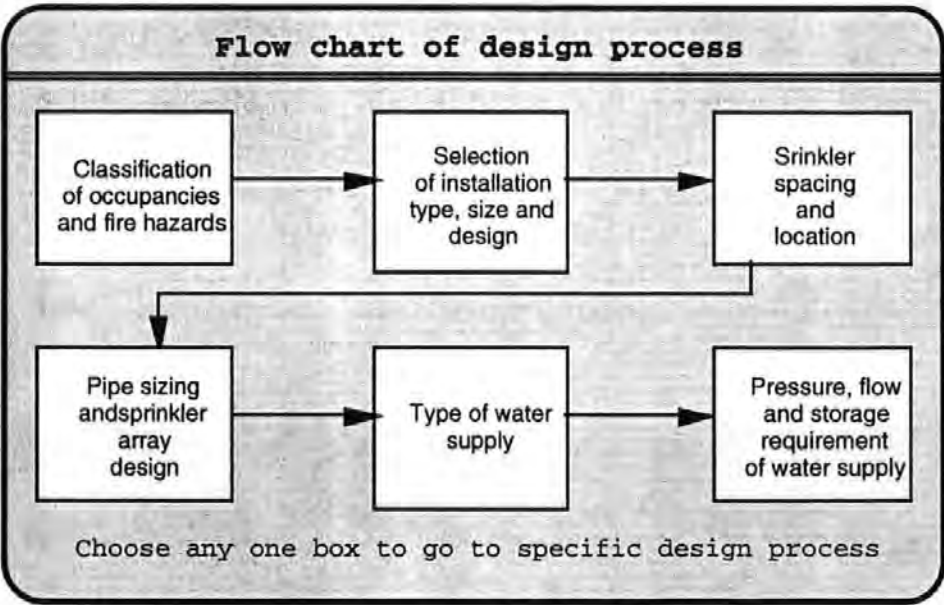


Figure 2.2: “Six design sub-processes defined as a main menu” (a card from a Hypercard system, re-drawn from Gilleard and Lee, 1996, p 48).

In 1996 Gilleard and Lee responded to the evaluation of BSE by presenting the design of CITIS, a system that recognised the semi-structured nature of the domain of fire sprinkler layout design. In CITIS the need for iterative navigation between sub-stages of fire sprinkler layout design was addressed by a main menu on a card representing a flow chart of links to cards supporting the main design sub-processes (see figure 2.2). These

cards each contained relevant and detailed information about a sub-process, including miniature expert systems.

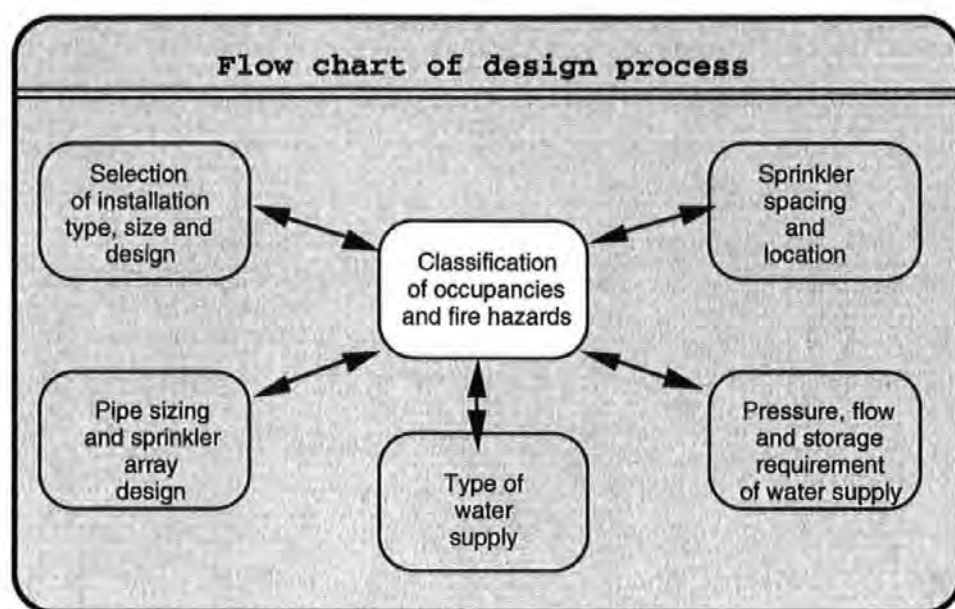


Figure 2.3: Modified main menu showing Classification of Fire Hazard as the focus of navigation (a card from a Hypercard system, re-drawn from Gilleard and Lee, 1996, p 48).

An important and undesired feature of this first attempt at designing this main menu, Gilleard and Lee argued, was that it implied an ordering of events. Unfortunately, with one exception, no such logical ordering either existed or was likely to be followed by designers (the implied ordering is heavily underscored by the unidirectional arrows). This exception was the need to declare a class of hazard early in the design process. Accordingly the menu card was altered (see figure 2.3) so as to highlight this central stage in the process and to show that each of the other stages could be visited and revisited from it at will (this is indicated by the bi-directional arrows to and from the central and ancillary stages).

Although Gilleard and Lee's CITIS system is highly domain specific it is groundbreaking for two reasons. First, it presents its users with a depiction of the work process as well as or in place of the work item. Second, it not only makes the iterative nature of design in this domain explicitly apparent but also helps designers to navigate through their design process with diagrams that show where they are, where they have been and where they could go next. CITIS, then, brings recognition of the fact that this is a semi-structured domain to supporting these design activities. A possible criticism of its user interface is that the buttons on the menu card representing the sub-stages of the design process do not provide an indication of the work that has been done or that remains to be done in these stages.

It is important to stress that CITIS is a successful occupant of a semi-structured and highly constrained niche. However its limitation from the perspective of the arguments presented later in the thesis is that it would be difficult to scale its architecture to address design activities that require more creative input or evaluation. This is because the expert systems at the core of CITIS still demand known entities with well-defined behaviours. Moreover, the design process which is illustrated in the interface is tightly coupled to existing practices in the domain. It is not clear that CITIS could be used in projects that step outside of the familiar boundaries of the artefacts and practices in its application domain.

The work of Candy and Edmonds (1996) explores how computers can be used to support the design of artefacts requiring more creativity on the part of designers. Like Blessing, Candy and Edmonds argue that both prescriptive and descriptive design knowledge could make important contributions to such support but warn “these [both] span a wide range of domains and have yet to be generalised across these domains” (*Op Cite* p 72). They realised the importance of observing the actions of creative designers and respecting their hunches when it came to the subject of whether they would use computer support tools for specific design activities or whether they would prefer to experience the familiar engagement they felt with traditional materials.

On the basis of their studies of creative designers and their own experience in knowledge based and knowledge aided design, Candy and Edmonds defined a set of issues which would need to be addressed by systems that were intended to support creative design:

- Creative designers need to be able to discover, originate and store new knowledge about unforeseen subjects within a support system as a design progresses.
- Creative designers need to be able to freely work upon and to freely integrate parallel solutions to a given problem during the design process (See also Lawson, 1993).
- Creative designers need to be able to share knowledge with one another and to identify and distinguish one another’s skills.
- Creative designers need to be able to formulate problems in a way that means they can be reformulated throughout the design process.

- Creative designers often employ idiosyncratic strategies: from an organisational perspective it would be desirable if these could be recorded and disseminated to other designers.

The kinds of systems which Candy and Edmonds sketch in their illustrations of systems that might help to achieve these goals are very unstructured and navigable, with windows featuring rich graphics, photography and video, as well as exclusively textual windows (see figure 2.4). Importantly, they integrate hyper links between the designers desktop and world outside the design studio.

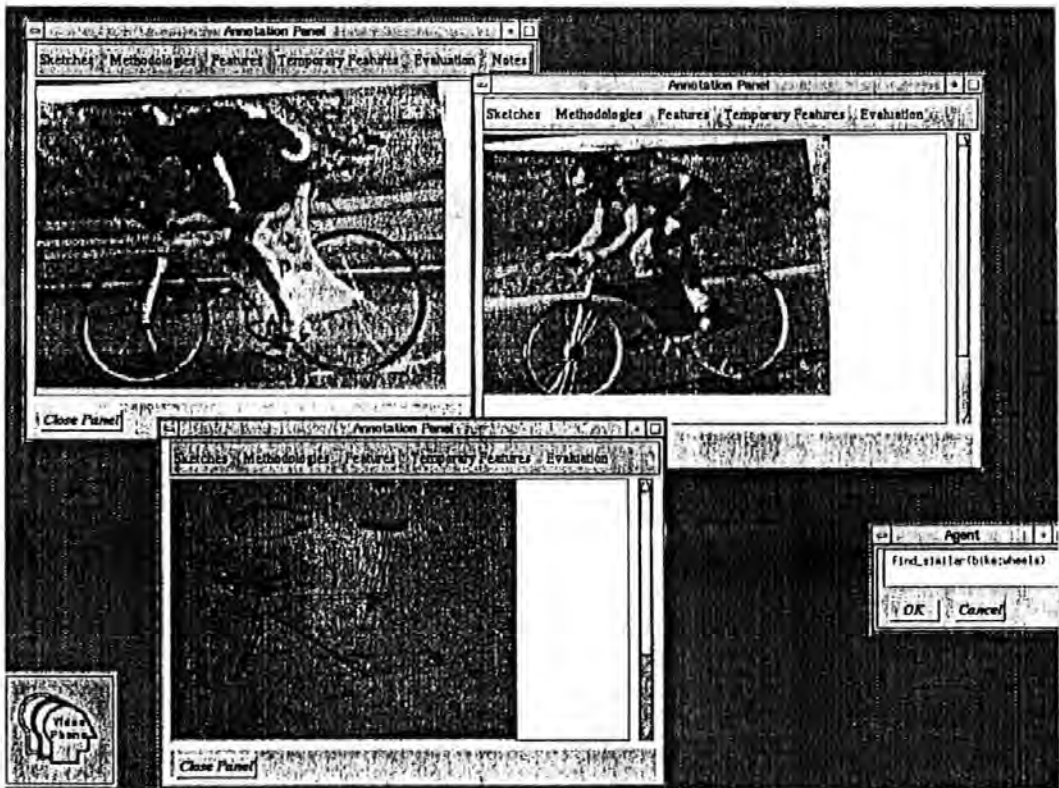


Figure 2.4: The proposed interface to Candy and Edmond's system to support creative design (from Candy and Edmonds, 1996, p 86).

Candy and Edmonds illustrations can be interpreted as showing that, in the opinion of two experts in this domain, single systems or paradigms are unlikely to service the needs of creative designers. One reason to support such a prediction is the observation that designers are so prone to adapting and subverting the intended use of such systems (Beardon *et al*, 1995) that discovering a fixed set of requirements would prove impossible. Moreover, since the costs of developing very large software packages are as prohibitive as the number of bugs encountered in large scale programming (Landauer, 1995), there are economic reasons to predict that large scale programs such as PROSUS,

which might support the entire design process, are less likely to be developed than small scale programs.

Instead a more cost effective future for knowledge aided design might be based on developing interacting and stand alone programs closer in spirit to CITIS. Such programs might address individual and tractable sized design activities using appropriate mixtures of knowledge and technology. It is an open question whether such programs will be tightly coupled to particular domains. From an economic view-point it might be cost effective to develop systems with a broad domain of applicability so that development costs could be recovered from a larger potential customer base. At present the question of whether future development of knowledge aided design systems will take place in academic or commercial environments is also open. By the prior argument, the answer to this question seems quite likely to affect whether these systems are tightly or loosely coupled to particular domains and methods.

Wherever such development takes place, creating knowledge aided design systems and evolving conventions for communication between them appears to be a research programme which will require its share of technological advances. More importantly, however, it will probably also require system designers to collaborate with students of design, working designers, teachers of design and with design researchers. The prospect for such collaborations is the subject of the next section.

The systems which have been discussed in the closing section of this history can all be seen as putative alternatives to traditional knowledge-based and product centred approaches. However, to pursue the goal of expanding the choice of types of system that are available, it appears that the range of utility of each of these types of system will need to be accessed within a common framework. The next section suggests that such a framework should be derived by transferring knowledge from the world of design research to that of knowledge aided design.

2.4 Prospects for synthesis between design research and knowledge aided design

This section examines the possibility and desirability of collaboration and knowledge transfer between the disciplines of design and knowledge aided design. The key point to be discussed is the need within the knowledge aided design communities for knowledge about design. A specific area for collaboration is proposed which will be pursued in Chapter Three. This is the need that has been highlighted during this chapter for an enriched model of design to underpin the development of new systems and classes of system.

The history of the design research movement shows that this community has generated a number of useful procedural methods which have been subjected to critical re-evaluation. Many of the results of the first and second generations of design methods have made their way into the curriculum and the working practices of professional designers. Design research has also amassed a great many ethnomethodological studies of design. So design research has a large store of theoretical and practical design-knowledge which it can contribute to knowledge aided design.

This is fortunate because the problems associated with acquiring and representing the design-knowledge that is a crucial input to knowledge aided design systems remain both open and pressing. There is clearly a need for collaboration on the part of the knowledge aided design community with people who can supply design-knowledge. Design researchers are the obvious candidates for this role. As Pugh (1989) argued, knowledge based systems for design should be treated as designed objects in their own right. He therefore suggested that “major steps forward in the area of knowledge based systems and design will only be made when multi-disciplinary teams are put on to the problem—with a thorough understanding of the design activity—since it is a design problem; after all that is what we do nowadays for successful product design” (1989, p 226).

Engaging in participatory design with students and professional designers is an alternative approach. This might expand the field's own traditions of knowledge engineering. However it is beset with the problems associated with translating knowledge from the parlance of the professional designer to that of the professional systems developer. For example, Scaife *et al* (1994) show that collaboration between designers, ethnomethodologists and computer scientists can be risky and Tunnicliffe and Scrivner (1991) show that eliciting knowledge from design experts via traditional means is also problematic. So building a shared technical vocabulary and frame of reference appear to be the most pressing issues to address if collaboration is to ensue. The advantage of collaborating with design researchers is that understanding the technical languages used by designers is part of the ethnomethodological design researcher's skill set. Therefore, if design researchers treat both the domain experts and the system developers as designers, it is possible that they can fill the much needed role of translator in Pugh's multi-disciplinary teams.

Turning now from the general to the specific, one area where collaboration or knowledge transfer might be both beneficial and directly relevant to the research questions at hand is the type of models of design which underpin the development of knowledge aided design systems. As argued above it appears that the existing models which are in use within the artificial intelligence community may be partially responsible for the lack of systems that support problem setting and evaluation.

Moreover models that contain more design knowledge might help us to analyse the range of systems that are on offer and identify (1) their expected range of utility and (2) new types of systems that could be developed to fill gaps between such ranges of utility.

2.5 Summary

The histories that were presented here each contribute to the work that will be conducted in later chapters:

- The histories of design and design research introduced the first and second generation methods, which provide the core knowledge for the class of knowledge aided design systems that will be proposed in Chapter Five and for the system to be developed in Chapters Six and Seven.
- The history of knowledge aided design illustrated that the field has inherited models of design and technologies for supporting design from cognitive science and artificial intelligence. It argued that these models have led to the production of systems that do not support the broad range of designers needs.
- The history of knowledge aided design also introduced a number of systems that appear to lie outside Green's cluster of existing systems. It suggested that if the range of systems on offer is to be expanded in a principled way, then these systems will need to be evaluated within a suitable framework.
- The final section suggested that a transfer of knowledge from design research to knowledge aided design, in the form of a model of design activities, is needed. This is the central topic of the next chapter, it also underpins the analytical work presented in Chapter Four and the proposal of MODEs in Chapter Five.

Chapter Three

Creating a design activity space

3.0 Overview

The work of creating a design activity space reported in this chapter addresses the first of the hypotheses presented in the introduction:

- H1** It is possible to construct a design activity space whose axes include strategies sourced from design research for characterising and addressing design projects. This new design activity space should enable new insights to be generated into the classes of knowledge aided design systems that can and should be created.

The chapter begins with a short section describing the motivation and methodology for this work which also summarises the similarities and differences between this space and its forerunners. Next, the three main sections each define one of three axes of a design activity space which encapsulates knowledge sourced from the design research literature:

- The first axis describes a generic three stage model of design activities. The model separates design activities into analysis, synthesis and evaluation. The relevant design research literature includes work by Asimow (1962), Jones (1970), Luckman (1967) and, in particular, Lawson (1990).
- The second axis describes a model of the amount of innovation required by four types of design activities: repeat orders, configurative variant-design, true variant-design, innovative-design, and strategic-design. The relevant literature includes work by Pugh (1988), Brown and Chandrasekaran (1989), and Culverhouse (1995).
- The third axis contains the first generation and second generation design methods that were mentioned in Chapter Two's history of design research. The design research literature includes work by Jones (1970, 1992), Crickmay and Jones (1972), Bucciarelli (1988, 1994) and Rittel *et al* (Rittel, Grant and Protzen, 1972; Rittel and Webber, 1973).

A final section synthesises the complete space from these axes and discusses ways in which it might be enhanced with further research if this pilot study is deemed to be fruitful.

3.1 Motivation and method

The review of prior models of design presented in Chapter Two argued that use of the existing models has been partially responsible for the lack of knowledge aided design systems that support problem setting in design. The review culminated by reporting Goel and Pirolli's work and their claim that design has some invariants that tend to emerge across its family of disciplines (Goel and Pirolli, 1992). This claim is axiomatic to the work presented here. An argument was also presented in that chapter to suggest that findings from design research could give rise to a new model of design activities that could be used to match knowledge aided design systems to design tasks. With such a design activity space in hand it is hypothesised that the current set of types of system for knowledge aided design could be expanded.

It seems that the creation of a model of design activities which includes results of design theory and which is uncomplicated enough to serve as a starting point for system design would be a promising direction for such study. If this simple model provides theoretical insights into the nature of the relationship between designers' needs and technologies with which to support them, then the value of further studies which generate more complicated and detailed models will have been indicated.

The model that is built in this chapter is synthesised from design research which is summarised as each axis of the space is defined. So each section can be viewed as a critical literature review of design research that also serves as knowledge transfer from that field to the field of knowledge aided design.

3.1.1 Similarities and differences with prior models

The design activity space presented here is different from those of Chandrasekaran (1992), and Goel and Pirolli (1992) in the following ways:

Chandrasekaran's task analysis of design categorises a design problem as being specified by "a set of functions to be delivered by an artefact and a set of constraints to be satisfied and a technology, i.e., a repertoire of components assumed to be available and a vocabulary of relations between them" (Chandrasekaran, 1992, p 27). Chandrasekaran's model omits the internal structure of the way in which a problem is typically decomposed, it also omits the need for varying amounts of creativity in solving

the problem and it omits the type of activities which may be applied throughout the process. Such activities include problem setting, creating partial and full solutions and evaluation of the products and processes of each of these stages. In the work presented in Brown and Chandrasekaran (1989) a description of the amount of creativity that is required by a project is presented. However, the model is quite coarse grained as it only separates design into three classes.

As was noted in the introduction, the model presented by Goel and Pirolli (1992) incorporates a more realistic view of design which is based on ideas sourced from the design literature. However, it was designed for use in the setting of the cognitive psychology lab rather than in the workshop of system designers. Moreover, the model does not mention specific types of approaches to problem setting, solving or evaluation.

The design activity space presented here addresses each of the issues which have been identified as being missing from Chandrasekaran's task analysis. It also replaces Brown and Chandrasekaran's three class model of creativity with a finer grained four stage taxonomy informed by design research conducted by Culverhouse (1995). Although the space is close in spirit to the model of Goel and Pirolli, its intended use is different as it is aimed toward analysis of knowledge aided design and development of knowledge aided design systems. The space also includes explicit mention of types of design methods which can be used to address various types of project and stages within such projects. The model that is developed here is clearly influenced by those of Brown, Chandrasekaran, and Goel and Pirolli, then, but it differs from their work in terms of content and intended use.

3.2 Naive diagrams of the structure of design activities

The first product of design research that will contribute an axis to the space of design activities has been extensively described by Brian Lawson. Lawson devotes the second chapter of *How Designers Think: The Design Process Demystified* (1980, 1990) to a discussion of diagrams that reveal the structure of the design process—Similar diagrams can be found in the design research of Asimow (1962) and Luckman (1967). Lawson notes that the handbooks and training materials which many professional bodies of British designers distribute to their junior members depict this structure in strikingly similar ways. When these diagrams are compared to analogous diagrams in contemporary design research it becomes clear that they present comparatively naive descriptions of the design process. For example, they ignore irreducible ingredients of design such as human creativity and the ways that designers interact with one another,

their clients, their tools and their materials. In short, these diagrams ignore the subject matter of the remaining chapters of Lawson’s text.

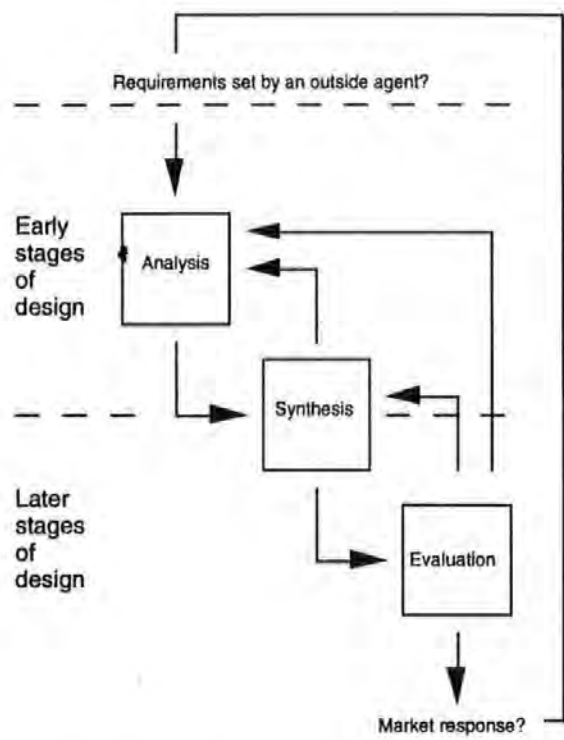


Figure 3.1: A general purpose model of design (re-drawn from Lawson, 1990) showing the stages of analysis, synthesis and evaluation being linked by edges that represent the flow of work between these stages.

The arguments presented in Chapter Two and best practice in user centred design (Draper and Norman, 1986; Martin, 1988; Laurel, 1990b; Zullighoven, 1992) suggest that understanding ‘how designers think’ should be a priority for researchers of knowledge aided design. The user centred approach to system development also suggests investigating artefacts, such as text books, articles, and diagrams, from the target domain in search of relevant knowledge. So it seems that these researchers would be justified in asking two questions about these simple diagrams. First: ‘Why are such simple diagrams in such widespread use?’. Second: ‘Despite their apparent naiveté, could they still contain knowledge that will help developers of knowledge aided design systems to understand how professional designers think?’. These questions are addressed in this section.

The diagrams of the structure of the design process which appear in Lawson’s text contain three common stages. First, an analysis of the problem and possible components with which to address it. Second, a synthesis of such components. Third, an evaluation of a proposed solution.

Typically these stages appear as nodes in directed graphs in which the edges represent the orders in which the stages may be worked through (see figure 3.1). The edges in these diagrams almost always form loops between these stages. These loops illustrate that designers work in opportunistic fashions by jumping back and forth between these stages as inspiration strikes

These graphs also illustrate that design is iterative. There are two main reasons for such iteration. First, designers tend to revisit 'preceding' stages in a design activity once their understanding of the project as a whole has been improved by addressing its 'later' stages. Some researchers argue that each iteration should become more 'focused' as designers converge toward a solution (see Pugh, 1991). Another reason for such iteration is that as a project develops, its potential market might evolve. Ideally designers become aware of this evolution and decide that they need to revisit a 'completed' stage of a design so that it can be fine-tuned towards the demands of this evolved market (Hollins and Hollins, 1991).

Lawson (1990) suggests that the main utility of such diagrams is as a navigational aid to help novice designers to find their bearings and track their progress through a design process. However, he shows that if such graphs are intended to map a full-sized design activity then they are quite unrealistic. In practice each node representing a stage in figure 3.1 might be 'expanded' to reveal a smaller version of the entire graph (this is illustrated in figure 3.2). The graph is recursive in this way because the three constituent stages of design each tend to contain elements of one another. For example, a designer may need to perform some synthesis and evaluation during analysis. As the amount of creativity or the number of components required to complete a design activity increases, it is likely that the number of recursive graphs that are embedded within each stage will also increase. This suggests that an accurate graph of a given activity might need to contain so many nested sub graphs that a designer would find it difficult to use it as a navigational aid for a full-sized design project.

Further examples of how simple such graphs seem when they are compared to full-size design processes can be found by referring to diagrams that are used to illustrate contemporary design research. For example, in *A Coherent Description of the Process of Design* Kristian Hertz (1992) investigates the creative processes that empower design. Hertz draws inspiration from such diverse disciplines as psychology, philosophy and planning. Some of the additional variables that Hertz feels are necessary to even begin the task of illustrating design projects in a realistic fashion are illustrated by figure 3.3. These include variables to model perception and the designer's environment, as well as means of physical expression and intentional communication.

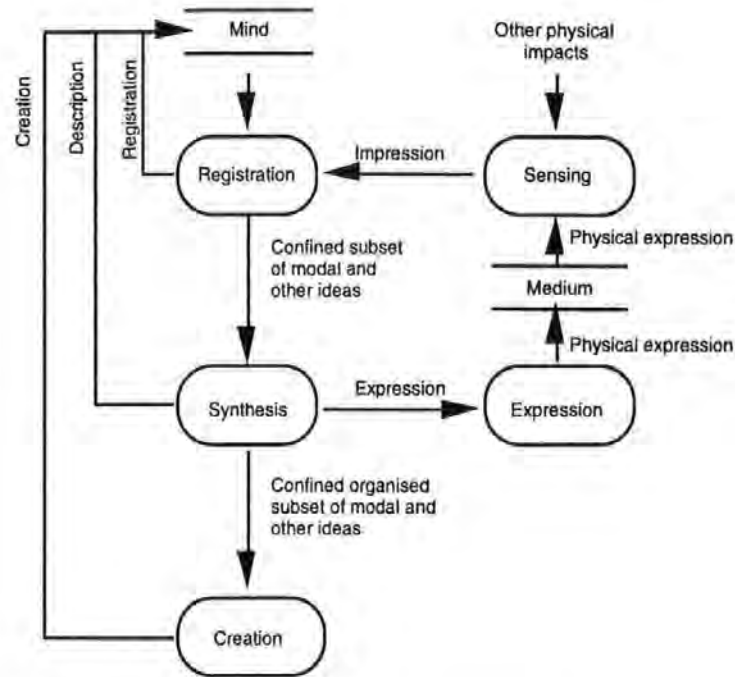


Figure 3.3: A diagram re-drawn from Hertz (1992, p 405) which shows some of the additional variables that are introduced by endorsing a more 'realistic' model of the structure of design activities.

In *How Designers Think* Lawson notes, citing Jones (1970), that it is very hard to formulate a linguistic description of design which captures the purposes and experiences of all designers (see also Potter, 1980). Later, in *Parallel Lines of Thought* (1993), Lawson suggests that drawings offer a mode of representation that supports private ambiguity within the mind of a designer and that does not commit a designer to attach a specific meaning to a thought in public before they are ready to do so (Ferguson, 1992). For example, an architect could privately view a mark they had made on a sketch as a wall, a path or as both of these architectural components. At a given point the architect could chose to state that the mark designated one or other of these components. While natural language has been found to be unsuited to the task of defining design, highly abstract diagrams might offer a means to approach an operational definition. This is because representing the structure of design processes in a graphic format allows designers to shift their private definitions of the diagram's constituent elements while maintaining a public appearance of consensus.

Achieving such consensus may be seen as a valuable goal for institutions that regulate design practice for two reasons. First, researchers into academic subject areas, such as Swales (1991), stress the importance of training new recruits to use a shared vocabulary as this promotes communication and cohesion within the group. So, these simple diagrams encode a form of shared knowledge which does not prevent designers

from maintaining private and fluctuating opinions about the nature of the stages of the design process. Second, it is arguable that the simplicity and neatness of such diagrams conveys an image of professional design as a well-managed practice to designers and their clients. So, while it can also be argued that words may be used to achieve a positively ambiguous effect, such ambiguous text may not serve the function of impressing clients¹.

Turning to the second question: 'Could these diagrams contain knowledge that will help developers of knowledge aided design systems to understand how professional designers think?'. If the preceding arguments are accepted, then the simple diagrams discussed above might represent a useful source of knowledge for theorists and developers of knowledge aided design systems. Such researchers could use them to formulate an understanding of their target domain in terms that are familiar and useful to a broad range of professional designers. This could create analogous opportunities for communication and cohesion between people working in knowledge aided design and in design itself.

However, while the kinds of diagrams that illustrate contemporary design research also contain knowledge that might be useful to theorists and developers of knowledge aided design systems, it seems important to stress that understanding the conventions of the design research community is a slightly different goal from understanding the conventions of the design community itself.

In academic terms both of these activities seem to lead to desirable goals. However, as we shall see in Chapter Four, the present theory of knowledge aided design is very sparse. Moreover, the number of existing knowledge aided design systems which are available for study is also small. To expand its theoretical base, knowledge aided design needs to study systems that are accepted by professional designers. So building systems which meet the needs of working designers seems a more pressing goal than building systems that implement cutting-edge views of the design process. From this point of view the naive diagrams discussed in this section offer a point of entry into the culture of the professional British designer that is useful to researchers of knowledge aided design.

Chapter Four will present arguments and examples to support the claim that each stage in a design activity might be best emulated or supported with different knowledge aided technologies or combinations of such technologies. If these arguments and examples are valid, then the insight that each stage is likely to contain elements of the other two stages implies that a knowledge aided design system, which is aimed at a particular stage, had best contain components to support each of the stages.

¹ I thank Professor Beardon and Dr Hollins for bringing this issue to my attention.

Alternatively, if a system aims to support the whole design activity, then it is important that it is easy for a designer to move between the component technologies without losing the dominant context of whichever stages they are currently working in.

A final reason for treating these diagrams as a useful source of design knowledge is based on the premise that designers often use external representations during the design process. Since designers are familiar with these particular representations, it seems plausible to suggest that they might be embedded within graphical user interface to certain knowledge aided design systems. The CITIS system (Gilleard and Lee, 1996), described in Chapter Two, provides an embryonic illustration of this approach.

3.3 Taxonomies of design, creativity and inventiveness

The second product of design research that will contribute an axis to the space of design activities is based on Brown and Chandrasekaran's (1989) taxonomy of different types of design activities. This contains three types of design activity which progressively call for a greater amount of design, creativity, and inventiveness on the part of a designer. It also provides a rough estimate of the proportion of design briefs which fall into each class. Brown and Chandrasekaran call these stages class-one, class-two and class-three design-problems.

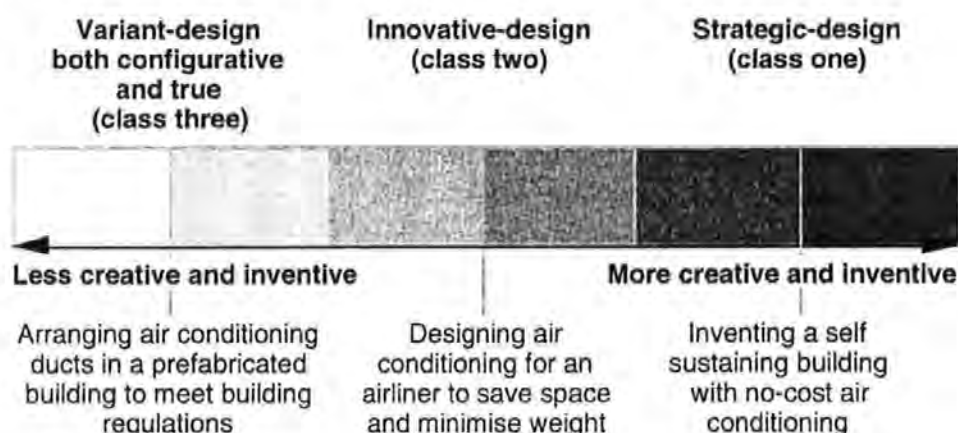


Figure 3.4: A continuum of design activities. Tasks that require less creativity and which are approached with familiar components, objectives or working practices are shown in a lighter tone than tasks that require more creativity and the invention of components, objectives or working practices. Examples of a variant-design activity, an innovative-design activity and a strategic-design activity are shown in their place along the continuum.

Brown and Chandrasekaran's taxonomy will be updated by synthesising it with a model presented by Culverhouse (1995). Culverhouse presents a similar taxonomy with four

stages and a more descriptive set of names: repeat orders (Culverhouse seems to have included repeat orders for completeness as these do not entail design), variant-design, innovative-design and strategic-design. Gero and Maher (1993) provide a high level distinction between variant-design and innovative-design and strategic-design. They refer to variant-design as ‘routine design’ and to innovative-design and strategic-design as ‘non-routine design’. They argue that artefacts (or working methods) which do not introduce new design variables, but merely require new combinations of values for these variables to be selected, are routine. In contrast, artefacts (or working methods) that call for the invention of new variables as well as the selection of appropriate values are non-routine. In this section Culverhouse’s names for these classes of design will be adopted (see figures 3.4 and 3.5).

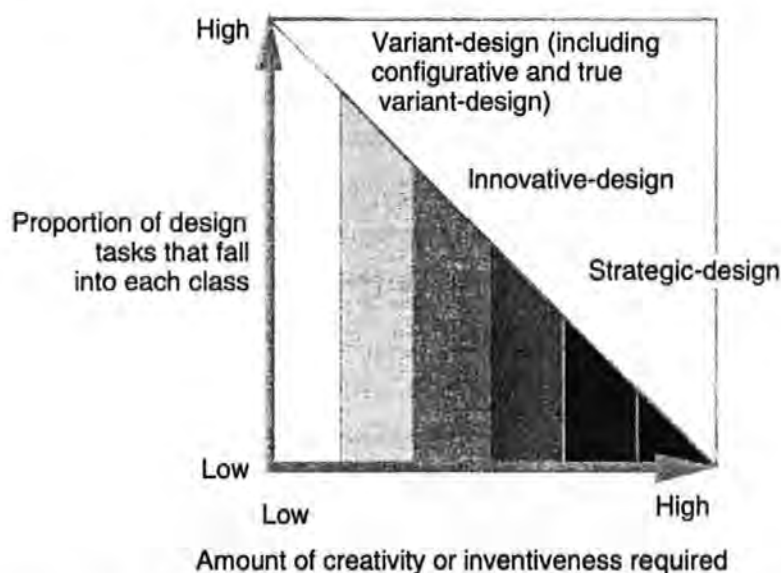


Figure 3.5: A continuum of design activities represented as a histogram which indicates how often each type of task is approached in proportion to the other types of task (e.g. Brown and Chandrasekaran (1989) and Heath (1984, 1993) for estimates to support the shape of the histogram). Tasks that require less creativity and which are approached with familiar components, objectives or working practices are shown in a lighter tone than tasks that require more creativity and the invention of components, objectives or working practices.

One methodological issue which is raised by employing Culverhouse’s taxonomy is his use of a percentage value of new ideas and practices that are necessary for a project to fall within a particular category. For example, Culverhouse estimated that innovative-design activities call for the invention of 20% to 50% of new ideas or processes. Clearly estimating such figures before design is undertaken with a fine degree of accuracy is hard. Moreover, criteria must be agreed upon with which to define what qualifies as a new idea or a new process. These are the types of issues which second generation design methods flagged as being problematic (this will be discussed in the next section).

For these reasons the numbers associated with categories in Culverhouse's should be read as heuristics to guide and inspire rather than as rigid or even provable statistics¹.

The most elementary type of design in the hybrid Brown and Chandrasekaran / Culverhouse taxonomy is repeat order-design. In repeat order-design previously designed goods or services are re-supplied. Since repeat orders entail a minimal amount of design, at best, they will not feature in the model space being built here.

The second most elementary type of design in the hybrid Brown and Chandrasekaran / Culverhouse taxonomy is variant-design (Brown and Chandrasekaran called this 'Class Three design'). Culverhouse estimates that variant-design activities call for the invention of 1% to 20% of new knowledge in the design stage or during production engineering. Simple variant-design activities demand the routine configuration of known components, which perform known functions, according to known compositional rules and testing procedures. The arrangement of 'off the shelf' air conditioning ducts in a prefabricated building to meet well articulated building codes is a prototypical variant-design activity. Culverhouse suggests that more complicated variant-design activities may be approached by: (i) innovating existing products by extension; (ii) refining existing technological usage or (iii) modifying manufacturing technology. At the risk of introducing extra terminology, then, variant-design might usefully be subdivided into 'configurative' variant-design, which is entirely routine, and 'true' variant design, which begins to demand more innovative thinking (see the overlap between these categories in Figure 3.6).

An innovative-design activity, or what Brown and Chandrasekaran name 'Class Two design', is more challenging: designers working on such a task may need to re-design the familiar components and functions of a well known type of product. They may also need to synthesise competing concepts and to design new testing procedures. In some cases they will need to invent or create new ways of working. Culverhouse estimates that innovative-design activities call for the invention of 20% to 50% of new knowledge in the design stage or during production engineering. Brown and Chandrasekaran observe that this class is practised less often than variant-design. The arrangement of air conditioning ducts is the latest in a sequence of aeroplanes to harmonise with concurrently redesigned subsystems is a prototypical innovative-design activity. Class two design activities can also be called ordinary design activities. Culverhouse suggests that innovative-design activities may be approached by: (i) combining features from different existing products; (ii) converting the technologies underlying a key feature of a product (e.g., from analogue to digital) or (iii) applying significantly new manufacturing technology.

¹I thank Professor Beardon and Dr Hollins for raising this issue.

Pugh provided some interesting additional information on the differences between the process of designing a variant product and an innovative product (Pugh, 1989). He asserted that a variant product would be designed by addressing the needs of the market, specifying a product, engaging in concept design, performing detail design and then manufacturing and selling the product. In the case of innovative-design the order of these activities would be changed so that concept design preceded the specification stage. This is because a new technology, material or other ingredient is a logical precursor to a viable innovative-specification.

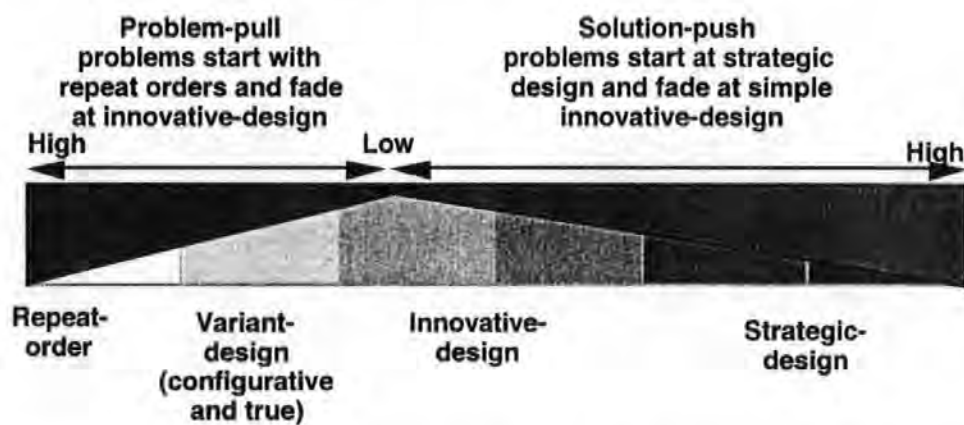


Figure 3.6: Problem-pull projects are represented by the triangle on the left covering the whole of repeat orders, slightly less of variant-design and the simplest cases of innovative-design. The triangle on the right represents solution-push projects which cover the whole of high-end strategic design, slightly less of low end strategic-design, and progressively less of innovative-design and variant-design. The overlap illustrates that it can be hard to predict whether a project is an example of 'pull' or 'push'. Some projects might contain elements of both types of problem.

Strategic-design problems are even more formidable than innovative-design problems. A designer working on a strategic-design problem must innovate to address novel, ill-specified goals and should expect little orientation from known problem decomposition strategies. Culverhouse notes that strategic-design is likely to be addressed in multi-disciplinary work groups rather than by individual designers and that such groups do not normally work to short-term deadlines. Culverhouse also estimates that innovative-design activities call for the invention of more than 50% of new knowledge in the design stage or during production engineering. The creation of a self-sufficient building with no-cost air conditioning is an example of strategic-design. To create such a product a group of architects might have begun by securing a novel source of funding. The process could entail other 'political acts' such as affecting the beliefs of politicians as they become expressed in new building codes. It could also require inventive acts such as arranging air conditioning in structures that do not require the familiar specialised duct. Culverhouse suggests that strategic-design activities may be approached by: (i)

gaining insight into new physical principles; (ii) exploring the operational limits of such principles; (iii) defining new manufacturing tolerances; (iv) developing demonstration products that embody these new principles. However, by its nature, strategic-design appears harder to prescriptively characterise than variant- or innovative-design.

Strategic-design is at the frontier of design activities that professional designers tend to encounter. However it does not represent the limit of human endeavour. Unfortunately, while the methods which lead to successful strategic-design are dimly-understood, the processes that underlie creativity and inventiveness, which would appear to the right of strategic-design in figures 3.4 and 3.5, remain utterly opaque to the science of cognitive psychology (c.f., Boden, 1990, 1993) and the art of business management (c.f., Clark, 1988; Druckner, 1985). The difference between inventiveness and creativity on the part of a designer in this context is that, as Pye observed, inventiveness involves the discovery of something in the world—something that can subsequently be applied in useful ways—whereas creativity involves the juxtaposition of existing ideas in a novel way: “Invention is the process of discovering a principle. Design is the process of applying a principle. The inventor discovers a class of system—a generalisation—and the designer prescribes a particular embodiment of it to suit the particular result, objects and source of energy he is concerned with” (Pye, 1964, p 19).

In the later 1990s strategic-design probably defines the current limit of types of design which knowledge aided designers can create specific products to support (see papers in Gero and Maher, 1993)—for contrary opinions see papers in Ford, Glymour and Hayes (1995) where, for example, the circumstances surrounding the award of an American patent to Douglas Lenat’s EURISKO program are discussed.

Pugh’s insight that the conceptual design phase of the Total Design method would logically precede the specifications stage is even more appropriate for Strategic-design. As the percentage of new ideas rises toward the theoretical maximum it becomes harder to rigorously specify a product whose components will be fabricated using unknown materials, configurations or manufacturing techniques. Crickmay and Jones (1972) clarify this point by inventing the terms ‘problem-pull’ and ‘solution-push’. Respectively, these terms distinguish situations where a known problem exists which needs to be solved from those where solutions are invented ahead of the problems they will address. Crickmay and Jones also suggest that the design of standard parts, e.g., ‘Meccano, numbers, the alphabet’ is typical of ‘solution-push’. This argument could be extended to include the invention of design methods. Prototypically, repeat-design and variant-design are progressively less canonical examples of ‘problem-pull’, then, and

innovative-design and strategic-design are progressively more canonical examples of 'solution-push' (see figure 3.6).

This taxonomy contains knowledge that should be useful to researchers and developers of knowledge aided design systems. Chapter Four will argue that design activities which require different percentages of creative or inventive thought are best emulated or supported using different knowledge aided technologies or hybrids of such technologies. If these arguments are accepted, then a mapping between this axis of the design-activity space and Green's model-space for knowledge aided design system might provide an initial set of components for a system that is aimed at a design activity that appears to require a particular percentage of creative or inventive design.

Pugh's observation that the specification and conceptual design phases of product design sometimes need to switch their logical ordering implies that software needs to be designed so that this is easy to achieve. This is especially true for software to support innovative- or strategic-design, as the category that a project will eventually fall into may not be clear at its outset. Crickmay and Jones' 'problem-pull' and 'solution-push' terminology provides a very high level description of the differences between these two types of design.

3.4 First and second generation design methods

The third product of design research, that will contribute an axis to the space of design activities, is the concept of a design method. As Chapter Two described, design research has spawned two generations of design methods and both of these will contribute to the contents of this axis. Consequently, this section will be divided into two subsections. Each of these will describe a generation of design methods in greater detail than was required in Chapter Two.

3.4.1 First generation design methods

Chapter Two presented a history of design research in Great Britain which showed that the first generation of design researchers addressed two goals:

- to help designers to perform design activities in a rational fashion
- to model these activities in an objective fashion.

Pursuing these goals with the methodologies of the social sciences, cybernetics and operations research, led these researchers to invent ordered strategies which named specific types of design activities and decomposed them into a finite sequence of provably feasible sub-activities. These named and logically ordered strategies will be referred to here as procedural design methods. However, naming this generation of design methods is quite problematic. This is because there is little agreement in the literature. Some design researchers chose to emphasise the systematic nature of their methods and others drifted away from the idea. Cross (1984), for example, uses the terms systematic design models and systematic procedures. Jones set out by using terms such as systematic design methods (1970) but later simply chose to call them design methods (1992). With the benefit of hindsight, the procedural aspect of these methods will be emphasised in contrast to the less procedural nature of the second generation of design methods.

The fact that procedural design method is composed of a sequence of sub-activities, which have previously been found to be feasible in some situations, does not imply: (i) that these sub-activities will be appropriate in arbitrary situations; (ii) that they can be approached without the needs for creativity and experience on the part of a designer; or (iii) that there is no need for shared experiences on the part of a design team. On the contrary, this section will conclude with a précis of an ethnographic study by Bucciarelli (Bucciarelli, 1988, 1994) which shows that a particular procedural design method can break down due to these factors.

However, it should be stressed that in appropriate situations the potential advantages of adopting a procedural design method include the facts that they can help designers to navigate a particular type of design activity and can carry low costs for training and dissemination which makes them attractive to managers of design projects. These training and dissemination costs can be low because a procedural design method can often be represented using one or two pages of undergraduate-level text.

For example, Bucciarelli's study shows how a design manager assumes that his team will be able to learn the 'Pugh method' from copies of a text which he has distributed: "Sergio [speaking to design team]: *That's all we could come up with. I know it's late, so we're going to have to get together again ... and do the Pugh method. I sent you all a copy of that, didn't I?*" (Bucciarelli, 1994, p 39) . Such text is also often illustrated with directed graphs in which the nodes stand for the clearly articulated activities and the directed arcs stand for temporal orderings of these activities.

The text describing the nodes within these graphs will usually describe explicit inputs to the associated activities and will also describe their expected outputs or end-products. It may also describe associated working materials. For example some of Pugh's

descriptions include layouts for structured notebooks (Pugh, 1991). The nodes representing sub-activities in a procedural design method can often be expanded to reveal further nested sub-activities. These nodes and their nested sub-activities may be separated into what Jones calls glass-box and black-box methods (Jones, 1970, 1992). The difference between these two types of activities is analogous to the difference between a quantitative research method in the physical sciences and a qualitative research method in the social sciences.

In the case of a glass-box component of a procedural design method, it should ideally be possible to reduce the content of the method to a set of objective rules that any suitably trained person could follow. Moreover, two people who start from the same inputs and who follow the same set of rules should produce equivalent outputs. As Jones has observed, the scientific ideology behind a glass-box approach to design assumes that people are akin to information processing devices, who can be ‘fed’ with fixed informational inputs, who will follow a predetermined series of steps to generate an output, and who will recognise an optimal output when one is found. This shadows the viewpoint of the early research into artificial intelligence which, as Chapter Two mentioned, was inspired by the world view of cognitive science and psychology (c.f., Boden, 1988). This approach was dominant in the period between the 1940s and the 1980s and it is still embraced by certain philosophers of mind while being rejected by others.

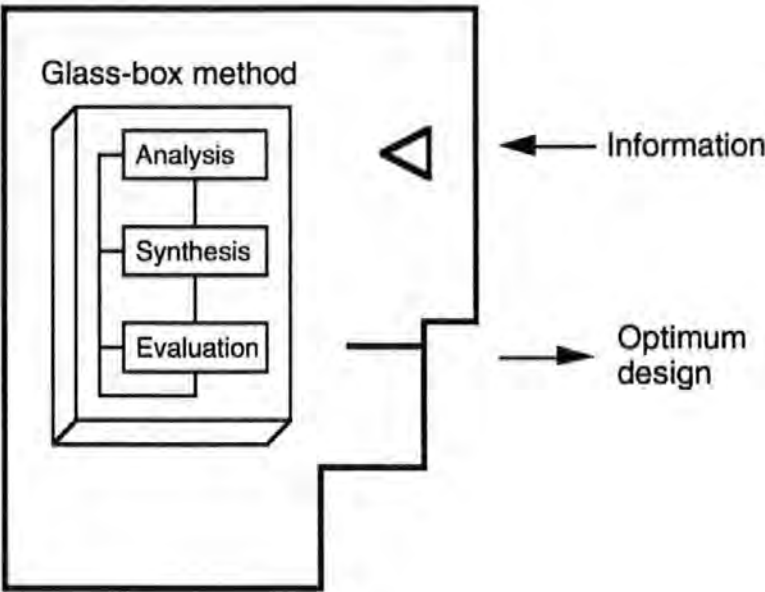


Figure 3.7: ‘The designer as a computer’, adapted from Jones (1992, p 50). From the perspective of a developer of a glass-box method a designer is treated as an information processor or human computer.

Black-box sub-methods are very different and they are grounded on an alternative theory of mind with roots in behaviourism (c.f., Ryle, 1949), cybernetics (c.f., Heims, 1992) and neural networks (c.f., Clark, 1989). These methods both assume and require the context sensitivity, imagination, creativity and inventiveness which human designers bring to their work. For example, Green (1992, pp 16-17) recounts how an expert designer brings context to a task of engine design which might, at first blush, seem so well defined that it could be automated or addressed with an entirely glass-box method.

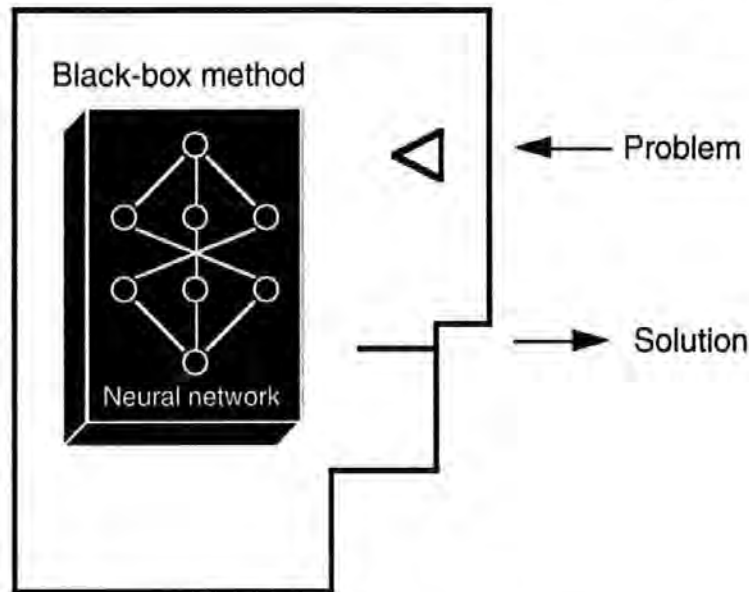


Figure 3.8: The designer as black-box. From the perspective of a developer of a black-box method a designer is treated as a being in the world with a nervous system and a point of view that cannot be explained by current psychology.

Since people tend to have diverse life experiences and associated creative abilities¹, it is likely that different people would characterise the inputs to a black-box sub-method differently, respond to its instructions in divergent ways, then complete the method by producing distinguishable outputs.

Since glass-box and black-box sub-methods in a procedural design method are so different, the requirements for specifying them are quite distinct. Since these are concepts that will play an important role in Chapter Four, this distinction will be described here in detail.

Specifying a glass-box sub-method requires a theory that encompasses a design activity in a wholly explicable, observable and rational way. In other words, a glass-box

¹See Rothenberg and Hausman (1976) for a set of readings on the scientific history of the study of creativity. See Gero and Maher (1993) for a collection of papers describing the cross-over between *Modelling Creativity and Knowledge-Based Creative Design*.

sub-method is the equivalent of an 'algorithm' that requires a finite set of inputs. These include: known objectives, known variables, known design strategies, as well as known criteria with which to assess the success of each step in the algorithm.

That is a lot of knowledge for a design researcher to acquire and, as the section in Chapter Two on knowledge aided design mentioned, it is difficult to represent such knowledge in an unambiguous format. Moreover, as Jones adds, this knowledge needs to be given context by an understanding of the design domain which is so complete that the glass-box design method can be used to evaluate its own steps by logic rather than experiment (Jones, 1970, 1992).

This requirement causes glass-box sub-methods to be hard to formulate and limited in scope. This is because when people perform design activities they sometimes discover aspects of the task that they had not considered when they set their requirements. Indeed they often operate on the basis of tacit knowledge which they either do not or could not, articulate beforehand (see Dreyfus, 1992; Coyne and Snodgrass, 1993). Moreover, when a design is evaluated, a designer's world model sometimes turns out to be at odds with the real world. At this point, the designer needs to update their world model on the basis of some empirical experiments. Since a glass-box sub-method is simply a product of invention and design, there is no special guarantee that it will be immune from the problems associated with incomplete or faulty information that beset conventional designers.

The types of design activities that *can* be modelled by glass-box sub-methods are restricted to relatively context-free activities which can be addressed by means of mathematical optimisation. A specific example of such a method can be found in Pye's discussion of optimising the layout of the beams in a roof (Pye, 1964). Other examples, which are mentioned by Crickmay and Jones (1972), include domains in which problems can be 'split', such as chemical plant design, electricity supply design and telephone systems design, as well as domains in which problems cannot be 'split', including beams, rotor design, circuit design and electric motor design.

At present, few design activities can be formalised as context free, optimisation problems. One of the reasons for this is that certain design processes, such as Gordon's (1961) Syntectics, rely on these 'inventive' human abilities which, while frequently observed, are incompletely understood. Indeed many design researchers have noted that the mechanisms behind the most powerful human design abilities, which are typified by such folk psychological terms as 'invention' and 'analogy', are hidden from designers' powers of introspection. For example, when Pye describes how he has invented an artefact he observes that his: "thinking has not been conscious, but it must have been done" (Pye, 1964, p 43) and, more regrettably, that: "It is unfortunately impossible now

to discover what trains of thought did in fact lead to the invention of our devices" (*Op Cite*, p 68). Unfortunately, these powers are also hidden from researchers who are seeking formal accounts of creativity in design (c.f., Pye, 1964; Broadbent, 1967; Jones, 1970, 1992; Lawson, 1990. For a corresponding view from the world of cognitive psychology and artificial intelligence see Boden's reviews of creativity literature, 1990, 1993).

Jones described such processes as black-boxes because: "we can say that the human designer is capable of producing outputs in which he has confidence, and which often succeed, without his being able to say how these outputs were obtained" (Jones, 1992, p 46). At present, given psychology's lack of understanding of the mechanisms that empower these 'outputs', any design processes whose instructions call, however tacitly, for skills such as imagination, analogy, or close abilities, should be regarded as a black-box. This brings the discussion of procedural design methods to the particular difficulties involved in inventing a black-box sub-method.

Black-box sub-methods have properties that make them harder to discover than glass-box sub-methods. This is especially apparent in the areas of sub-dividing a task and of providing designers with external representations such as Pugh's structured note books (Pugh, 1991).

One contribution that a procedural design method can offer a designer is a means to 'divide and conquer' a task. This allows the task to be approached in parallel or for designers to concentrate their resources on one particular part of the problem at a time. As Jones (1970, 1992) observes, the kind of design activities which will make good glass-box sub-methods take place 'in public' where techniques from the decision sciences can be applied. However, since the constituent black-box processes and their interactions take place 'in private', within the designer's head, it is less likely that such techniques will enable a design researcher to uncover ways to sub-divide the activity.

A second benefit that a design researcher's procedural design method can offer a designer is a means to record some of their thinking in an external representation. This helps a problem solver to concentrate on solving the unknown part of a problem by freeing some of its known parts from their short and long term memory. This is likely to be easier to achieve for an activity that will make a satisfactory glass-box sub-method than for an activity that has black-box characteristics. This is because glass-box sub-methods tend to be analytical and are hence prone to solution via means of standard diagrams and other symbolic representations including flow charts. The kind of problems that are amenable to skills, such as analogical thinking, often benefit from more personalised, idiosyncratic and temporary external representations.

As the introduction to this subsection mentioned, procedural design methods can be very easy to learn but there are circumstances where they break down. This was predicted by Crickmay and Jones (1972) who argued that, in the case of 'unsplittable and non-repetitive problems', a leading designer needs to make the critical decisions and glass-box methods are hard to apply and in the case of 'unfamiliar problems' the "critical insights must occur in one head" (*Op Cite*, p 25).

One circumstance that led to a breakdown was documented in detail by Bucciarelli in *Designing Engineers*. In this book Bucciarelli describes and interprets the activities of three teams of engineering designers. The members of these teams develop diverse products such as an X-ray luggage inspection system, a photo print machine and a solar-powered desalination plant. The text features descriptions of individual design activities. Bucciarelli also presents edited highlights from his objective records and subjective memories of meetings with members of the individual design teams. In one example, a few pages of conversational transcripts show how an established prescriptive design method can fail.

When Bucciarelli dissects his transcripts, he draws attention to the fact that the manager in charge of a meeting at which the team had attempted to follow the method came to refer to that meeting as "the disaster meeting". In this meeting it became apparent that each designer was working from a different set of assumptions toward their own version of a goal for the product. More perniciously, this fact took some time to become apparent to the manager and the other participants. Bucciarelli analyses the transcript and shows that while the method assumes that a group of designers share a clear and well-articulated interpretation, it cannot provide a means with which to generate these goals nor a shared understanding of them.

This example illustrates that while procedural design methods can seem to contain glass-box components, which appear as if they should lead designers from similar starting points to similar outputs, these components are generally 'wrapped up' in black-boxes. As Jones came to realise: "There were black-box methods, like synectics, which worked but nobody knew why, and glass-box methods, like decision theory, which were logically clear but which didn't work ..." (Jones, 1977, in Cross, 1984, p 331). Jones reflected upon this for some years and discovered "...what's striking is that each method begins with a first stage that is extremely difficult to do, which has no description of how to do it, which is intuitive." (*Op Cite*, p 331). In other words the inputs and outputs of glass-box methods were buffered by black-boxes.

One way to interpret this insight might be to claim that there are no differences between glass- and black-box methods. However this move seems too drastic, especially since certain glass-box methods, such as mathematical formulae for roof

beam design, demonstrably exist. Moreover, in the current context, these glass-box methods are valuable: since they are grounded by a computational view of mind, they are kinds of methods that should be amenable to being emulated by a knowledge based system.

It would probably be better to claim that glass-box methods do exist but they are fragile and require the support of a black-box 'scaffold': Pugh (1989) was realistic on this point by noting that objective methods for roof beam design only tended to produce similar results. As Crickmay and Jones observed: "Rational thinking on its own wastes the vast information planning capacity of the nervous system. Intuition on its own depends too much on the experience of one designer." (1972, p 10). Figure 3.9 shows a hybrid model of design in which rational and teachable glass-box methods exist in the supporting context of black-box methods and in the native human abilities that empower them. In such a model, problems have to be recognised by the designer and certain design activities can be evaluated 'in the head' using a self-contained model whereas others must be evaluated 'in the world' by experimentation.

These observations introduce the subject matter of the next subsection. This will survey second generation design methods. These are found in a body of literature which recognised that design problems are hard to pre-define. They therefore require designers to know when to surpass pre-defined procedural methods and bring their own creativity and inventiveness to strategies for problem setting, solving and evaluation (see figure 3.9).

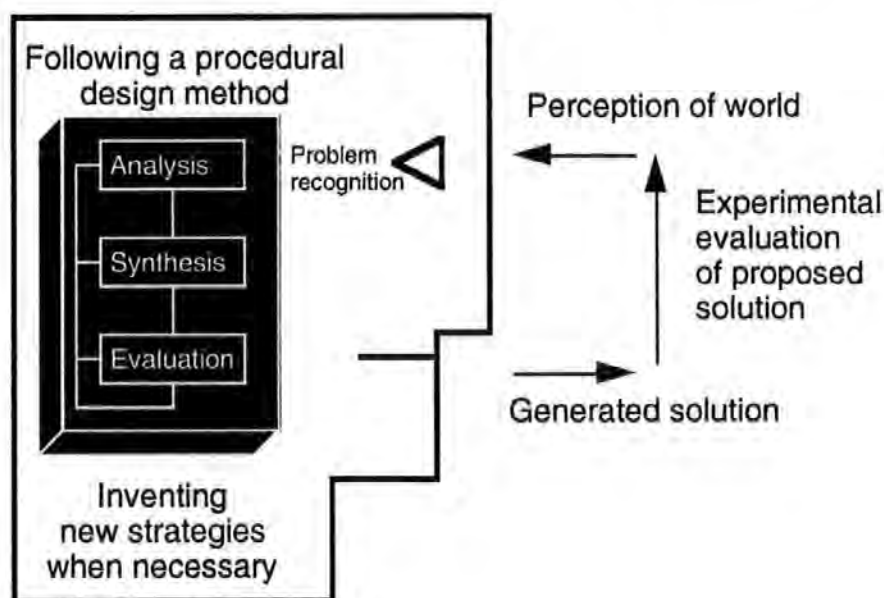


Figure 3.9: A hybrid model of the designer following a procedural design method. The glass-box methods are surrounded by, embedded in and supported by black-box methods.

3.4.2 Second generation design methods

In Chapter Two it was suggested that the 'algorithmic shape' of the first generation of procedural design methods was influenced by various political factors which came to the fore at the height of the cold war era. During the 1970s there was a reaction against these methods which has been termed the second generation of the design methods movement. Researchers such as Alexander and Jones (Alexander and Jacobson, 1971; Crickmay and Jones, 1972; and Jones 1970, 1977, 1991) came to mistrust the kind of methods they had invented and documented in the 1960s. Meanwhile, others, such as Rittel and Webber (Rittel, Grant and Protzen, 1972; Rittel and Webber, 1973), looked to new traditions, such as social planning, in order to critique such methods. Although the contents of this literature have been termed second generation design methods, they are visibly less methodical than first generation design methods. This may explain why Crickmay and Jones commented of some of their works that represented their second generation methods: "the status of the [works] is academically low: it is that of personal opinion. The opinions are, however, informed by the experience of many design theorists and teachers." (Crickmay and Jones, 1972, p 5).

Despite this perception, the 1970s witnessed the publication of a range of new methods that characterised design as a more difficult problem than goal-directed search or optimisation and hence proposed methodologies that were less algorithmic, methodical or prescriptive. It is probable, then, that the new design forums, conferences and academic institutions which had been developed by the 1970s, together with a changed social and political climate, made this possible. This subsection will review some of the critiques which this second generation of design researchers offered of the first generation of design methods. It will also explain some of the working strategies which they proposed that designers should adopt. In particular, it will examine the work of Alexander, Jones, and Rittel and Webber.

Alexander had been one of the early and key players in the first generation of design researchers. For example, his *Notes on the Synthesis of Form* (Alexander, 1962, 1971) was an oft-cited text. However, as Cross (1984) observed, the second edition of this text, which was published in 1971, contained an introduction that critiqued his earlier approach. Essentially Alexander could be said to have removed the foundation of a prior decade of work.

In the same year Alexander explained to Jacobson that "my feeling about methodology is that there are certain mundane problems that have been solved—and I mean incredibly mundane...The fact is that [design methods have] solved very few

problems for me in my design work. Most of the difficulties of design are not of the computable sort.” (Alexander and Jacobson, 1971, p 312).

These beliefs led Alexander to recognise that gathering requirements from people was one such non-computational element of design (Alexander *et al*, 1975). Hence he had begun to embrace the kind of participatory design in architecture which was mentioned in Chapter Two. He felt that involving people in the specification, design and evaluations of buildings was an important way to gather requirements so that he could build ‘beautiful buildings’, which was what he had searched for through use of first generation design methods.

Later, however, Alexander began to collect and publish what he called ‘design patterns’ and ‘pattern languages’ (Alexander *et al*, 1977). These were essentially *just* catalogues of ‘good ideas’, grouping components of a building which could be reassembled in subtly different ways in different situations. However, since these were models rather than methods, the onus for the intelligent re-usage of these patterns was placed with the architect rather than being pre-specified by a methodology: Crickmay and Jones (1972) observed that choosing and adapting a design method required a skilful mixture of rational and imaginative thought on the part of a designer, and selecting and adapting these patterns required similar skills from architects.

These design patterns have met with demonstrable success in architecture in the late 1970s, 1980s and 1990s and they are now being embraced by the software engineering community in the later 1990s (Gamma, 1995). The success of Alexander’s pattern languages in these two domains demonstrates that while documenting ‘good ideas’ might lack some of the academic prestige of capturing design methodologies, this approach has led to endorsements from working architects, designers and software engineers, as well as design theorists.

Jones contributed to the second generation of design methodology in the areas of design education and design research. In the context of creating material for an open access course within the Open University, he collaborated with Crickmay (Crickmay and Jones, 1972) to write *Imagination and Method: Designing as a Response to Life as a Whole*. This text introduced a generation of students of design to the backlash against first generation methods. He also wrote more detailed articles in the specialised design research literature which advocated a move toward using ‘chance’ to generate new ideas and to broaden a designer’s thinking. Many of these articles were collected in *Essays in Design and Designing Designing* (Jones, 1984, 1991).

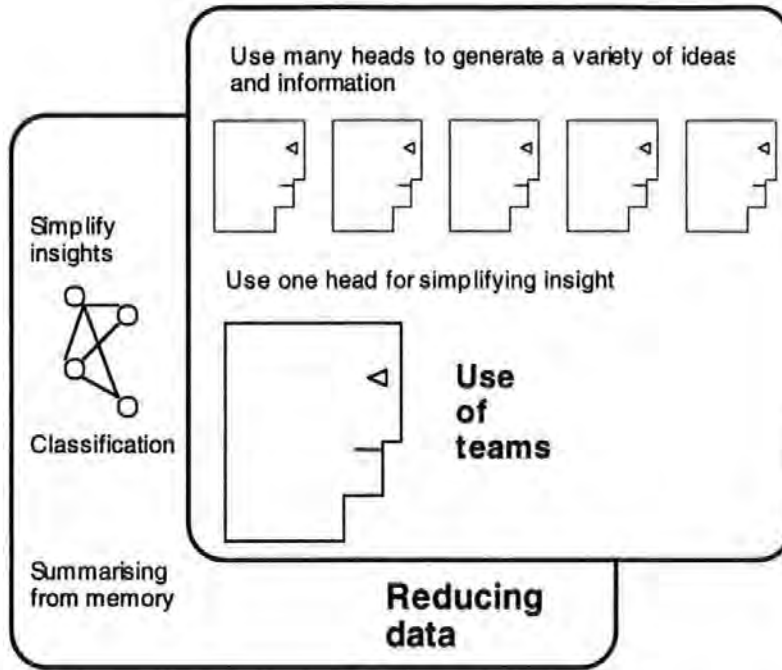


Figure 3.10: Two meta-cards re-drawn from Crickmay and Jones (1972). The paper presents nineteen such cards on topics including: strategy reviews, professional procedure and testing.

Imagination and Method: Designing as a Response to Life as a Whole is a valuable text for two reasons. It stresses very heavily the need for designers to design their own methodologies which balance procedural design methods with more imaginative and risky approaches. For example, the text gives grounded advice on topics such as: ‘allowing the problem to emerge’, ‘breaking with the status quo’, ‘switching strategies’, ‘transforming problems’, ‘meta thinking (thinking before acting)’, and ‘setting social, mental and physical conditions’. It also provided a set of meta-cards each of which illustrated “a rational description of an obscure mental process” (Crickmay and Jones, 1972, p 21). The purpose of these cards was to remind design students to think about designing the process of designing at the same time as they were designing an artefact or system (see figure 3.10). So each card can be seen as representing a miniature version of a second generation design method.

The papers collected in *Designing Designing* (Jones, 1991) are more philosophical in scope. Jones’ *Design Methods: Seeds of Human Futures* (1970; 1992) had already illustrated the inapplicability of craft-based approaches to modern design problems. This text broke new ground by copiously illustrating case studies of applying chance-based methods to assignments such as illustrating a lecture. Much of the philosophy which inspired this work was derived from that of the composer John Cage, who used dice and other random media to configure his musical compositions. As Cross notes, although these ideas had a rational basis, which was to extend the range of inputs that a

designer would consider, some observers came to perceive them as “difficult to accept...[thinking they were]...the antithesis of design” (Cross, 1984, p 306).

While some of Jones’ ideas of the 1970s were too radical for the climates of the 1980s and 1990s, Rittel and Webber’s critiques of the first generation of design methods have continued to exert as profound an influence as those of Alexander—especially within the software engineering community: (see Fischer, 1992; Budgen, 1995). The American design theorist Richard Buchanan has championed the importance of this work in mainstream design literature throughout the 1990s, especially in Buchanan (1995). This is rather surprising as Rittel and Webber’s ideas are highly philosophical and do not supply a methodology. Instead they characterise design in a way that is more complicated and probably more accurate than first generation models. Their major contribution to the design methods literature is to have transferred the concepts of ‘wicked’ and ‘tame’ problems from the literature of social planning to that of design.

In *Planning Problems are Wicked Problems*, Rittel and Webber defined a class of problems that designers of social systems faced as being ‘wicked’. They argued that these contrast with the kind of ‘tame’ problems that scientists, and ‘some’ engineers routinely faced. This was because: “problems in the natural sciences, which are definable and separable may have solutions that are findable...[whereas]...those of governmental planning—and especially those of social or policy planning—are ill-defined and call for elusive political judgement for resolution” (Rittel and Webber, 1973, p 163).

Rittel and Webber deliberately opted for the term ‘resolution’ in place of ‘solution’ to indicate that such problems are never wholly solvable and might need to be continuously ‘resolved’. This process of continual problem resolution contrasts with problem solving which is embedded in a clear set of rules and an accompanying mission statement. Thus ‘defining human rights’ is an activity which might lead to a sequence of resolutions whereas ‘solving a particular chess puzzle in a given number of moves’ is a problem that might lead to a sequence of verifiable solutions. In Alexander’s terminology, problems which require resolution rather than solution are ‘not of the computable sort’ (Alexander and Jacobson, 1971).

In order to distinguish wicked problems from tame ones, Rittel and Webber (1973) provided a checklist of ten inherent features of wickedness. These can be summarised as follows: (i) the problem has no definitive description; (ii) the problem solvers cannot tell when the problem is solved; (iii) problem solvers must decide whether a resolution is good or bad rather than correct or incorrect; (iv) the perceived quality of a resolution might increase or decrease over time; (v) a finite set of candidate solutions cannot be enumerated at the outset; (vi) each attempt to solve the problem incurs unrecoverable

costs; (vii) the problem cannot be reduced to a familiar problem that has already been solved; (viii) the problem can be symptomatic of worse problems that may not reveal themselves until it has been resolved; (ix) the symptoms which identify the problem might be accounted for in many different ways; and (x) a resolution has the potential to cause harm and, hence, problem solvers may owe responsibilities to themselves or others.

Although Rittel and Webber distinguish tame scientific and engineering problems from wicked societal problems there are reasons to argue that the reach of wickedness extends to science, engineering and, perhaps, even to recreational problem solving. For example, it can be argued that *if* certain classes of scientific and engineering problems are tame, then this is because glass-box methods have been developed to guide people and provide them with frames of reference with which to set individual problems and to evaluate their solutions. However, the development of these methods would have been a wicked problem in its own right and, as the preceding discussion of glass- and black-box methods argued, the application of these glass-box methods requires the context of black-box methods. In the specific case of science, Kuhn (1962) has shown that paradigms change, opinions differ and only routine science takes place in the kind of glass-box idyll to which Rittel and Webber allude—and then only until a paradigm shift occurs. In the case of engineering, ethnomethodological studies show that engineering ‘design-worlds’ are the subject of constant negotiation and interpretation between designers and others (Bucciarelli, 1994). In the case of recreational games, Haugeland has argued that people often try to solve a puzzle in order to “earn public recognition and esteem and thereby validate or augment [their] own self-esteem” (Haugeland, 1985, p 239). Thus, while the problem might not meet all of Rittel and Webber’s conditions, their sixth rule, i.e., ‘each attempt to solve the problem incurs unrecoverable costs’ appears to apply.

Given Crickmay and Jones’ characterisation of *Designing as a Response to Life as a Whole* (Crickmay and Jones, 1972), it is for designers and design researchers to conjecture whether, since the concept of wickedness is applicable to a range of activities which have so much in common with design, it might then also be applicable to many areas of design itself. Hence, when a designer embarks on a design project it can be worthwhile to review Rittel and Webber’s list in order to see if the project as a whole, or elements of the project, qualifies as a wicked problem.

For designers, recognising that a project is wicked has two major implications. First, they can not expect to be able to pre-plan a fixed design methodology. Second, they can not assume that there is a definitive way to frame their project. Instead, Rittel and Webber imply that their design methodology and their frame of reference must co-

evolve with their growing understanding of the project and its context. Rittel argued that the best responses to a wicked problem include collaboration and argumentation. Thus second generation methods should be invented which “should be based on a model of planning as an argumentative process in the course of which an image of the problem and the solution emerges gradually among the participants as a product of incessant judgement, subjected to critical argument.” (Rittel, Grant and Protzen, 1972, p 38).

In terms of the graphs which were the subject of section 3.1, Rittel’s prescription does not call for a mere sequence of iterations though the stages of a pre-defined method. Instead, Rittel is arguing for the simultaneous design of a product and the process of designing that product. In other words, second generation methods imply that designer and others such as customers and product-users become active participants in Jones’ process of *Designing Designing* (Jones, 1991).

One problem with second generation design methods is that they are not methods in the sense of being pre-packaged procedures that designers can expect to follow. Instead, they are spurs to wider activity to make designers think in terms of design as being a process that involves elements including, but not limited to the following: (i) problem identification; (ii) multi-disciplinary collaboration with professional designers, (iii) collaboration with diverse participants; (iv) recognition of changing market conditions and societal attitudes; (v) accommodation of new ideas; (vi) reflexive re-design of the design process.

Some artefacts and techniques have been developed to carry these messages to designers. As this subsection has shown these include flash-cards, random generation techniques, and checklists. However, by their nature second generation methods cannot be embodied in anything more than a temporary ‘resolution’ that should remain open to debate and re-evaluation.

In conclusion, perhaps the most important implication of the second generation is ethical in nature. While paid work on a particular design might cease at a given point in the development process—Hollins and Hollins (1991) suggest that ‘sales’ is a frequent stopping point for professional designers—an artefact or system which is a product of that activity might continue its function in an environment, or to affect that environment in some other way, long after this point. As Rittel and Webber noted, ‘the perceived quality of a resolution might increase or decrease over time’, ‘the problem can be symptomatic of worse problems that may not reveal themselves until it has been resolved’, and ‘a resolution has the potential to cause harm and, hence, problem solvers may owe responsibilities to themselves or others’. Although this is an ethical point, it seems to have been influential because consideration of what happens to a product once it has been sold, shipped and disposed of has become a key element of the methods and

design education of the 1980s and 1990s—see the ‘Total Design’ method (Hollins and Pugh, 1990; Hollins and Hollins, 1991; Pugh, 1991).

The spirit embodied in the work of such second generation methods continues to evolve in the 1990s in studies such as Latour’s *Aramis or the Love of Technology*, (1996). Latour’s work integrates first-hand and fictional accounts of the design of a product into a narrative featuring imaginary ‘interviews’ with the product, which recall Gordon’s *Synectics* (Gordon, 1961).

3.4.3 Implications of first and second generation design methods for knowledge aided design

The thesis will present several reasons why the type of design knowledge and design research that has led to the development of first and second generation design methods should be valuable to researchers and developers of knowledge aided design systems.

In Chapter Four, it will be argued that procedural design methods could provide a source of design knowledge that offers an alternative to Fischer’s concept of a DODE and allow for principled exploration of new areas of Green’s model-space. In particular, it will be argued that glass-box components of a prescriptive design method can be emulated by knowledge based components of a knowledge aided design system whereas black-box components can be supported by more human-driven components. Further, it will be argued that black-box components of a procedural design method are miniature versions of full-size wicked design projects. So the kinds of technologies that are aimed at supporting black-box methods should serve as miniature ‘testing-grounds’ for configurations of technologies that are intended to support full-sized wicked design projects.

Given the preceding arguments, developers of knowledge aided design systems to support black-box activities or full-size wicked design problems should expect designers to invent unforeseen ways of working with their products (Beardon, et al, 1995). This is because challenging established practice and use of materials is a hallmark of addressing wicked problems. Moreover, perhaps developers of knowledge aided design systems should take an ethical and long term view and ask the following questions. Should they consider the roles that their software will play once it has left the research institute? Will a system restrict designers to a given domain, to a given way of working or will it be sufficiently extendible to allow end-user programming and the expansion of its knowledge base? Thus second generation design methods could play an important role in helping to establish a Total Design ideology for knowledge aided design systems.

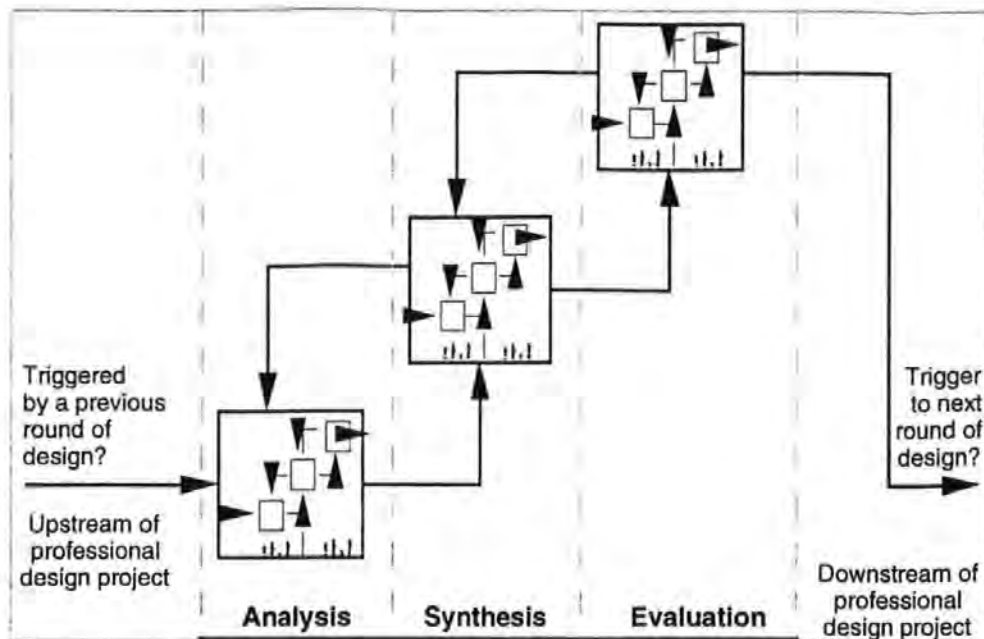


Figure 3.11: the generic three stage model of design. The domain of the axis is divided into three sections and these are shown along the bold section of the line at the bottom of the diagram.

3.5 Synthesising the design activity space

This section summarises the main sections of the chapter with the aid of a sequence of diagrams. These diagrams will also help to define the three dimensional space of design activities that was outlined in the introduction. Each diagram will depict the domain of its axis as a bold line in its lowermost area. This will be enhanced by contextual information, aimed at the developer of knowledge aided design systems, which will be suspended above the axis.

3.5.1 The first axis: a generic three stage model of design

The first axis describes the generic three stage model of design which was discussed in section 3.2. This axis is illustrated in figure 3.11. This figure contains the axis itself (shown by the line in the lowermost part of the diagram) and a flow chart of the stages of design (shown in the central area of the diagram).

The bold portion of the line in the lowermost area of figure 3.11 shows the three typical stages of the design activity: analysis, synthesis and evaluation. These stages are the domain of the first axis of the space. The lighter sections of this line, to the left and right of these main stages, illustrate the existence of two less commonly recognised stages of design. In these stages, people identify problems and encounter the results of a

designer's craft. These stages can be thought of as 'triggers' to the design activity. It is possible that they might not need to be explicitly modelled or supported by a knowledge based design system. However, second generation thinking implies that these can generate important inputs to the design process and that designers should consider how an environment will be affected by their artefacts and systems. So it seems prudent for system developers to deliver systems that exhibit some awareness of the existence of these upstream and downstream stages.

The flow chart of the design activity suspended above the main sections of the axis illustrates that each stage of design is likely to contain nested versions of the entire flow chart. This is included to remind system developers that a system which is aimed at a particular stage had best contain components to support each of the other stages. Similarly a system which supports the total design process should allow and encourage designers to move between its constituent stages at will.

3.5.2 The second axis: how much innovation do various design activities require?

The second axis describes a model of the amount of innovation required by five types of design activities. This axis is illustrated in figure 3.12. This contains the axis itself (shown by the bold line in the lowermost area of the diagram), some numerals which indicate Culverhouse's estimation of the percentage of new ideas which are required by each type of design (the central area of the diagram), and some lines that indicate which types of design activity are 'pulled by a problem' or 'pushed by a pre-existing solution' (the uppermost area of the diagram).

The bold portion of the line in the lowermost area of figure 3.12 shows the three main types of design activities that designers routinely face: variant-design (which is further divided into its configurative and true portions), innovative-design, and strategic-design. This is the range of the second axis of the space. The lighter sections of this line, to the left and right of these main stages, illustrate the existence of repeat orders and invention. It is probable that these will not need to be explicitly modelled or supported by a knowledge aided design system. This is because repeat orders do not constitute design and the process of pure invention is too far outside our current range of understanding for cognitive or other sciences to furnish the explicit knowledge that a knowledge aided design system would require. As Chapter Four will argue, the types of design in the bold section of the line will each need to be supported with different types of technology or hybrids of technologies.

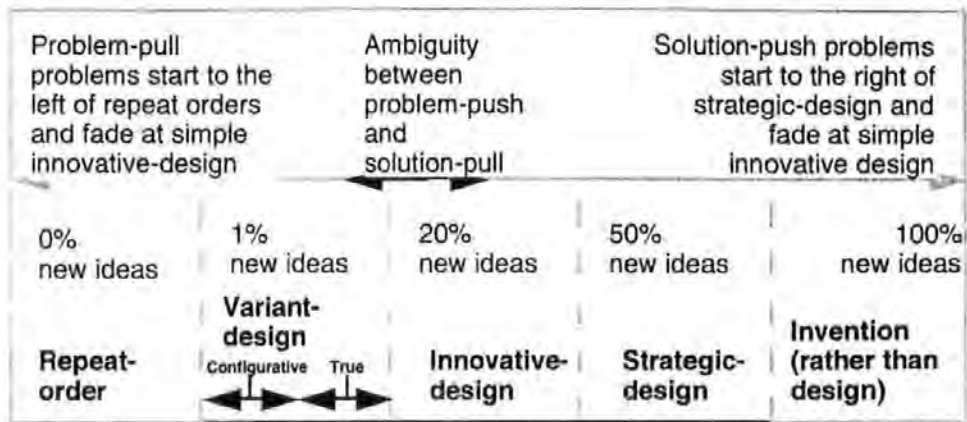


Figure 3.12: The model of the amount of innovation required by four types of design activities: repeat orders, variant-design, innovative-design, and strategic-design. The domain of the axis is shown by the bold line which is divided into the three main types of design. Above this the numerals show Culverhouse's estimation of the percentage of new ideas which are required by each type of design. The lines at the top of the diagram indicate which types of design activity are pulled by a problem or pushed by a pre-existing solution.

With this in mind, the central area of the diagram shows Culverhouse's estimation of the percentage of new ideas which are required by each type of design. This provides designers and developers of knowledge aided design systems with a heuristic with which to categorise a given design activity.

The uppermost lines in the diagram indicate the range of tasks that are pushed by a problem or pulled by a solution. It also indicates that there is some ambiguous overlap between these categorisations across variant-design and innovative-design. Designers sometimes need to switch the logical ordering of their methods according to whether a problem is pulled or pushed. So this distinction shows developers of knowledge aided systems that this need should be especially well supported at the cross over between variant-design and innovative-design.

3.5.3 The third axis: first generation and second generation design methods

The third axis describes the ratio between first generation and second generation design methods which are applied to a task. The axis is illustrated in figure 3.13. This contains: the axis itself (shown by the bold line in the lowermost area of the diagram); a visual representation of this ratio (shown by the two triangles in the central area of the diagram); and representations of the ways that first and second generation methods can be characterised (shown by the two rectangles in the uppermost area of the diagram).

The bold line in the lowermost area of figure 3.13 shows a way of characterising the methods with which designers address specific design activities using a ratio of first to second generation methods.

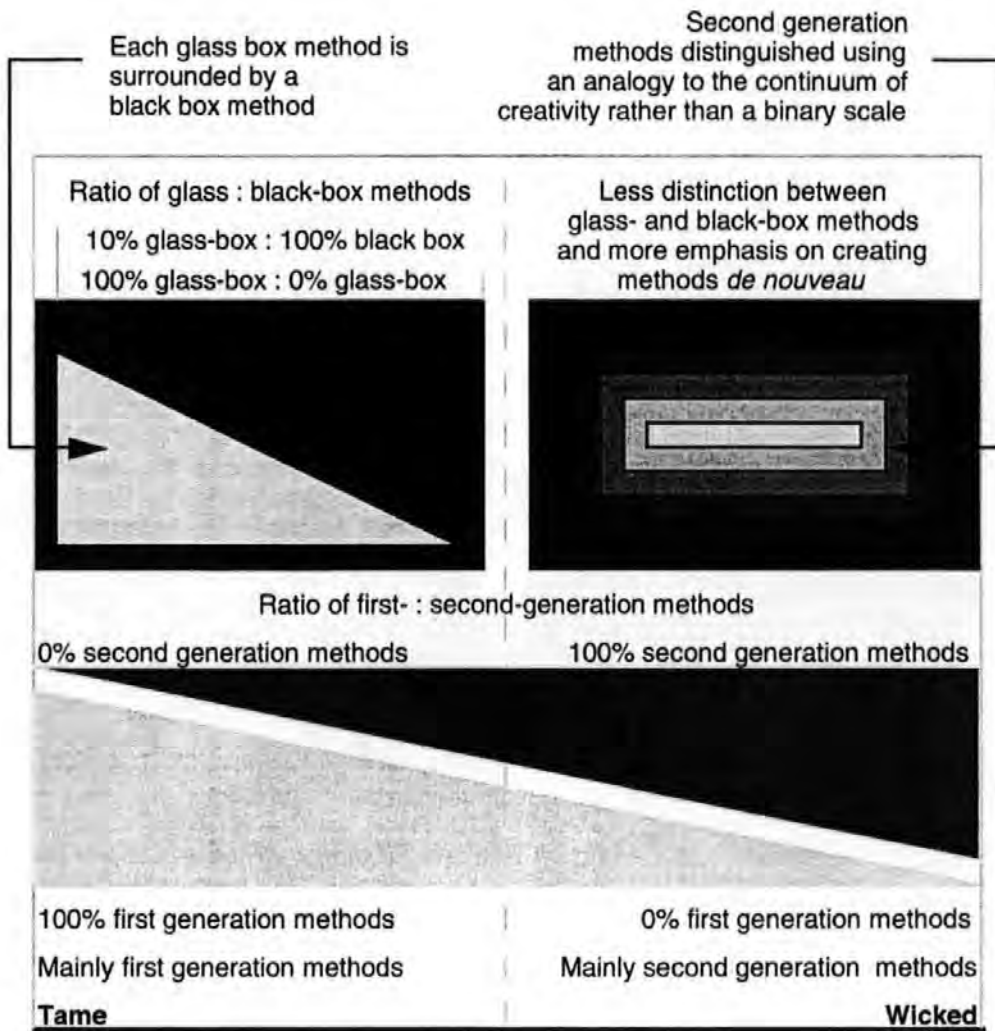


Figure 3.13: First generation and second generation design methods. The axis is divided by a notional equilibrium between these types of method. This is shown by the bold line in the lowermost area of the diagram. A visual representation of this ratio is shown by the two triangles in the central area of the diagram. The ways that first and second generation methods can be characterised are shown by the two rectangles in the uppermost area of the diagram.

The twin triangles directly above this line, in the central area of figure 3.13, depict this ratio graphically. The first generation methods are shown in a light grey triangle to indicate the existence of glass-box methods. The second generation methods are shown in a darker grey triangle to indicate an overwhelming presence of what the first generation of design researchers would have called black-box methods.

However, while this shading scheme captures a very high level distinction between the types of method, given the arguments presented in subsections 3.4.1 and 3.4.2, it seems a little coarse. Therefore the rectangles in the uppermost area of figure 3.13 are used to reveal more of the internal structure of the first and second generation methods.

The leftmost rectangle indicates that first generation design methods contain their own ratio of glass- to black-box methods and that glass-box methods are always embedded in a supporting structure of black-box methods. In Chapter Four, it will be argued that glass-box components of a first generation method can be emulated by knowledge based components of a knowledge aided design system whereas black-box components need to be supported by more human-driven components. If these arguments are accepted then this rectangle indicates that even knowledge based components need to be embedded in a human-driven environment.

The rightmost rectangle in figure 3.13 indicates that second generation design methods contain a version of the model of the amount of innovation required by design activities. Chapter Four will argue that black-box components of a procedural design method are miniature versions of full-size wicked design projects. So the kinds of technologies that are aimed at supporting black-box methods should serve as miniature 'testing-grounds' for configurations of technologies that are intended to support full-sized wicked design projects. It will also argue that different kinds of technology are appropriate for the variant-, innovative- and strategic-design. If this argument is accepted, then this diagram indicates that these choices of technologies should ideally be applicable to any parts of a program that represents a method as well as the part that stands for the artefact or system that is being constructed.

3.5.4 Some caveats

The complete space of design activities which has been analysed and synthesised in this chapter is illustrated in figure 3.14. Defining this space has primarily been an exercise in what Crickmay and Jones called 'solution-push' (1972). This is because the space aims to 'tame' some of the wickedness inherent in the task of identifying and characterising activities for knowledge aided design systems to address. However, it has also involved some elements of what Crickmay and Jones called 'problem-pull'. This is because, as Chapter Two showed, knowledge aided design has a history of developing systems that do not address the kinds of problems that designers face.

As Rittel and Webber's list of characterisation of wicked problems testifies, proposing such a space to absorb some wickedness from the domain of knowledge aided design system's development is a wicked activity in itself. For example, Rittel and Webber's list contains the following warnings: wicked problems have no definitive description; problem solvers cannot tell when a wicked problem is solved; and the perceived quality of a resolution might increase or decrease over time. It is therefore

worth stressing that the current space *must* be proposed as a resolution to the current problem rather than advanced as a definitive solution.

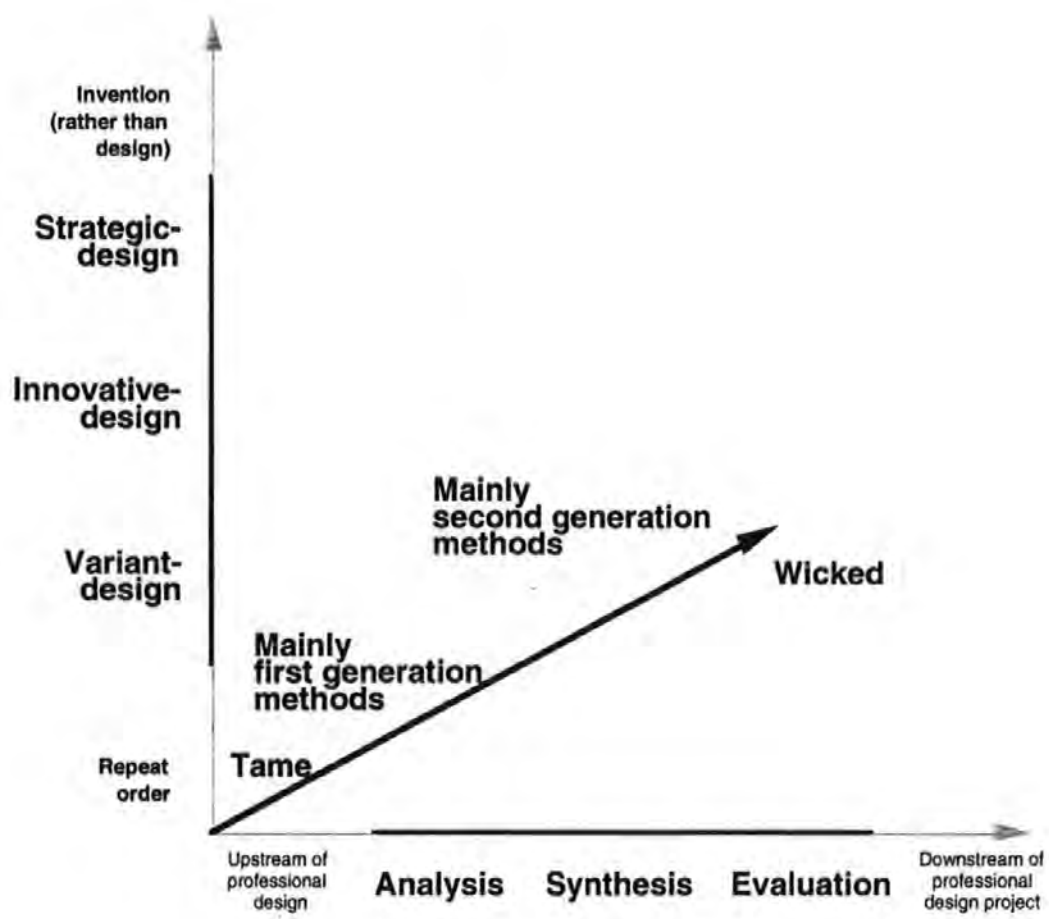


Figure 3.14: The three dimensional space of design activities which has been defined in this chapter.

With the caveat in place that future design research may well generate new knowledge that could supersede the research which grounds this space, it will be used to make two contributions to the remainder of the thesis: in Chapter Four, it will be mapped onto Green's space of knowledge aided design technologies; in Chapter Five, it will provide a framework for the analysis and synthesis of technologies to support a particular Procedural design method with a knowledge aided design system.

For now, then, the space of design activities will contain: generic stages of design projects, i.e. triggers, analysis, synthesis and evaluation; a description of the amount of innovation required by a project; and a description of the kinds of methodologies used in a project. Aspects of design that could be included in future versions include:

- (1) the differences between design-in-the-small and design-in-the-large—these include the need to support 'team design', the need to support integration of new

team members during a project and the need to maintain an accurate shared representation of the product and its design process during a project

- (2) the differences between short and long term projects: there is a need for investigations and characterisations of projects that take weeks, months and years to complete.

The desirability of adding these aspects of design to a design activity space will be returned to in the agenda for future research, which is set in the concluding chapter of the thesis.

3.6 Summary

This chapter has defined a design activity space in response to the first of the hypotheses reported in Chapter One. The space encapsulates knowledge about: generic stages of design projects, i.e. triggers, analysis, synthesis and evaluation; a description of the amount of innovation required by a project; and the kinds of methodologies used in a project. The space's forerunners are the research of Brown and Chandrasekaran (1989), Chandrasekaran (1992), and Goel and Pirolli (1992). However it was claimed that this model is more detailed than those of Brown and Chandrasekaran, it serves a very different purpose from that of Goel and Pirolli's model and its content also differs from that of Goel and Pirolli's model.

The design activity space will contribute to the work reported in subsequent chapters in the following ways:

- In Chapter Four the main task is to couple the design activity space with Green's (1992) model space for categorising tools and systems for knowledge aided design. This coupling will enable the hypothesis that it is possible to develop mappings between this design activity space and Green's model space of knowledge aided design systems to be addressed.
- In Chapter Five the main theoretical results of the research are described and the preliminary mappings between the design activity space and Green's model space are explored to address the third hypothesis, i.e., it is possible to search the design activity space for at least one niche that is unsupported by existing classes of

system and to propose a new class of system, composed of tools located in Green's model space, that appears suited to supporting that niche.

- In Chapter Six the design activity space and the mappings between it and Green's model space are used to help derive the design of Patina. This addresses the fourth hypothesis: is possible to design and implement an example of this new class of system by locating a design practice in the design activity space and then applying the mapping to Green's model space in order to propose an aggregate of components that will support this practice.

Chapter Four

Mapping the space of design activities onto a set of knowledge aided design tools and a model space of knowledge aided design systems

4.0 Overview

The work reported in this chapter addresses the first half of the second of the hypotheses described in the introduction:

H2 It is possible to develop mappings between the design activity space and Green's model space of knowledge aided design systems. These mappings should enable the approximate range of applicability of individual tools and classes of systems to be indicated. These mappings should also make it possible to re-characterise the accepted range of applicability for some existing classes of system.

The chapter begins by describing a set of tools for knowledge aided design and then describes and discusses a model space which categorises such tools along seven dimensions. These were both devised by Green (1992a) who used them to support a clarion call for the development of new generations of knowledge aided design systems. The second half of the chapter maps the space of design activities with Green's space of knowledge aided design technologies and the content of his tool kit. The chapter concludes with a summary of the issues that have been raised. It also prepares the way for an investigation of the territory between DODEs and process based design environments, to be presented in Chapter Five, which will identify a niche for the new class of MODEs.

4.1 Green's tools and technologies for knowledge aided design

In Chapter Two, Green's *Conceptions and Misconceptions of Knowledge Aided Design* (Green, 1992a) was cited as being a key 'paper in the evolution of knowledge aided design. Green's paper made three contributions to this subject area:

- it highlighted the fact that designers bring contextual knowledge to the kind of design activities that members of the knowledge aided design community had previously regarded as entirely routine

- it named a number of new types of knowledge aided design technologies
- it also defined a model space which Green used to indicate niches for new technologies and styles of interaction that could be incorporated into knowledge aided design systems.

The second and third of these contributions will be reviewed in this section. The section will begin by introducing and expanding Green’s tool set for new types of components for knowledge aided design systems. Green’s list of tools will be expanded by adding a ‘design-visualiser’, which should help designers to explore and interpret abstract numeric data relating to an artefact that is being designed.

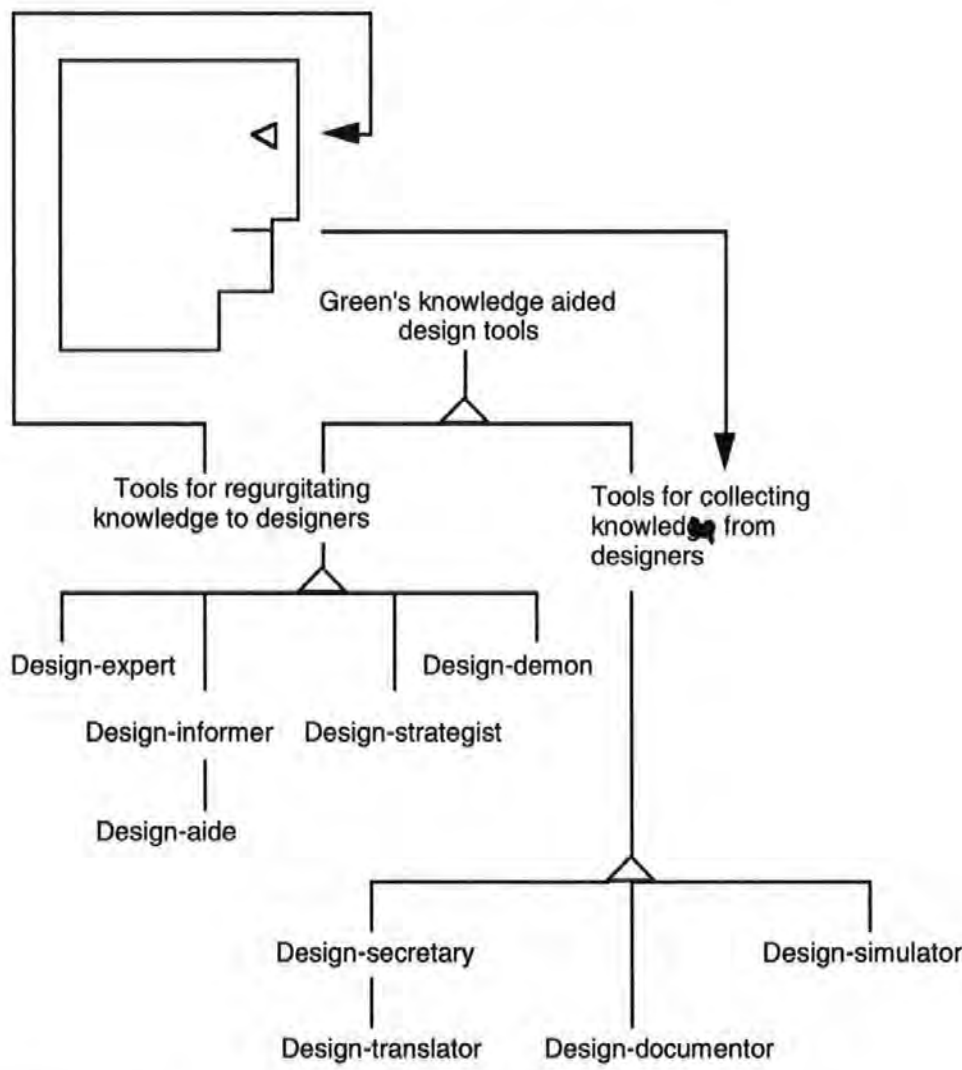


Figure 4.1: A hierarchy showing Green’s two types of knowledge aided design tools.

4.1.1 Green's tools for knowledge aided design

Green introduced nine types of tool for knowledge aided design. In the terminology reviewed in Chapter Three, Green's tool set can be seen as an example of 'solution-push' problem solving (Crickmay and Jones, 1972). This is because Green prefabricated and catalogued a set of standard parts in order to reduce the wickedness of this genre of problem. The advantage for system developers of adopting Green's tool kit is that it provides them with components whose functions are easy to conceptualise and which are easy to describe to one another. The potential disadvantages of adopting such components include a loss of the flexibility which would be associated with bespoke tools and a concomitant loss of the opportunity for radical thinking at the outset of a project.

Green divided these tools into two groups (figure 4.1). These groups were respectively intended to present design knowledge to designers and to collect such knowledge from them. Since both of these groups of tools are key to the subsequent work to be reported they will be described here in detail.

4.1.2 Green's tools for presenting design knowledge

The first group of Green's tools inherited many of their component technologies and their role in knowledge aided design from expert systems. These tools were intended to present design knowledge to designers at their request. The nature of such presented knowledge could range from a newly generated plan of how to approach a design activity to a finished design for an artefact (or system / service) that would meet the demands of that task. Since these tools inherited their functionality from expert systems, and since they were intended to follow pre-existing practice, they can be categorised as place holders for programs based on glass-box design methods and on fairly tame problem solving techniques. The specific tools in this first group included: design-experts, design-aides and design-informers, design-strategists, and design-demons.

Green envisaged design-experts as straightforward implementations of expert systems technology, which happened to have been targeted at a design domain. Green commented that: "such a system might be feasible when context is less important. It is possible that, if the domain is nearly decomposable, routine parts of the design might be completely automated." (Green, 1992a, p 19). The arguments presented in Chapter Three suggest that such a tool would rely on the existence of a glass-box design method and that it is unlikely that such a tool could be used independently of a supporting black-box method.

Green's design-aides and design-informers are closely related. While a design-aide would contain similar technology to a design-expert, it would not attempt to design an artefact for a designer. Instead it would suggest alternative tools or techniques that a designer might choose to employ during the design process. A design-informer would exhibit similar functionality and work from a similar knowledge-base. However, it would not contain an inference engine that would respond to prompts from a designer. Instead it would present a repository of useful guidelines that the designer could chose to browse through in search of inspiration. From the context of Green's discussion it appears that both of these types of tool were envisioned as offering symbolic, text-based interfaces to their users.

Green's design-strategists are similar to design-experts except that they are intended to apply expert systems technologies to the strategic part of design rather than to the artefact that is being designed. It is hard to find a definitive way to distinguish these tools from design-aids and design-informers but it seems implicit in Green's discussion that design-strategists are concerned with macro strategies while design-aids and design-informers are concerned with micro strategies. Such a tool would rely on the existence of a glass-box design method but in this case it would need to be a glass-box design method for designing the process of designing. This restricts much of the applicability of design-strategists to the tame end of the axes of the space of design activities concerned with innovation and wickedness. Moreover, from the context of Green's discussion it appears that a design-strategist was also envisioned as presenting symbolic, text-based interfaces to the user.

Green's design-demons would act rather like the critics which were described in Chapter Two. They would be embedded in a larger system where they would attempt to 'shadow' a designer's decisions in order to warn of obvious constraint violations. A design-demon is slightly easier to implement than a design-expert because it does not need to make its own decisions about which paths to explore. However, the problem of inferring a designer's intentions from their actions is a very pressing research issue for such tools. See Silverman (1992) for an extended discussion. One of Silverman's main prescriptions for the kind of critics that are synonymous with design-demons is that they should be embedded in graphical user interfaces. However, designers of such systems leave too many opportunities open for system crashes to be caused by unpredictable though sane actions on the part of these users. Hence the very technologies which are employed in the design of graphical user interfaces make it hard to make inferences about complex intentions on the part of the user because the opportunities for expressing these intentions are so minimal.

4.1.3 Green's tools for collecting design knowledge

Most of the second group of Green's tools were more innovative than those in the first group. This is because, as Green argued, they were: "as much concerned about collecting, communicating and storing expertise as about regurgitating it on demand" (Green, 1992a, p 19). The tools in this group include: design-secretaries, design-translators, design-documentors and design-simulators.

A design-secretary would represent the evolving states of a large design in a format that would allow collaborating designers to track the 'global situation'. It is possible that a design-secretary could represent natural language descriptions of design decisions, graphical images or working drawings, or some combination of such representations of knowledge. Green argued that a design-secretary might help to translate between the design-languages of various experts in a multi-disciplinary team. This would be a valuable role for a program to be able to play. However, as the arguments in Chapter Two stressed, Bond (1992) has found that experts within a particular discipline have 'private justification languages', which *even they* find hard to translate. It is difficult to imagine that a program could perform such a task on behalf of a multi-disciplinary team, particularly if they are inventing new concepts as part of innovative-design or strategic-design.

A design-translator would be a more specialised version of the design-secretary which would take on even more of this inter-lingual role. Again, as Green argued, "it would be valuable to create a design representation which could be viewed from different perspectives before the design is complete" (Green, 1992a, p 20). However, if it is hard to imagine a stronger version of the design-secretary, it is also hard to imagine that such a language would offer the freedom of multiple representations which diverse and field-specific design languages bring.

A design-documenter would be embedded in a larger system where it would prompt designers to record the reasoning behind their design decisions. Such a system synthesises Issue Based Information Systems (IBISs), which record debates between people in the graphical format of a web or decision tree, with Fischer's paradigm for 'seeding and evolutionary growth' (Fischer *et al*, 1996). Green advocated embedding such design-documenters in graphic simulations of artefacts so that designers could investigate how a particular item came to achieve a current state. In principle, this approach could be applied to new generations of graphically rich design-aids, design-informers and design-strategists so that designers could investigate how a particular strategy evolved—Blessing's proposal for the PROSUS system is a good example of how such tools could be used in the context of wicked design (Blessing, 1994).

Design-simulators allow designers to create abstract representations of artefacts which do not yet exist outside of the combination of the system and their imaginations. These representations are useful because they obey whatever physical or logical constraints the designers choose to include within their model of a domain. A design-simulator can therefore reveal that a designer has broken or ignored a constraint before the costs of debugging this error are encountered in real world prototypes or products.

However, while these tools can be beneficial, particularly when safety is at issue, it is problematic for designers to predict which constraints need to be included within their domain model. In fact the process of inventing a simulation is wicked. This is because a designer can accidentally omit a constraint that would have drastically affected the outcome of a simulation or the fate of a resulting product in the real world. With this problem in mind, system developers will sometimes implement a design-simulator, which includes a domain model, and present the tool to designers working in that domain as a complete package. In such cases the wickedness involved in building the domain model passes upstream from the designer to the system developer.

Design-simulators can be divided into low-end and high-end models. Low-end design-simulators tend to have two dimensional interfaces and to support very simplified animation. Such tools represent objects in their target domain in the stylised and iconic fashion. The interfaces to high-end design-simulators tend to simulate a third dimension and to support photo-realistic animation. Such design-simulators therefore overlap with the subject area of virtual reality. During the later 1990s usage of such high-end simulation tools within the design community is mostly found in architecture (see Mitchell, 1995) and in the European automotive and aerospace industries (see Page, 1997). For various technical and social reasons virtual reality "is not yet accepted as part of the design loop" (Radford quoted in Page, 1997, p 32IS 11).

4.1.4 Adding a design-visualiser to Green's tool set

Although the scope of Green's two groups of tools is quite comprehensive, neither group includes a tool that acknowledges the importance of statistical diagrams and their exploration in the early stages of design. Moreover, from a technological perspective, they omit to mention technologies for scientific visualisation which can support the exploration of such diagrams (c.f., Brodlie *et al*, 1991). Scientific visualisation has been defined as a way of "exploring data and information graphically as a means of gaining understanding and insight" (Earnshaw and Wiseman, 1992, p 6). In this tradition computers are used to present representations of multi-variate data that can be rapidly updated to reveal different aspects of a data set, different sections of large data sets, or

different representational formats for a data set—see Hofland and Utsler (1997) for some illustrations of ‘intuitive’ interfaces to commercial programs for visualisation and data mining.

There are two reasons why the omission of tools to support such visualisation is unfortunate. First, scientific visualisation is a mature subject area which has delivered many widely-used programs. Second, while objective science differs from design (deVries *et al*, 1993), techniques for scientific visualisation have been adapted from scientific domains to meet the demands of designers from a wide range of arts and industries. In light of these observations, this subsection will document the evolution of statistical diagrams into interfaces for scientific visualisation programs and advocate expanding Green’s tool set by adding a ‘design-visualiser’.

Statisticians have long regarded statistical diagrams as an: “extremely useful technique and most would advocate [their examination] as an essential prerequisite to more formal analyses” (Fisher, 1936, quoted in Everitt, 1978, p 1). However, statisticians also advise that judgements should not be based solely on the inspection of statistical diagrams: “diagrams prove nothing, but bring outstanding features readily to the eye; they are therefore no substitute for such critical tests as may be applied to data, but are valuable in suggesting such tests, and in explaining conclusions founded upon them” (Fisher *Op Cite* in Everitt *Op Cite* p 1). Hence, the exploration of statistical diagrams can be equated with a black-box method in design, which sets the context for the application of glass-box methods for formal investigation and subsequent black-box methods for interpretation.

Engineering psychologists have investigated the relationship between the perception and interpretation of a variety of types of statistical diagrams (Wickens, 1992). The results of their experiments show that people are liable to misperceive, and hence misinterpret, information that is presented in an inappropriate statistical diagram. For example, Wickens’ review of such experiments concludes that “not all graphs are equal, and some serve certain tasks better than others” (Wickens *Op Cite* p 117). This problem has been compounded by the development of computer programs that present statistical diagrams in a generic fashion that does not meet the specific requirements of such users as engineers or designers (Petroski, 1995).

In response to such observations, researchers from such areas as data-graphics, cognitive psychology, and information design have begun to investigate ways of presenting diagrams that accommodate the theories of graphical design and cognitive ergonomics (Tufte, 1983, 1992; Cleveland, 1994; Colman, 1994; Larkin and Simon, 1987; Lowe, 1993). The premise for this cross-disciplinary research in the emerging subject area of external representations has been summarised by Lowe: “the application

of established principles of graphic design is a vital part of developing effective diagrams, but tends to focus upon external aspects of representation ... however, the internal (mental) representation of a specific set of subject matter is important in influencing what sense viewers make of a diagram." (Lowe, 1993, p 3).

During the 1980s and 1990s, advances in the processing speeds and graphical displays of personal computers and workstations spurred the development of an array of specialised visualisation programs. Some of the computer scientists who produced these programs began to adopt some of the principles and formats for data display produced by researchers in the subject area of external representations. These developers embedded these formats in graphical user interfaces which allowed people to examine clear and responsive illustrations of multi-variate data that were targeted at the specific needs of various professions (Earnshaw and Wiseman, 1992; Hofland and Utsler, 1997).

However, while the process of matching commercial software to the levels of graphical awareness and the needs of its customers is well under way, observers such as Petroski (1995) argue that more attention still needs to be paid to these aspects of constructing such software. In short, the wide range of available products testifies that the technology for scientific visualisation is in place, but an infusion of techniques from the worlds of knowledge engineering and user centred design still appears to be needed in the later 1990s.

Although this subject area is called *scientific* visualisation, scientists are far from the only users of this technology. As Clark has observed: scientific visualisation is of value beyond strictly scientific applications ... the same technology is now used in such diverse applications as clothing design, industrial design, automobile design, aeroplane design, chemical and drug design, and chemical and nuclear power plant design..." (foreword to Earnshaw and Wiseman, 1992, p iv).

So a wide range of designers are already familiar with visualisation software and some effort has been invested in bringing design expertise to bear upon its construction.

This discussion has demonstrated that scientific visualisation is a mature technology which has been adapted from the demands set by scientists towards those set by designers. It is important to note that this technology has the capacity to bridge Green's groups of tools for the regurgitation and collection of design knowledge. This is because the interfaces to these tools tend to be designed to encourage the exploration *and* interpretation of data. It therefore seems appropriate to add a design-visualiser to Green's tool set.

A design-visualiser can be defined as a visualisation tool whose functionality differs from computer aided drafting, as well as from certain design-simulators, because it is not concerned with helping designers to visualise the appearance of the work item itself.

Instead such a tool would help designers to explore features of an artefact, or class of artefacts, that are contained within numerical data about a product and its market place. An example of a design-visualiser can be found in the interface to the system that will be described in the second half of this thesis.

Now that Green's tools for knowledge aided design have been introduced and expanded, they will be used to illustrate the ranges of the dimensions in Green's model space of technologies for knowledge aided design.

4.2 Green's model space of technologies for knowledge aided design

Green's model space of technologies for knowledge aided design contains seven dimensions. Each of these will be illustrated to show the range of known ways that people can interact with knowledge based or knowledge aided technologies. These dimensions measure: the way that a tool communicates with its user; the amount of autonomy associated with a tool; the depth of knowledge found in a tool; whether a tool is designed for individual or shared use; the degree to which a tool presents or collects knowledge; the degree to which a tool enhances or replaces a designer; and the degree to which explanations offered by a system were calculated in terms of syntax, semantics or pragmatics. Each of these axes will now be discussed in detail.

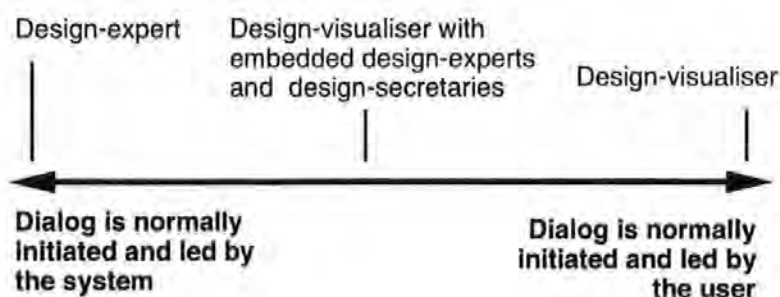


Figure 4.2: The first dimension in Green's model space measures the way that a dialogue is initiated between a tool and its user(s).

4.2.1 The first dimension of Green's model space

The first dimension of Green's model space measures the way that a tool initiates or responds to requests for dialogues with its user¹ (figure 4.2). At the leftmost extreme of the dimension the dialogue is initiated and led by the tool and at the rightmost extreme it

¹For simplicity it will be assumed that single user systems are under discussion unless the axis in question is concerned with individual-shared use.

is normally initiated and led by the user. A design-expert would appear on the leftmost position on this dimension. This is because such a tool would typically prompt its user for an initial description of the problem at hand (this is one of the black-box component of the activity), then prompt the user for more information as and when this is required. A data-visualiser would appear at the far right as such a tool generally waits passively until a designer commanded it to perform some calculations and subsequent data presentation. A hybrid of these tools, which might also include some of Green's tools for knowledge collection would appear in the centre of the dimension.

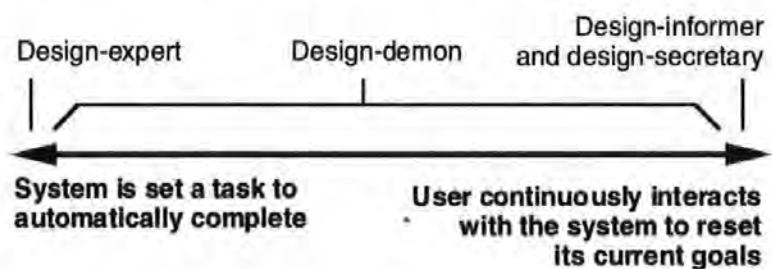


Figure 4.3: The second dimension in Green's model space measures the amount of autonomy found in a tool.

4.2.2 The second dimension of Green's model space

The second dimension of Green's model space measures the amount of autonomy found in a tool (figure 4.3). This dimension is very much related to the first dimension. One difference between these axes can be found in the granularity of the interaction between the user and the tool. At the leftmost extreme of the dimension, a design-expert might be set a task which it would automatically complete. At the rightmost extreme a user might continuously interact with a design-secretary to change its current goals and to supply the tool with new contextual information. A design-demon might intervene at various points along the range of the dimension depending upon the instructions that were given to a design-expert or the nature of the interaction between a designer and other tools.

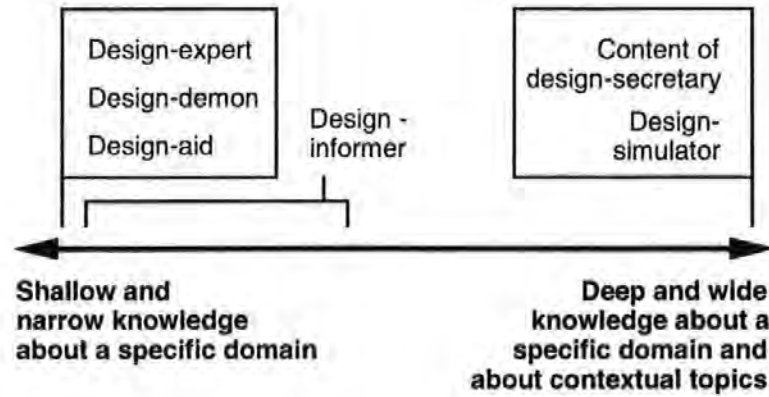


Figure 4.4: The third dimension in Green's model space measures the depth of knowledge found in a tool.

4.2.3 The third dimension of Green's model space

The third dimension of Green's model space measures the depth of knowledge found in a tool (figure 4.4). A shallow tool would have access to detailed knowledge or heuristics about a particular domain. However it would lack the context found in a deeper tool. A design-expert would typically have access to a fairly shallow and narrow channel of knowledge about a given topic (see Dreyfus, 1992). Design-demons and design-aids are based on similar technologies. So they are likely to occupy a similar position on the dimension. Depending on how thoroughly a design-informer is implemented, it might contain slightly deeper knowledge than a design-aid. This knowledge might also be slightly wider if it encompasses the design domain as well as suitable strategies. Thus a design-informer *might* occupy a more central position on the dimension. The content of a design-secretary is likely to appear quite deep to the people who recorded this knowledge. So the content, rather than the shell itself, would appear toward the right of the dimension. A design-simulator contains very deep, if narrow, knowledge. So it would appear on the extreme right of the dimension.

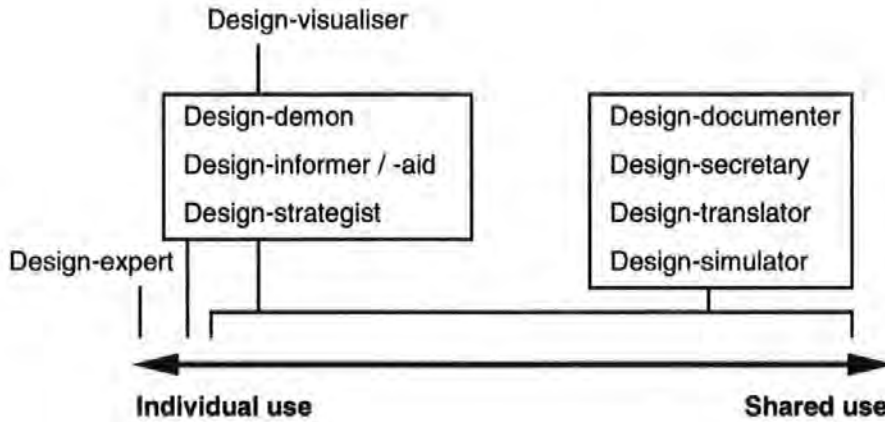


Figure 4.5: The fourth dimension in Green's model space measures whether a tool is designed for individual or shared use.

4.2.4 The fourth dimension of Green's model space

The fourth dimension of Green's model space measures whether a tool is designed for individual or shared use (figure 4.5). At the leftmost extreme of this dimension lie traditional design-expert systems and tools that interact with a single user. At the rightmost extreme lie tools developed by the computer supported collaborative work community. The tools have been illustrated here in a way that measures their traditional usage. The design-expert, for example, is shown at the leftmost of the scale because this is the standard mode of interaction with an expert-system. The other members of Green's first list of tools are shown slightly to the right of the design-expert to show that, while they are traditionally associated with individual use, they could each be adapted toward shared use. The design-visualiser is shown at the left of the figure, above the individual use tools, but it is linked to a range of the shared use tools. This is intended to show that statistical diagrams offer a method of representing data that is appropriate for both types of interaction. The list of shared use tools are illustrated at the right of the figure to indicate that this was the area where Green saw their greatest applicability. However, there is no reason to suppose that a design-secretary, for example, could not help an individual designer to record design-decisions in a way that would help them to understand their own motivations at a later date. Hence, the potential range of these tools is shown as extending from individual to shared use.

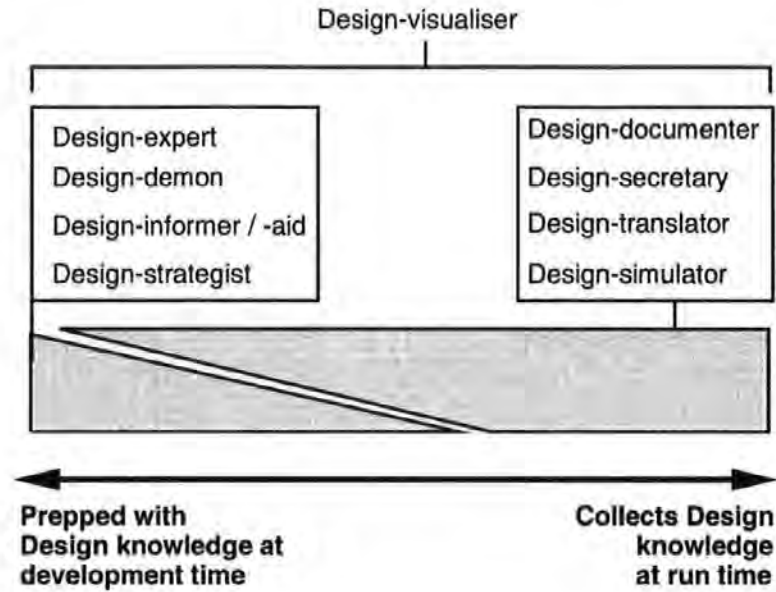


Figure 4.6: The fifth dimension in Green's model space measures the degree to which a tool presents or collects knowledge.

4.2.5 The fifth dimension of Green's model space

The fifth dimension of Green's model space measures the degree to which a tool presents knowledge to users or collects it from them (figure 4.6). At the leftmost extreme of the dimension the tool presents knowledge with which it has been prepped by knowledge engineers at its development time. At the rightmost extreme the tool collects knowledge from designers at run time. Since this is the main distinction which Green draws between his two lists of tools it is unsurprising that the first list clusters at the left of figure 4.6 and that the second list clusters at the right. However, that model seems a little coarse. Hence the two triangles indicate that a design-expert, for example, could contain a knowledge-acquisition module and a design-secretary or design-translator might well contain a set of domain-specific knowledge. The design-visualiser would be prepped with domain specific knowledge. This would be enhanced through continual usage and the representations and perceptions of the underlying model were continuously updated through interaction with designers.

4.2.6 The sixth dimension of Green's model space

The sixth dimension of Green's model space measures the degree to which a tool enhances or replaces a designer (figure 4.7). The design-expert appears at the leftmost point in the dimension. This is because it is a descendent of expert systems based

technologies and is hence intended to serve as an invention factory. Ideally, such a tool transforms specifications into designs for new products. Other members of Green's first list of tools appear slightly to the right of the design-expert. This is because, while they are also descendants of expert systems, each of them has been slightly adapted and is intended to help the designer to take decisions at micro-level within a full-size design activity. Both the design-visualiser and Green's second list of tools appear at the other extreme of the dimension. These tools require designers and ideally help them to build and share an understanding of the artefact that they are designing or of the methodology that they are employing.

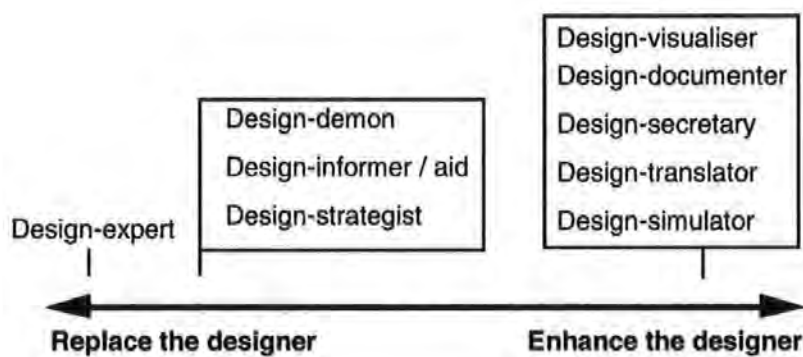


Figure 4.7: The sixth dimension in Green's model space measures the degree to which a tool enhances or replaces a designer.

4.2.7 The seventh dimension of Green's model space

The final dimension of Green's model space measures the degree to which explanations offered by a tool are calculated in terms of syntax, semantics or pragmatics (figure 4.8) . A syntactic decision might be expressed in terms of physical distance. A semantic one would contain variables that were represented by meaningful words. A pragmatic decision might involve judgements about political or aesthetic aspects of a product in a given cultural context.

Figure 4.8 shows a hierarchy of tools that apply to the sections of this dimension. A design-visualiser is restricted to the numerical section as it is mainly intended to show numeric relationships between aspects of a domain. Design-experts can be used to perform numeric calculations. However, they are mostly a semantic class of tool because they deal with textual representations which are understood by the user rather than the tool. This accounts for their limited presence on the pragmatic section of the scale. Similar arguments apply to design-informers and design aides, although these tools are less likely to deal with mathematical knowledge. In an ideal situation design-

documentors and design-secretaries would be able to deal with numeric, semantic and pragmatic knowledge. At present, however, the pragmatic section of the dimension is a distant goal for developers of these tools.

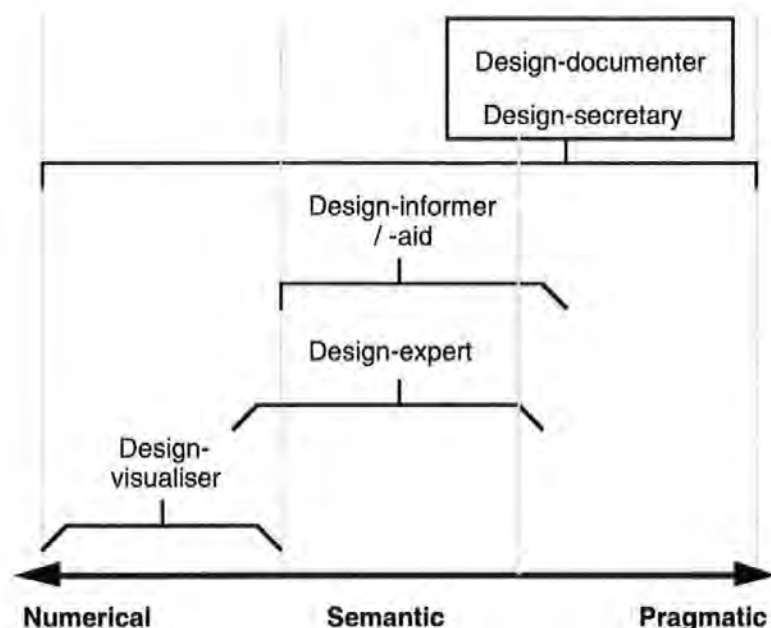


Figure 4.8: The seventh dimension in Green's model space represents the degree to which explanations offered by a tool were calculated in terms of syntax, semantics or pragmatics.

4.2.8 Implications of Green's model space for developers of knowledge aided design systems

In the early 1990s Green used this space to illustrate the observation that most of the systems which had been built up to that point were composed of expert systems based tools. Hence these systems clustered in the left side region of each dimension of the space, which is where one would expect to find the dominant expert system based technologies. This left a considerable volume of the space largely unexplored.

So Green's space is particularly valuable to developers of knowledge aided design systems because it allows them to visualise the fact that there is a huge space of tools, and combinations of tools, which remains available for exploration. It might seem even more useful to these developers because it is grounded in the discourse and familiar technologies of knowledge aided design systems development. However, this means that the space is more suited to describing styles of interaction between users and

programs than between designers and design activities. This criticism suggests coupling Green's space with the space of design activities proposed in Chapter Three.

4.3 Mapping the space of design activities onto the space of technologies for knowledge aided design

This section will begin the process of mapping the areas of the space of design activities onto the space of knowledge aided design technologies. There are two reasons why this process can only be initiated rather than completed. One of these is contingent on the history of knowledge aided design and the other is dependent on the concept of wickedness. The historically contingent reason is that, in the later 1990s, many of the newer tools in Green's catalogue have either not been implemented or have not been implemented a sufficient number of times for empirical data to be available. The other reason is that it is not possible to tell whether the current tool set is incorrect or incomplete. Green's tools can be seen as examples of the kind of design patterns that Alexander became concerned with identifying and cataloguing during the 1970s (Alexander *et al*, 1977). So it can be argued that these patterns afford similar functionality to those which Alexander catalogued by studying the domain of folk-architecture. However, since the store of folk-wisdom available in the field of knowledge aided design is not yet as abundant as that found in architecture, it would be prudent to consider the current crop of tools for knowledge aided design as a useful yet currently incomplete catalogue. Due to these reasons many of the arguments that can currently be provided must rest upon deduction from the literature of design research and artificial intelligence rather than upon empirical data.

This strategy seems necessary in order that temporary 'scaffolds' can be erected to help developers implement systems that will eventually generate such empirical data. Historically, this pattern is quite typical within the subject area of artificial intelligence. For example, Chapman (1987) observes that artificial intelligence research follows a pattern of 'neat' and 'scruffy' cycles. In a 'scruffy' cycle, broad guidelines for system development are suggested. In the 'neat' cycle these guidelines gradually become formalised as systems are field tested. Thus a scaffold might help developers to implement systems whose evaluations will cause it to be replaced by an improved scaffold.

With this caveat in place the remainder of this section will be devoted to establishing an analytic coupling between the design activity space and Green's model space.

4.3.1 An overview of tools for the first axis: a generic three stage model of design

There are two very general points which should be made about this axis before its constituent stages are mapped against the dimensions of Green's model space. The first point applies to the entire axis representing the traditional stages of design. The arguments presented in Chapter Three suggest that each of the main stages of design which are represented on this axis are likely to contain nested versions of the entire axis. This implies that a tool which is aimed at a particular stage in the design process had best either contain components to support each of the other stages or allow designers to move between these stages at will. The second point refers to the triggers which surround the central stages of analysis, synthesis and evaluation. These triggers lie beyond the current scope of the model being developed in this thesis. This is because these stages are likely to occur outside the formal components of design activities that can be defined with sufficient accuracy to be amenable to dedicated computer support. Unless and until design research can offer more formal descriptions of these stages it would seem premature for developers of knowledge aided design systems to address them.

4.3.2 Tools for supporting the stage of analysis

Since the purpose of the stage of analysis is to understand a problem it is unlikely that tools which initiate a dialogue will be required. The exceptions to this rule include systems that are intended for educational use and systems for which the intended domain is a paradigm example of a glass-box. This implies that tools ranging from a design-visualiser with embedded design-experts and design-secretaries to a stand alone design-visualiser would probably be appropriate.

For these reasons it also appears that tools which are set tasks to complete automatically would generally be less useful than tools for which a user is continuously able to reset the current goals. This implies that a combination of design-demons, design-informers and design-secretaries would be beneficial. An exception to this reasoning is that small scale design-experts could be employed to perform explicit and finite knowledge regurgitation tasks or to assist with the syntactic evaluation of a partial analysis.

Subject to the breadth of the domain being analysed it appears that tools providing deep and wide knowledge about the domain would be preferred at the beginning of the analysis. Subsequently, more narrow tools might reasonably come into play as specific features of the analysis begin to be evaluated. This implies that design-simulators and

design-secretaries would be useful at the beginning of an analysis and that the range of expert systems based tools and design-visualisers could be employed at its later stages.

The preference for tools with interfaces for individual and shared use would largely depend on the size of the design team performing the analysis. This dimension of Green's space is therefore rather orthogonal to this axis of the design activity space. The exception to this reasoning, which applies throughout this axis of the design activity space, is that design activities will typically begin and end with processes in which designers share information with one another. Hence tools for shared use and computer mediated communication are likely to 'frame' whichever other tools are used in an activity.

The process of analysis requires the assimilation, accommodation and eventual sharing of knowledge. It is therefore quite likely that tools from both groups in Green's tool set would be beneficial during these stages. In many cases developers of knowledge aided design systems have grouped such tools around a design-simulator which acts as a central work item. An alternative scenario would place a design-visualiser in a similar role.

In all but the most trivial domains, analysis is concerned with building an understanding of a problem. Various arguments in the artificial intelligence literature indicate that knowledge aided design tools would not currently build understandings in the same sense or to the same depth as humans. It therefore seems clear that tools that enhance the designers would be preferable to tools that attempt to replace them. This implies that design-visualisers, design-documentors, design-secretaries and design-simulators would be the most preferable tools from the perspective of this dimension of Green's space.

The final dimension of Green's space measures the numeric, semantic and pragmatic knowledge that a tool can collect or present. Each of these types of knowledge is important in most cases of design. However, tools which can accommodate pragmatic knowledge have yet to be developed whereas tools which allow the designer to distribute such knowledge are very feasible. It therefore appears that design-visualisers, design-experts, design-aids as well as rudimentary design-documentors and design-secretaries would be appropriate.

4.3.3 Tools for supporting the stage of synthesis

If designers are using an expert system based tool with a highly structured interface at the stage of synthesis, then the dialogue between the designer and the system could sway slightly in favour of system-initiated dialogue. This is because the synthesis of

components is the most proven application area for expert systems in design. Thus design-experts, design-aids and design-informers would apparently be worthwhile tools on this dimension of Green's space.

For the above reason the bias toward system autonomy might also be slightly higher during synthesis than analysis. Hence there is another reason to suppose that Green's first group of tools would come into play during this stage of design.

However, the need for the deep and wide contextual knowledge which Green attributes to designers, is also important during synthesis. Hence tools from Green's second group are probably needed to influence the work of the expert-systems based tools.

Again the topic of individual versus shared use is rather orthogonal to this stage of designing. So this dimension of Green's space should not exert a great deal of influence on a developer's choice of tools.

By the time a full scale synthesis is performed it is likely that the relevant tools will either have been prepped by the system developer or populated with knowledge by the designers. So the choice of tools here will probably be dictated by the interfaces and interaction styles of the tools which were implemented to cope with the stage of analysis. However, the tools for collecting knowledge will become important at this stage if the system is intended to capture design rationale for purposes of re-seeding the system itself or to help construct an external knowledge base.

Synthesis is the most promising stage on this axis of the design activity space where expert systems based components could take a small step toward replacing the designer. This implies that the expert systems group of tools could become as useful as the tools for collecting knowledge and simulating designs. However, once again the important criterion is how well the necessary knowledge and heuristics for these tools were captured by the tools in group two during analysis or the early stages of synthesis. The main conclusion seems to be that powerful tools for the black-box aspect of synthesis are needed as a pre-requisite for the glass-box tools to replace the designer.

The need for numerical, semantic and pragmatic tools during synthesis is very similar to that encountered during analysis. Once again, then, it appears that fully fledged design-secretaries would be beneficial. However, given the current state of artificial intelligence technology, design-visualisers, design-experts, design-aids and rudimentary design-documentors and design-secretaries are more feasible.

4.3.4 Tools for supporting the stage of evaluation

The direction of initiation of dialogue between designers and knowledge aided design tools during evaluation depends upon the type of artefact being evaluated. If the artefact belongs to a glass-box domain, then there is considerable likelihood that a tool could guide designers through a proven sequence of evaluative steps. If the artefact requires more innovation in its construction or wickedness in its design process then the likelihood is that designers will need to initiate and control the dialogue. This observation implies that in the case of glass-box domains a selection of expert systems based tools, such as design experts, would be beneficial. However, in black-box domains tools that respond to user initiated dialogue, such as design-simulators, might be appropriate.

The above argument also applies to the dimension of system autonomy. In glass-box domains a design-expert might be sufficiently adapted to its task environment to perform tests that are largely uninterrupted. However, as the number of black-box elements of a design increase there will be an accompanying need for designers to add contextual knowledge to the evaluative process. This implies a need for smaller scale expert system based tools. It also implies the need for design-informers and design-strategists to alert designers to the possibilities of adding contextual knowledge to mathematical, syntactic or pragmatic aspects of an evaluation.

The requirement for shallow and narrow or deep and wide knowledge bases is once again dependent on the nature of the artefact being evaluated. Hence this dimension of Green's space is rather orthogonal to this stage in design and it seems far more applicable to the axis representing innovation and wickedness.

The requirement for individual and shared use tools will be highly influenced by the design strategy that is being employed. In the case of design teams, one successful strategy employs a group of designers to generate criteria for the evaluation then pares the team down to a single designer who refines these criteria and reports back to the team (Crickmay and Jones, 1972). This strategy suggests that: tools for shared use will be beneficial at the beginning of an evaluative cycle, tools for individual use will come into play during the next stage of the cycle, and that the team's needs will revert to groupware based tools once the designers needs to report back to the team. In the case of individual designers the dominant need is clearly for individual use tools. Conversely, in the case of participatory design, tools for shared use are a tautological requirement.

The choice between tools that are prepped with knowledge at development time and tools that collect knowledge probably shifts during the course of the evaluation. If

possible it would be advantageous to begin an evaluation with well prepped expert system based tools. However as the purpose of an evaluation is to compare one's mental model of a design against some internally or externally imposed criteria, it seems sensible to provide access to tools for recording the results of such comparisons. Hence a selection of tools from Green's group of tools for collecting knowledge would become more valuable as the evaluation begins to generate resolutions to existing issues and new issues to resolve.

The previous argument should also influence the choice between tools that replace and enhance the designer. Moreover, the technological issues that should govern the choice of technologies on this dimension are similar to those that were discussed for the stage of analysis. To recap, unless and until artificially intelligent tools can be provided with epistemological abilities to match those of human designers, there is a clear need for tools that allow designers to apply their own skills for measuring the success of their designs. The implication, then, is for design-informers and design-strategists to make knowledge as readily available as possible and for tools to help designers record their observations in as unobtrusive a way as possible.

The choice between tools for presenting and collecting numeric, semantic and pragmatic knowledge should once again be governed by the nature of the artefact being designed. It should also be governed by its need for technical innovation and innovation in design methodology. Hence, the relevant selection criteria will emerge in the discussion of the axis relating to innovation and wickedness.

4.3.5 Tools for supporting variant-design activities

As noted in Chapter Three, variant-design is the first stage on the axis of the design activity space concerned with innovation which qualifies as a design activity. Hence the discussion of tools on this axis of the space will begin with variant-design.

In Chapter Three, variant-design was subdivided into configurative variant-design and true variant-design. For the sake of brevity the discussion of knowledge aided design tools to support variant-design will focus on true variant-design. This is because configurative variant-design is the design domain which is most closely associated with expert system based technologies. For the purposes of this discussion there are two important implications which can be drawn from this observation. First, expert systems based knowledge aided design tools have been so well researched (c.f., Rychener, 1988; Winstanley, 1990) that deciding on which technologies to apply to configurative variant-design is almost a glass-box problem (the black-box component of the problem is deciding that the domain for the end product to be designed with the tool is a glass-

box domain). Second, each of these technologies cluster on the leftmost side of each dimension in Green's space. In other words, the choice of technologies for configurative variant-design is a tame problem whose most common solution is to select an expert systems based tool.

True variant-design resembles configurative variant-design insofar as it is a very well structured domain. However it also requires more innovation and wickedness from designers. In the terminology that was introduced and discussed in Chapter Three, then, true variant-design requires a good deal of glass-box and black-box thinking and communication on the part of individual designers and design teams. The arguments presented in Chapter Three therefore suggest that technological support for true variant-design demands an array of tools that address these black-box design processes in addition to a slightly reduced array of tools to support glass-box design processes.

This line of argument suggests that the selection process for tools to support true variant design along each dimension of Green's model space can be subsumed by the following four guidelines. First, the tools that would be used to support a configurative variant-design process should be included and encapsulated within black-box interfaces. These interfaces can be supplied in the form of design-informers, design-aides, design-simulators and design-strategists. Second, design-simulators and design-visualisers should be used to provide graphical interfaces to the knowledge contained in these tools. Third, where possible these interfaces should contain design-demons to act as critics to alert designers to any problems that can be identified by an underlying knowledge based component (see Silverman, 1992). Fourth, these newly encapsulated tools should be augmented with a selection of tools from the right hand side of Green's tool set.

An exception to this last guideline is that certain graphical tools, such as design-simulators, become harder to usefully implement as the amount of innovation required by a project increases. Moreover, they may also come to constrain a designer's creativity (see the discussion of DODEs in the next chapter). This is because 'intelligent' iconic drawings of an artefact being designed require an underlying model of the artefact's intended behaviour. This makes it difficult for designers to create artefacts within the tool which diverge from the appearance or behaviour that the tool was prepped with at development time. Once the borders between true variant-design and innovative-design start to become indistinct within a project, then, this rule appears to become much harder to apply.

To summarise, then, design-experts appear to have a role to play as core technology along each dimension of Green's space. However these expert systems based tools need to be enhanced with the softer and more interactive tools for regurgitation and collection that appear on their immediate right hand side on each of these dimension.

The tools also need to be augmented by stand alone tools of the kinds that appear on the far right of each dimension.

Hybrid systems of such tools have begun to be implemented by researchers such as Fischer and Silverman (Fischer *et al*, 1996; Silverman, 1992). The strengths and weaknesses of these hybrids for configurative variant-design, true variant-design and innovative-design will be discussed in Chapter Five.

4.3.6 Tools for supporting innovative-design activities

As innovation becomes a more pressing need within a design project, the enhanced expert system based tools discussed above are likely to become less useful and to work less well than some of the other tools in Green's tool set. This is because conventional expert systems based tools require an integrated store of logically consistent knowledge. Unfortunately, as Lawson shows (1990, 1993), it is much less likely that designers will maintain such a store of logically consistent knowledge during a process which requires innovation (Bucciarelli, 1988, 1994; Latour, 1996). Moreover, as the second-generation design researchers discovered, the combination of public and private knowledge found within a design team is even more likely to be contradictory and ambiguous.

If the ambiguous nature of this knowledge makes tracking designers via expert systems based tools problematic, then it can also make it harder to implement tools which represent design knowledge in graphical formats. This is because the graphical formats found with tools such as design-simulators are usually linked to underlying textual or logical representations of the artefacts they represent.

If the dialogue between the tools and the designers is initiated by a tool rather than a designer then the likelihood of exploring new avenues has the potential to decrease. Unfortunately, however, for the reasons mentioned above, a design-visualiser with embedded expert system based components would be a less viable way of allowing designers to initiate and control the dialogue. One alternative would be to support dialogues between people rather than between people and machines.

For philosophical reasons, in innovative-design it is difficult to envisage that an autonomous system could perform the entire task of analysing, synthesising and evaluating a radically new artefact. This is because, as was argued in Chapter Three, each of these processes requires a measure of each of the other processes. In the case of a knowledge aided design tool this means that the evaluative criteria for each step needs to be formally analysed, synthesised and evaluated (Elton, 1993) in a computational fashion. This unfortunately leads to an infinite regress which is caused by the necessity for a program to generate innovative criteria with which to evaluate its own evaluative

criteria. When the amount of innovation required is small it is possible that such philosophical niceties can be side-stepped. However, given current technology and understanding of creativity, as the demand for innovation increases, it becomes less likely that a program could incorporate new concepts into its knowledge base and perform useful evaluations.

Innovative design seems to require a mixture of both shallow and narrow domain specific knowledge and deep and wide contextual knowledge. For these reasons it might be sensible to step outside of the subject area of knowledge aided design and provide links to the external world via full scale computer mediated communication technologies (Candy and Edmonds, 1996).

Once again the provision of tools for individual and shared use is appears to be governed by the size of the design team and their problem solving strategies. It is thus orthogonal to the demands set by innovative-design (except when knowledge needs to be shared in line with organisational norms).

The choice between providing tools that are prepped with knowledge and tools that collect knowledge should be influenced by the argument relating to shallow and broad knowledge in the context of innovative-design which was presented above. Since the knowledge that innovative designers will come to require can not be predicted, designers might be best served by conduits to external sources of knowledge and forums for communication with other designers.

Similarly the argument relating to the autonomy of knowledge aided tools that was presented above suggests that tools which enhance designers are a more appropriate choice than tools which replaced them.

Finally, the choice between numeric, semantic and pragmatic tools should probably be weighted in favour of the pragmatic section of this dimension. This is because innovative products are expensive to research and develop. Ideally, then, the needs of customers in a market place need to be pragmatically evaluated before investments are made. Unfortunately, with the exception of tools for analysis which invite designers to draw their own pragmatic decisions, the few knowledge based tools offer significant pragmatic functions. This suggests once more that the need is for tools for computer mediated communication between designers and other stake holders. Such tools include design-secretaries, design-translators and design-simulators.

4.3.7 Tools for supporting strategic-design activities

There is a paucity of theory about how to stimulate and manage creativity in commercial design (Druckner, 1985; Peters, 1993). Therefore the selection criteria for

tools to support variant-design and innovative-design rest on tenets that are more probably reliable than those for selecting tools to support strategic-design. Moreover, as Chapter Two observed, design that requires a great deal of creativity is on the cutting edge of research into knowledge aided design systems (Candy and Edmonds, 1996). Hence it is unrealistic for commercial designers to expect well-founded assistance from knowledge aided design at this point in the subject area's history.

It is therefore suggested that slightly stronger versions of the arguments presented about innovative-design should be considered a viable starting point for the development of guidelines for supporting strategic design.

These arguments should also be buttressed by the prescriptions of Candy and Edmonds (Candy and Edmonds, 1996) which were reviewed in Chapter Two. These can be summarised as follows:

- provide means of access to knowledge about unforeseen subjects within a support system as a design progresses
- provide ways to explore parallel solutions to problems
- provide means for designers to communicate with one another and to recognise one another's skills
- provide ways to adapt the current formulation of a problem and the representation of strategies for its solution.

These prescriptions can be enriched by Pugh's (1989) argument that concept design naturally precedes the specification stage in innovative-design projects. This seems even more relevant to strategic-design. Hence systems and their component tools should probably be biased toward this new arrangement of the standard stages of design.

In summary, following the prescriptions of Candy and Edmonds, the demands set by strategic-design suggest the need for computer mediated communication rather than for expert systems based tools. It is open to question whether this conclusion could be altered by an improved understanding of strategic-design, which is an issue for design researchers, or whether new paradigms of knowledge aided design could offer viable alternatives.

4.3.8 Tools for supporting tame design activities

In the previous chapter it was argued that tame design activities tend to equate with glass-box and first generation design methods. It was also suggested that glass-box domains and first generation design methods imply the potential viability of expert systems based approaches. Unfortunately, however, it was further suggested that this problem is more complicated for the majority of design activities. There are two reasons for these complications. First, true glass-box domains and their associated first generation design methods are rare. Second, most first generation methods require encapsulation within a black-box method which addresses some of the problems that were identified by second generation design researchers.

This argument implies that expert system based tools are only appropriate in the majority of glass-box domains and tame design activities when they can be provided with a black-box interface. However, this implication presents knowledge aided design systems developers with the two problems. The first is deciding upon an appropriate content for their expert systems based tools. The second is deciding upon an appropriate content for their black-box interfaces. In short these developers face a miniature version of the full-size problem of designing interfaces to accommodate wicked design activities. However this problem is partially ameliorated by the fact that they can also rely on some of the design knowledge which was structured by first-generation design researchers.

Since the design of expert systems is a relatively tame and glass-box activity, the more pressing of these problems is that of deciding upon an appropriate content for the interfaces to the expert systems based components. One strategy which might be appropriate is to balance an appropriate ratio of content from the relevant design domain with structure from a first generation design method in an iconic interface. In the next chapter discussions of various systems that have been implemented or proposed by Fischer and co-writers (1992, 1994a, 1994b, 1996), Silverman (1992) and Blessing (1994) reveal that this strategy is plausible. However they also reveal that the wickedness of designing a design activity becomes entangled with the amount of innovation that is required by a design project. Although the prescription of encapsulating a design-expert in a selection of Green's less traditional tools appears to be a well motivated response to this problem, it is a response that becomes less useful when designers require support for tame design methods that are applied to more innovative products.

In the context of relatively tame domains the above arguments suggest that designers should probably initiate dialogues with their traditional expert systems based tools. This

could be addressed by draping an interactive tool such as a design-visualiser or a design-simulator over the underlying framework of expert systems based tools. If a first generation design method is available, then this process could be guided by a passive or interactive design-informer or design-strategist. One promising strategy would be to depict the method in question in a visual representation that simulates the design process rather than the development of the artefact itself. This is reminiscent of the interface to Gilleard and Lee's CITIS system (Gilleard and Lee, 1993). However, the main differences between these approaches is that CITIS is domain dependent and some first generation design methods are more domain independent.

By following the above strategy expert systems based tools could be set relatively autonomous goals so long as these can be interrupted and reset whenever designers generate insights about their project. Once again, visual tools such as the design-visualiser or a design-simulator might provide useful feedback which could be enhanced by a design-demon which is based upon the first generation method at hand.

Assuming that the problem is sufficiently tame to apply a first generation design method, the interfaces to the traditional expert systems based tools could be based on the relatively shallow and narrow knowledge contained in such methods. This implies that tools such as design-demons might well be feasible in the roles they have already been assigned. However these tools may become less feasible to implement as the need for innovation increases. This is because they will need to apply standard design knowledge to novel technologies.

The need for tools to support individual and shared use is fairly orthogonal to this axis of the design activity space. However, tools such as design-secretaries and design-documentors might well be useful to recover design knowledge from tame domains. This is because these tools offer knowledge engineers and design researchers some scope to understand the nuances that have been applied to familiar problem solving strategies and some hope that these can be reused in future systems and methods.

Even though the strategic tools that are being recommended for tame problems require prepping with a good deal of initial knowledge, such tools should also be networked with tools that can collect new design knowledge from their users. This hybrid approach should empower a process of seeding and re-seeding which will help designers and researchers to iterate toward better understandings of design methods and knowledge aided design systems. The problem with this approach is the need to persuade designers to invest time in recording design rationale which may not benefit them directly or may be perceived as not benefiting them directly.

In very tame domains it is tautological that tools which replace the designer should be sought. However, for the reasons that were given at the beginning of this discussion,

these domains are very rare. It is therefore suggested that tools which are designed to enhance the design are the most appropriate elements for the interfaces to systems which address even relatively tame projects.

The seventh dimension of Green's model space measures the amount of mathematical, syntactic and pragmatic knowledge that is contained within a tool. The interfaces to tools which are aimed at relatively tame projects should probably focus on supporting the semantic section of this dimension and provide some limited support for the pragmatic section. The challenge here is to invent semantic languages of design, whether textual or graphic, which will alert designers to the pragmatic outcome of following various paths toward the goal of completing their design. Example of such languages include the annotated flow charts which were discussed in Chapter Two, and which Gilleard and Lee implemented in the CITIS system (Gilleard and Lee, 1993).

To summarise, then, even tame problems will almost always contain some wicked elements that require some second generation style design strategies. The challenge for developers of knowledge aided design systems appears to be finding ways to leverage their expert systems based technologies with interfaces that include knowledge derived from first generation design methods. If this challenge is met, then these interfaces might help designers to delegate glass-box activities to the computer and to concentrate on performing the black-box design activities which are a particular human forte. However this is a relatively new area in knowledge aided design and the main example, which Gilleard and Lee implemented in the CITIS system (*Op Cite*), is extremely domain dependent.

4.3.9 Tools for supporting wicked design activities

At first blush the arguments which were applied to designing the underlying technologies and interfaces for tools aimed at tame design projects also seem applicable to tools that are aimed at wicked design projects. Upon deeper analysis, however, an exception to this guideline becomes apparent and the problem that Green's tool set might be incomplete also emerges as being especially problematic for developers of these tools.

The exception to this guideline is that if a system is intended to support a second generation design method, in which the 'designing of designing' is the key consideration, then its interface should logically allow designers to alter any representations of the method that they are following, redesigning, or inventing. This suggests an even greater need for tools to collect knowledge from designers engaged in

wicked projects. Such tools include design-secretaries, design-translators, design-documentors and design-simulators.

The prior warning that Green's tool set might be incomplete due to a lack of fully implemented systems becomes especially appropriate in the case of tools for wicked design. This is due to the following argument regarding the lack of implemented and evaluated systems that are aimed at wicked design projects.

Unfortunately, the best current example of a system which is designed to address wicked design is Blessing's simulation of PROSUS (Blessing, 1994). As the discussion of PROSUS in Chapter Five will demonstrate, one of the envisaged advantages of using an implementation of PROSUS would be the facility for adding new representations of design methods to the system at 'run time'. However, at present, this envisaged advantage must be tempered with two observations. First, PROSUS, in common with Candy and Edmonds' proposed systems for creative design, has not been yet implemented. Moreover, it appears that implementing PROSUS would be an expensive and research intensive project because it is intended to support the entire design process. Second, the representation of a design method which PROSUS employs is a matrix. As Chapter Five argues, this representation is powerful because it is a very general method of representation which can be applied to problems for which little strategic design knowledge exists at the outset. However, this very generality brings the weakness that a matrix lacks the specific structure found in first generation design methods. For example, iterative structures and decision points can not be represented in a matrix with the same ease as within an annotated flow-chart. So the specification of PROSUS can be interpreted as developing the notion of representing a design method as a graphic structure and providing insights into how this design pattern might be applied to supporting wicked design. However, it cannot offer design researchers and developers of knowledge aided design systems any empirical data about the resolution of such issues.

At the time of writing, then, the level of knowledge about tools and interfaces for supporting wicked design is similar to that about tools and interfaces for strategic-design and creative-design. This is because both types of tool remain research issues that lie beyond the currently implemented scope of the discipline.

For this reason the main suggestion that can be made here is that knowledge aided design tools need to be developed which enable designers to perform end-user programming in order to adapt the representations of their design methods. Since visual programming is an active and successful research area (Burnett and Baker, 1994), the key issues in such research will probably lie in the cross-over between design research,

external representations, information design and knowledge representation, rather than in the technical area of developing end-user programming tools.

In sum, the task of adapting design-informers, design-aids, design-documentors and design-simulators to allow interactive simulation and representations of design methods appears to be a pressing research agenda. As Chapter Two argued, pursuit of such agendas requires collaboration between design researchers and developers of knowledge aided design systems. Until such research is performed and evaluated it will be premature to offer more specific guidelines to developers of knowledge aided design systems to support wicked problems.

4.4 Summary

Two of the main aims of this thesis are to motivate the definition of the class of MODEs and to identify a niche for them in the design activity space. Many of the concepts and technologies that these moves will require have now been described and interrelated in this chapter. Specifically, this chapter has described Green's tool set for knowledge aided design and his model space for such tools. It has also coupled this tool set and the seven dimensions of Green's model space with the three axes of the design activity space which was defined in Chapter Three.

The work presented in this chapter will contribute to the work that remains to be presented in the following ways:

- Chapter Five will begin by using the theoretical work that has been presented here to estimate the range of utility of existing DODEs and some proposals for process based design environments that were discussed in Chapter Two. These estimated ranges of utility will indicate the existence of a niche for systems to support designers engaged in relatively tame and relatively innovative design. The identification of this niche will lead to the definition of the class of MODEs.
- Chapters Six and Seven will illustrate the proposal of the class of MODEs by developing a system to support designers engaged in relatively tame and relatively innovative design. The conceptual design of this system is determined by the mappings that have been presented in this chapter.

Chapter Five

Identifying a niche in the design activity space for MODEs

5.0 Overview

The work reported in this chapter begins with the identification of a niche within the design activity space which is in need of support from a new class of system for knowledge aided design. It then proposes the class of MODEs as a type of system that could fill this niche. This analysis requires estimating the range of utility of DODEs and process based environments in the design activity space to reveal a region which is not supported by traditional knowledge based tools nor by these recent alternatives. This work addresses the second half of hypothesis two and the whole of hypothesis three:

- H2** It is possible to develop mappings between this design activity space and Green's model space of knowledge aided design systems. These mappings should enable the approximate range of applicability of individual tools and classes of systems to be indicated. These mappings should also make it possible to re-characterise the accepted range of applicability for some existing classes of system.
- H3** It is possible to search the design activity space for at least one niche that is unsupported by existing classes of system and to propose a new class of system, composed of tools located in Green's model space, that appears suited to supporting activities in this niche. The use of the design activity space should indicate that the particular class that has been proposed is suited to supporting activities in the niche and that the surrounding classes of system are unsuited to offering such support.

The chapter contains seven sections. The first section discusses the method that was used to reveal the niche for MODEs. The second section begins the analysis by investigating the range of utility of DODEs. It is argued that the systems of Fischer and Silverman can be grouped together as the class of DODEs. The range of utility of existing DODEs is then investigated. In the third section the question of whether this range can be extended with envisaged technologies and approaches is addressed. In the fourth section the range of utility of process based environments is

investigated from a theoretical perspective. In the fifth section the niche for MODEs is located in the design activity space between DODEs and process based environments. In the sixth section a technical definition of MODEs is presented. In the final section this work is summarised and the function of the second part of the thesis is introduced.

5.1 Method

The main goal of this chapter and of the thesis is to expand the set of tools and systems that are available to developers of systems for knowledge aided design such that these will support design activities that are currently left unsupported. For this reason it is necessary to summarise the range of utility of traditional knowledge based systems and to analyse the range of utility of the recent alternatives that have been discussed in previous chapters. Once these ranges have been estimated within the design activity space, the focus can shift to identifying regions that lie outside of them. Such regions, it can be argued, represent niches in need of support from new classes of system. The proposal of such classes of system would therefore address the main goal of the thesis.

Green's arguments about the limitation of the range of utility of traditional knowledge based tools to domain oriented tasks requiring little inventiveness with materials or method were reviewed in previous chapters and were supported by the mappings presented in Chapter Four. If Green's arguments are sound, then it can be accepted that traditional knowledge based systems are suited to supporting a particular region of the design activity space. This region encompasses synthesis, variant-design and the 'design strategy' of heuristic search (which, it has been argued, is more properly referred to as configuration rather than design).

So this region of the design activity space can be 'ring fenced' leaving the remaining space to be matched to tools and systems for knowledge aided design. This leaves the following regions of the design activity space to be in need of support: analysis and evaluation; true-variant design, innovative-design and strategic-design; and first and second generation design methods. For this reason the range of utility of DODEs and the process based approach will be analysed along these axes with regard to the regions which remain in need of support.

5.2 Estimating the range of utility of DODEs

In this section the DODEs which were described in Chapter Two will be accessed in order to determine the types of design activities for which they seem to be suited. In particular, the section will be concerned with the axis from the task space which is concerned with innovation. It will be argued that the kind of DODEs which have been implemented by Fischer and Silverman share an approach to interface design and the underlying implementation of their programs, which restricts their range of utility on this axis to repeat orders and configurative variant-design.

The observations that will be made in the discussion of innovation also affect the estimation of the ranges of utility of DODEs along the other axes in the design activity space. It will be argued that since the common approach to interface design affords scant opportunity for problem setting on the part of the user. So DODEs are most suited to the stages of synthesis and, in a limited sense, evaluation. Since the tools underlying the interface objects in a DODE perform heuristic search, it will be argued that there is an implicit assumption that DODEs will be used in a very 'tame' fashion. This is because the designer uses the system to explore a constrained set of alternatives that were prepped by a knowledge engineer. However, as the hallmark of designers as users of software is that they will invent unforeseen ways to use it (Beardon *et al*, 1995), this point will be made subject to the caveat that one cannot predict exactly how designers will interact with a DODE.

5.2.1 The range of utility of DODEs on the axis concerned with innovation

Fischer and Silverman developed a number of knowledge aided design systems in the early 1990s. These systems addressed such diverse domains as kitchen design (Fischer, 1992, 1994a) and ship-board antenna placement (Silverman, 1992) (see figure 5.1 and figure 5.2). The commonality between these systems is that they are all centred upon graphical design-simulators. Fischer suggests that the advantage of including such design-simulators is that "they give designers the feeling that they interact with the domain rather than with low-level computer abstractions" (Fischer, 1992, p 206). These design-simulators present their users with an iconic representation of an evolving work item which is surrounded by textual information and feed-back from design-informers and design-demons. Fischer suggests that the advantage of including such tools is that they "identify potential problems in the artefact being designed" (Fischer, 1992 p 207). Since these systems contain such similar technologies they can be grouped together and categorised as DODEs.

Designers interact with the design-simulators within a DODE by selecting pre-defined objects from on-screen palettes and placing these to become elements within the confines of an evolving work item. Depending upon whether one's view point is from artificial intelligence or design research the work items within such design-simulators can either be referred to as 'partial plans' or as 'partial designs'. Since the current goal is to describe why the technologies used by Fischer and Silverman restrict designers to addressing particular types of design activities, the work items will be referred to as partial plans.

The objects which serve as elements in such a partial plan can represent real world items such as kitchen sink or an antenna which is destined for the deck of a naval frigate. However, since the designers can only select locations for these pre-defined objects they are restricted to either recreating a repeat order or to performing some configurative variant-design. In other words the designers can only recombine these objects with known behaviours into new configurations rather than being able to invent new fundamental building blocks for their designs.

The reason for this restriction is that the representation of the work item within the design-simulator must be conventionally represented as a series of 'low-level' computer abstractions so that its elements can be constrained by the rules of the particular design domain. This is because the design-simulator is typically surrounded and set in context by textual output from various design-experts and design-demons. Although in some cases the designers can also elect to adapt existing plans held in a design-informer (Fischer, 1992).

These design-experts and design-demons generate their output by applying processes of inference and deduction which refer to the structure of the elements and their relative positions within the partial plan. In other words, standard artificially intelligent critiquing technologies¹, derived from the subject area of planning (see Allen, Hendler and Tate, 1990), continuously access the current feasibility of the partial plan and report any conflicts or errors to the planner. The difference between such a system and standard expert systems of the 1970s and 1980s is that the planner in question is a human designer rather than an automated general problem solver (Newell and Simon, 1972) of the type described in Chapter Two.

¹ Unfortunately there is room for confusion here as Silverman appropriated the term 'critiquing' for his own purposes after it had already acquired a meaning in the subject area of planning. The term is used here in the traditions of planning, e.g., Sacerdoti (1975). In this usage a critic is an artificially intelligent component of a planning system which reflects on the systems efforts in search of pernicious interaction between sub-goals.

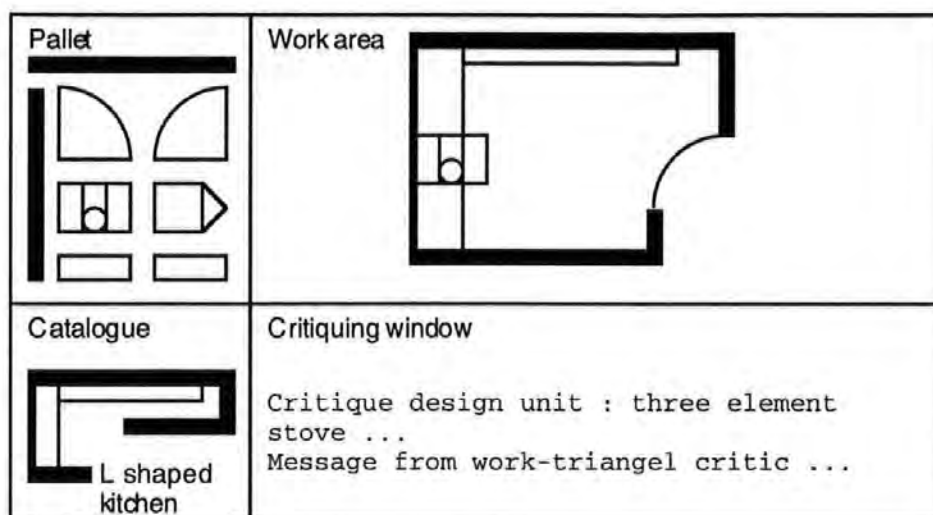


Figure 5.1: The interface to Fischer's Janus-CRACK system (re-drawn from Fischer, 1993, p 244).

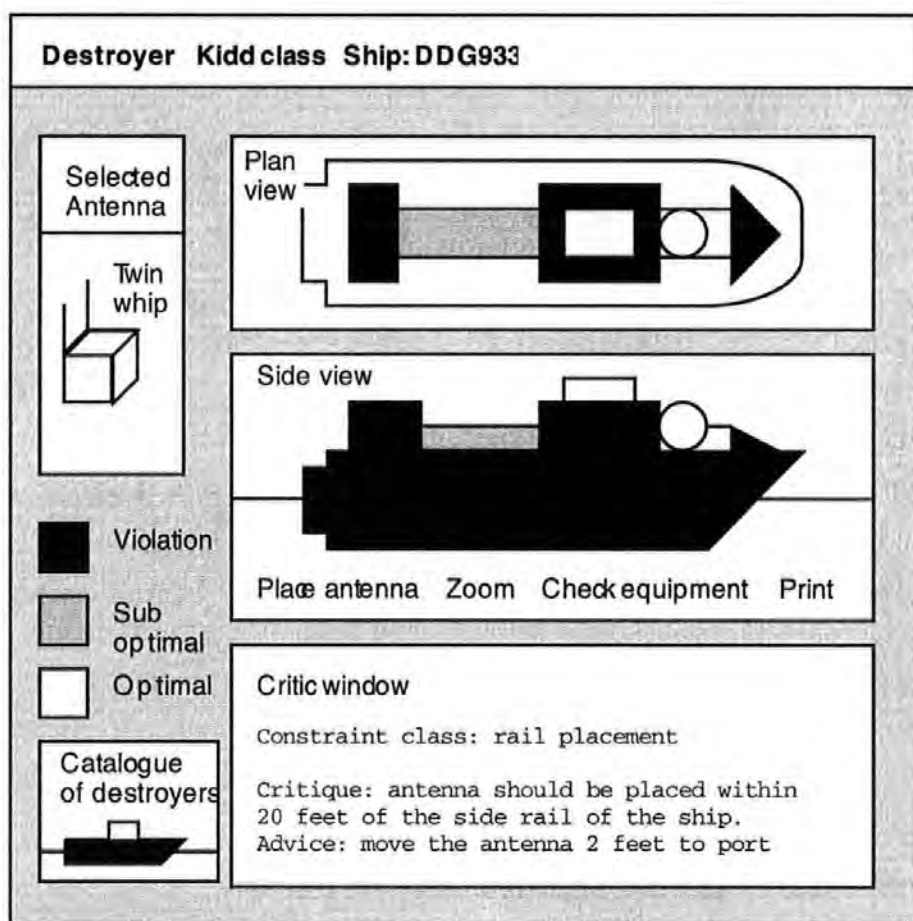


Figure 5.2: The interface to Silverman's system for placing antenna on warships (re-drawn from Silverman, 1992, p 110).

For example, when a designer places a sink within a simulation of a kitchen such that it violates an ergonomic building code, Fischer's JANUS system responds by 'noticing' that a building code is being violated and responding by issuing the following textual warning:

The length of the work triangle (double-bowl-sink-1, four-element-stove-1, single-door-refrigerator-1) is greater than 23 feet

(Fischer, 1992, p 208)

For the system to be able to issue this warning it needed to be able to refer to explicit knowledge about each element within the work item and to explicit knowledge about the validity of their interactions.

In some cases such feed-back and commentary might be presented graphically from within the representation of the work item. Indeed Silverman prescribes that "where feasible, it is best to deliver criticism through visual metaphors and by direct manipulation of the work item" (Silverman, 1992, p xiv). However, this method of delivering feed-back is an example of what programming environment designers term 'syntactic-sugar'. That is, although the information is being re-represented in a format which is more palatable to people, the underlying data-representation is the same. For this reason, graphical feed-back and commentary have the same requirements for elements that follow specific rules as its textual equivalents.

In summary, the developers of a DODE must try to understand a domain sufficiently to be able to provide designers with the fundamental building blocks to be able to solve problems within it. They must also implement these fundamental building blocks as 'low-level computer abstractions' and mask the abstractions with interfaces that will make sense to designers. However it seems impossible for the developers of a DODE to foresee all the ways that designers will conceptualise their chosen domains or predict all of their possible requirements for new fundamental building blocks. This argument suggests that DODEs which are centred around a combination of a design-simulator and a selection of expert system based tools must almost inevitably restrict their users to addressing repeat orders and configurative variant-design problems. For this reason the range of utility of DODEs on this axis is between repeat orders and configurative variant-design problems (see figure 5.3).

5.2.2 The range of utility of DODEs on the axis concerned with analysis, synthesis and evaluation

Turning to the axis of the design activity space that displays the stages of analysis, synthesis and evaluation, it can be argued that graphical design-simulators afford scant opportunity for problem setting and evaluation on the part of the user. Their main strength is at the stage of synthesis.

As argued above, the developers of a DODE need to provide the end-user with a set of 'smart' objects located on a pallet. This means that the user's opportunity to analyse a project in terms that can be 'understood' by the system is largely constrained by the objects and operations that are made available on the pallet and in any associated menus and tools. This argument should be made subject to the caveat that a designer is still free to analyse the situation and decide whether it should be addressed with other materials or working methods.

The main strength of a DODE on this axis of the design activity space is the stage of synthesis. This is because synthesis is the forte of the planning based technologies which design-experts and design-demons implement. If the knowledge acquisition phase of a DODE has been well executed, then, designers can expect useful support at the stage of synthesis. However this support will only be useful if the designer has chosen to work within the constraint of synthesising objects which are represented in the system.

The evaluation of a particular solution is largely restricted by the set of objects, operations and knowledge that are available to the user. Although slightly more support appears to be available at this stage than at the stage of analysis. This is because the designer can expect some useful feedback if they have chosen to configure items that are supported by the system. However, once again a designer will be free to decide that the systems' advice at the point of evaluation is too limited and seek advice from other sources or support from other materials.

In summary, although DODEs support the stages of analysis and evaluation in very limited ways, a DODE should offer reasonable support for synthesis providing sufficient knowledge about the domain is available to the developer. So the main range of utility of DODEs on this axis is the stage of synthesis (see figure 5.3).

5.2.3 The range of utility of DODEs on the axis concerned with design methodology and evaluation

Since the tools underlying the interface to a DODE help the designer to perform heuristic search, there is an implicit assumption underlying the design of DODEs that these systems will be used in a very ‘tame’ way. When using a DODE according to its developer’s intentions, the designer explores a constrained set of alternative configurations of a work item while receiving feed-back from the design-experts and design-demons within the system. Such a partnership between a person and a machine brings the advantages of human context-setting, which is the glass-box element of the problem area, and machine inference and deduction, which is the black-box element. As argued in Chapter Three, this can be an advantageous strategy. However, it is a limited one because less structured glass-box activities, such as synectics, cannot be supported with current technologies for implementing tools such as design-experts and design-demons.

However, this argument must be subject to the caveat that a designer could invent new ways to use a design environment. For example, a designer might decide to use a DODE to search the space of well-known possibilities in order to understand the rules that an innovative solution would break.

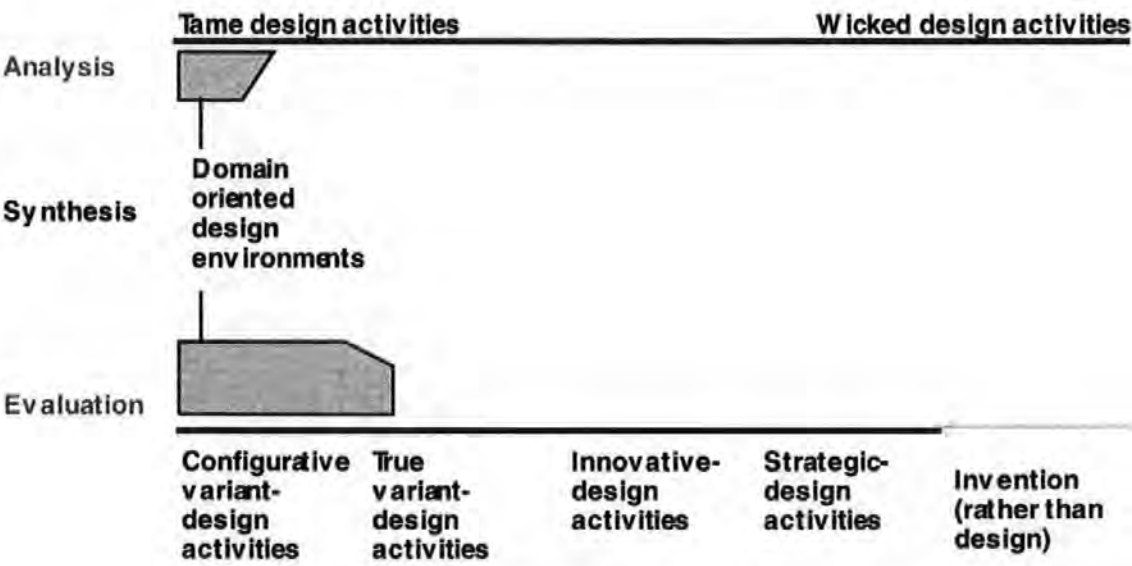


Figure 5.3: This diagram illustrates the estimated range of utility of DODEs along two important axes of the design activity space. The suggestion that DODEs are more suited to synthesis than analysis or evaluation is illustrated by the strength of the lettering for these stages in design on the left hand side of the diagram.

Subject to this caveat, then, the argument presented here indicates that the range of utility for DODEs on this axis of the design activity space is ordinarily restricted to tame design strategies (see figure 5.3).

Since DODEs are an emerging technology it is important to assess what whether additional research might extend this range of utility. For this reason the paradigms of user seeding and user programming will now be investigated to ascertain whether these could lead to enhanced DODEs, which it will be suggested might offer an alternative means to address tasks requiring slightly more innovation, analysis and evaluation.

5.3 Could an alternative approach to implementing DODEs with user seeding and end user programming extend their range of utility?

It appears that there is a plausible way to extend the range of utility of DODEs by taking an idea which Fischer and his co-workers have applied to the textual representations of knowledge about a work item and adapting it so that users could build their own low-level computer abstractions. In time, developers of knowledge aided design systems could help such designers to convert these abstractions into full sized objects within their DODE. In this section a path toward developing such systems will be indicated and the potential range of utility of such systems in the design activity space will be postulated.

As Fischer *et al* (1996) observe, there are two extreme approaches that can be taken to inserting knowledge into a knowledge base. At one extreme is the traditional approach which is followed by developers of conventional expert systems. This attempts to elicit and codify all of the knowledge which the developers and their subjects consider to be relevant before a system is put to use. At the other extreme is an approach in which the developers provide the end-users of a system with a domain independent shell then invite them to populate it with knowledge through long term usage of the system¹.

¹Research in this direction is well underway. For example, Rodgers and Huxor (1998) have developed a knowledge based system called CADET (Rodgers, 1995), which is aimed at product design evaluation (tests have been reported in the consumer product design domain of razors). Rodgers and Huxor see a primary role for artificial intelligence systems in design tasks involving variant and innovative design where the system's primary function is as a 'text', that is a medium for communication between designers, rather than as a 'problem solver'. In such a system automated problem solving is seen as an important by-product of a more important process that helps designers to think, communicate and record design decisions. Rodgers and Huxor also suggest that the world wide web will provide a useful media by which to share information in such systems (1998).

Fischer and his co-workers have synthesised these approaches for the following reasons. The first approach can fail because it is unable to keep abreast of changing circumstances and eliciting tacit and contextual knowledge, which might enable it to do so, is not a practical proposition (Dreyfus, 1992). The second approach can fail because asking designers to enter knowledge during the design process can interfere with their creativity and typically designers lack the specialist knowledge which is required to codify such knowledge into an appropriate format.

However both of these approaches have valuable components. The expertise in structuring knowledge which a developer of knowledge aided design systems applies to knowledge is a necessary ingredient for a practical knowledge base. If a knowledge base is to remain current then a continuous source of domain specific knowledge is required. So a designer brings invaluable expertise to system maintenance. It therefore makes sense for developers and designers to collaborate on a regular basis throughout the working life of a knowledge aided design system.

Fischer's synthesised approach to knowledge base construction and maintenance contains three stages: seeding, evolutionary growth and re-seeding. These stages can be described as follows.

During a seeding process, developers collaborate with designers to develop a DODE. The developers hope to acquire some knowledge from the designers in order to codify whatever system specific knowledge is felt to be required. Since the developers provide expertise in knowledge based systems and designers provide domain specific knowledge, this stage is quite traditional.

The process becomes novel in the larger context of the whole paradigm because Fischer recognises that this seeding process can never provide a complete knowledge base and that some of the requisite system specific knowledge must be constructed or discovered rather than merely elicited.

During an evolutionary growth process the designers add knowledge to the knowledge base of the DODE. Fischer observes that designers are unlikely to have the expertise or the desire to record putative knowledge in a format that will be acceptable to a formal system for knowledge representation. Moreover, insisting upon undue formality can restrict a designer's creativity (Jones, 1988; Lawson, 1990; Beardson *et al*, 1995). However tools from the second group in Green's tool set can be provided which take account of these issues. For example, tools can be made available which allow a designer to describe the background to a design decision in natural language at a time of the designer's choosing. Moreover, these accounts can be attached to the representations of objects to which they refer. In time, a web of design decisions, rationales and objects can emerge.

In more detail, Fischer *et al* (1996) recommend that designers record the rationale, or conflicting rationales, behind design decisions so that these can be referred to in the future. They argue that writing design rationales can help designers to improve their own decision making. They also argue that interpreting design rationales can facilitate collaboration with colleagues and help designers to understand the thinking of their predecessors. Fischer *et al* define design rationale as a set of statements of reasoning underlying the design process that explain, derive and justify design decisions (Fischer *et al*, *Op Cite*). Such statements can be written by designers who contributed viewpoints and knowledge to a decision. They can also be from a senior designer who made a decision on the basis of such viewpoints and his or her personal biases.

Unfortunately, a pair of potential problems are inherently associated with design rationale capture. First, generating the points that appear in a design rationale can be seen as creative and wicked. So the success of the paradigm is in the hands of individual designers who use the DODE. Second, finite design rationales are always doomed to incompleteness. This is because a number of contributory factors could be described at any level of detail. Since concise accounts of design rationale are likely to assume a lot of contextual knowledge on the part of a future reader, much of the value of a design rationale will be generated by the talent which a designer displays for correctly predicting the level of knowledge of the future audience. So, research from areas such as technical writing theory, document design and forms design in which analysis plays an important role is particularly pertinent to the design of tools for design rationale. Silverman's research into critic design for technical writing is an ideal role model for tools that will help to capture this kind of knowledge in a concise, semi complete and well targeted format (Silverman, 1992).

While Green's tools for knowledge acquisition might offer designers the facility to generate new and valuable insights based on design decisions that they take while using a DODE, these insights are at best likely to be cast in informal and contradictory notations. Such notations are unsuited to providing the basis for inference or deduction. Hence the expertise of the developer is once again required. During a re-seeding process the developer returns to collaborate with the designers on the task of organising and codifying this new knowledge to produce a revised knowledge base. For example, a web of design decisions, rationales and objects might be restructured into a set of hypermedia links. The role of the developer is to contribute expertise in areas such as knowledge representation and knowledge consistency checking. Hence, the developer can relieve the designers of these rather alien tasks. Once again, the roles of the designers are to ground this revised

knowledge base in their experiences and to keep it focused on the relevant design domain. In effect, this process repeats the original seeding process. But it has the advantages and disadvantages associated with adapting an existing DODE to an altered perception of its domain.

Although the suggestions which were summarised in this subsection originally referred to textual knowledge, it appears that they might be transferable to graphical knowledge given that visual programming is now a commercial reality in environments such as Visual Basic. However, recording design rationale in a textual format appears less problematic than creating visual icons to stand for elements of work items which require associated formal knowledge.

A solution to these problems might lead to DODEs that can be used to address more innovative projects than those addressed by the DODEs developed in the early 1990s. Moreover, if users of such DODEs can implement their own representations of products and processes then the systems could offer enhanced support to the stages of analysis and evaluation.

However, until solutions to the problems of design rational capture that have been discussed above are found, DODEs seem destined to remain viable tools for repeat orders and configurative variant-design. So given present technologies, DODEs leave true variant-design, innovative-design and strategic-design as niches for alternative forms of knowledge aided support. Turning to the axis that measures tameness and wickedness, even these envisaged upgrades to DODEs would not contain pointers toward problem solving strategies. Moreover, since Silverman's debiasers are almost exclusively concerned with addressing domains rather than strategies, it can be argued that DODEs are implicitly biased toward tame design problems. The implication of this argument is that even such user-programmable DODEs would leave wicked design problems as a largely unexplored niche. For this reason the next section will investigate an approach that is concerned with supporting wicked tasks: Blessing's process based environments.

5.4 Estimating the range of utility for process based design environments

Some indications about the types of knowledge aided design systems which might come to occupy the niches associated with true variant-design, innovative-design strategic-design and more importantly, wicked design, can be found in the research conducted by Blessing which led to the specification of her PROSUS system (Blessing, 1994, 1996).

The majority of this section will examine the applicability of systems such as PROSUS to supporting true variant-design, innovative-design, strategic-design and wicked design. The main conclusion will be that PROSUS is a good starting point from which to develop programs to support the high-end of innovative-design and the whole of strategic-design. However, the section will also suggest that PROSUS is constrained by its attempt to accommodate the wickedness that is likely to be found in projects that require a lot of innovation to being more general than designers working on the tamer varieties of true variant-design and the low-end of innovative-design require. This suggestion leaves tamer examples of true variant-design and low-end innovative-design as niches which are not well occupied in the space by currently envisaged approaches to knowledge aided design. It will also be argued that PROSUS is inherently suited to supporting the stages of analysis, synthesis and evaluation, so long as the project falls within the boundaries described above.

5.4.1 The functional specification of the PROSUS system

Blessing's functional specification for her PROSUS system (Blessing, 1992, 1996) proposed that the technologies constituting an Issue Based Information Systems (IBISs) should be the foundation for what she termed a process based approach to supporting design. Both of these concepts will be discussed before accessing their implications for the types of problems for which Blessing's style of knowledge aided design system seems a viable approach.

Issue based information systems were invented by Rittel (Kunz and Rittel, 1970) and have subsequently been implemented by various developers of knowledge aided design systems (Fischer *et al*, 1991). There are two functions that an IBIS can play within a knowledge aided design system.

The first function an IBIS can address is to promote and support discussion among designers and other stake holders working on a project. This function can be implemented by offering designers tools to support computer mediated communication about their work item. In Green's terms these tools are equally concerned with collecting knowledge from one set of users and regurgitating it to other sets of users. A secondary duty of these tools is to collect and summarise this knowledge and store it in a knowledge base.

The second function an IBIS can address is to promote and support formal reasoning on the part of individual designers. This can be implemented by offering designers tools that provide access to knowledge about the history and current of the current project or previous projects: as a system with an IBIS component is used

over a number of projects these tools can enable its knowledge base to become seeded and re-seeded with the histories of previous projects.

In Green's terms these tools are largely concerned with presenting knowledge from a knowledge base. In order to implement these functions IBISs are based on a second generation view of design. An IBIS developer should therefore opt to regard the design activity as a sequence of issues for designers to face, positions which designers elect to take in order to resolve these issues, and arguments posted for and against these positions.

These issues, positions and arguments can be represented within the IBIS as a combination of textual and graphic knowledge within a formal data structure. It is important that a data structure is chosen that is easy to search and which structures its content to reflect the order in which it was collected.

Such a data structure can be populated with knowledge by making design-documenters available to the designers. This enables them to record design decisions and design rationale during the process of design. These design-documenters can range from passive design-secretaries to artificially intelligent secretaries which attempt to infer design decisions from the designers' interactions with one another and with their software. In time the knowledge that is captured by interacting with these design-documentaries can be regurgitated via design-aids, design-informers, design-strategists and design-demons. Again, such tools can range from passive repositories to fully fledged artificially intelligent components.

There are two important points for developers to bear in mind about these tools: (1) the tools for collecting knowledge should not interfere with the spontaneity of the design process, and (2) the tools for regurgitating design knowledge should help designers to navigate through current or previous sequences of issues and resolutions in a way that makes their relation to the current state of the design as obvious as possible.

The way that Blessing's process based approach to design is expressed in the specification of PROSUS responds to the prescriptions of the first generation of design researchers. However, its content responds to the observations of the second generation of design researchers. PROSUS responds to the prescriptions of the first generation of design researchers by placing its tools in a design matrix that represents a two dimensional sequence of steps that designers are expected to perform. For example a simple design matrix would include a left-hand column of headings that represent the stages of problem definition, concept design and detail design (Blessing, 1994) (figure 5.4). The design matrix would also include a set of horizontal headings representing the steps of generation, evaluation and selection for

each of these items. To a great extent the content of Blessing’s design matrix can be seen as being influenced by the descriptive approach of the second generation of design researchers. However, as Blessing comments: “IBIS and its successors are mainly documentation aids. They do not provide the general issues to solve...for this reason these models are not able to guide the design process other than on a micro scale of deliberation” (Blessing, *Op Cite*, p 158).

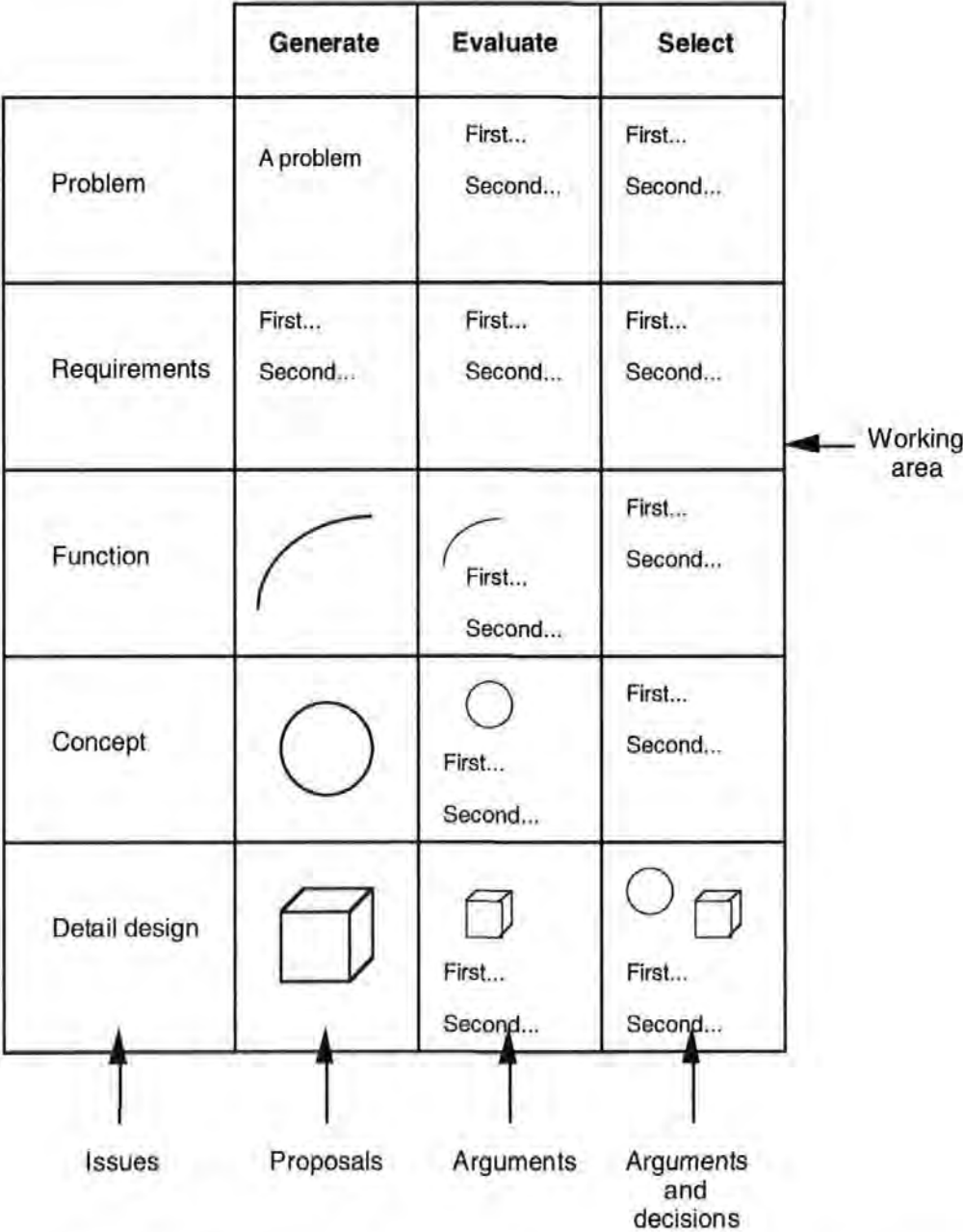


Figure 5.4: This diagram presents a simplified version of Blessing’s design matrix (adapted from Blessing (1996) p 116. Arguments and issues are represented by the textual labels and concepts and working drawings are represented by the geometric objects.

Blessing has contributed to the evolution of knowledge aided design systems and especially to its cross-over with the subject area of external representations. This is because the design matrix acts as a representation of the ordered stages that guide designers through first generation design methods. This representation steers its users toward cells containing individual IBISs for problem proposal, argumentation and resolution which encapsulate a micro-level stage in a design process within the context of a broad representation of its macro-level design strategy. In lay terms, the design matrix allows people to see the 'big picture' while its cells allow them to fill in the details.

PROSUS is quite radical, then, because it departs from DODEs by presenting designers with a structured domain independent shell whose contents will come to be populated with design knowledge. However, there are various open questions about the utility of PROSUS's structure, its content and the combination of these components. These need to be discussed in order to predict how beneficial this approach will be to supporting true variant-design, innovative-design, strategic-design and wicked design.

5.4.2 The viability of the design matrix for various types of design

The concept of employing a design matrix to structure an IBIS for supporting design is an imaginative and appropriate use of an external representation as an interface to a knowledge aided design system. It is particularly appropriate because one of the criticisms that can be made of concepts such as IBIS or other hypertext style programs is that their evolutionary style of growth can lead to unstructured and hence confusing knowledge. Moreover, this interface also makes use of the kinds of naïve diagrams of the design process which are familiar to designers. However, it appears that this representation might be most appropriate to designers who are engaged in high-end innovative-design, strategic-design and wicked design projects.

There are two reasons to suggest that the design matrix seems especially appropriate for designers who are engaged in high-end innovative-design and strategic-design, as well as for designers who expect to encounter a lot of wickedness during the course of a project. One reason is that the design matrix is a very generic two dimensional representation of the design activity which does not prescribe a fine grained and detailed course of action. This is important because in an innovative, strategic or wicked situation no such course of action can be prescribed before the

designers have encountered and tamed the unpredictable aspects of their problem. A second reason is that Blessing's specification has included the facility for designers to create recursively embedded matrices within the main structure. This seems to offer the possibility to build a more complicated and higher-dimensional representation of the design process during the design activity itself.

However, from the perspective of true variant-design and low-end innovative-design, the strength of design matrix can be recast as a weakness. This is because the generic nature of the design matrix which is so useful for innovative, strategic and wicked projects, might exclude some of the valuable strategic knowledge which has been stored in first generation prescriptive design methods. This suggests that designers who are working in such areas might benefit from having their knowledge aided design tools embedded in a more specific representation of the design process which shares the navigational strengths of Gilleard and Lee's CITIS system.

The problem with assessing the desirability of the content of the PROSUS design matrix is that PROSUS has not yet been implemented and tested with users. It is therefore an open question whether such a system could make use of very passive tools within its IBIS style components. If this were the case then a PROSUS shell could remain highly domain independent. However, if designers turn out to require more sophisticated knowledge based tools, then the PROSUS approach might be best regarded as a family of domain specific programs. This would have two implications. If the need were for domain specific systems then a PROSUS style system would probably be less applicable to design projects that involved innovation—although the architecture might still be appropriate for projects that involve a small amount of product based innovation and an amount of wickedness in the design process itself.

As the illustration of the design matrix figure 5.4 shows, the design matrix is designed to support the stages of analysis, synthesis and evaluation. However, until PROSUS is available, this must remain a matter of reasoned conjecture.

In summary Blessing's conception of a design matrix which structures a set of IBIS style components seems to offer a plausible path toward the development of knowledge aided design systems for innovative-design, strategic-design and wicked design projects. However there are reasons to suggest that the generic nature of the design matrix might be slightly less advantageous to designers who require support with true variant-design and low-end innovative design. Such designers might be better served by systems which apply the insight of structuring their tools within a diagram to the strategic contents of specific prescriptive design methods. One point which should be stressed is that these arguments are drawn on the basis of the

proposal for the PROSUS system. Until other process based design environments are proposed or implemented, however, these arguments must remain speculative.

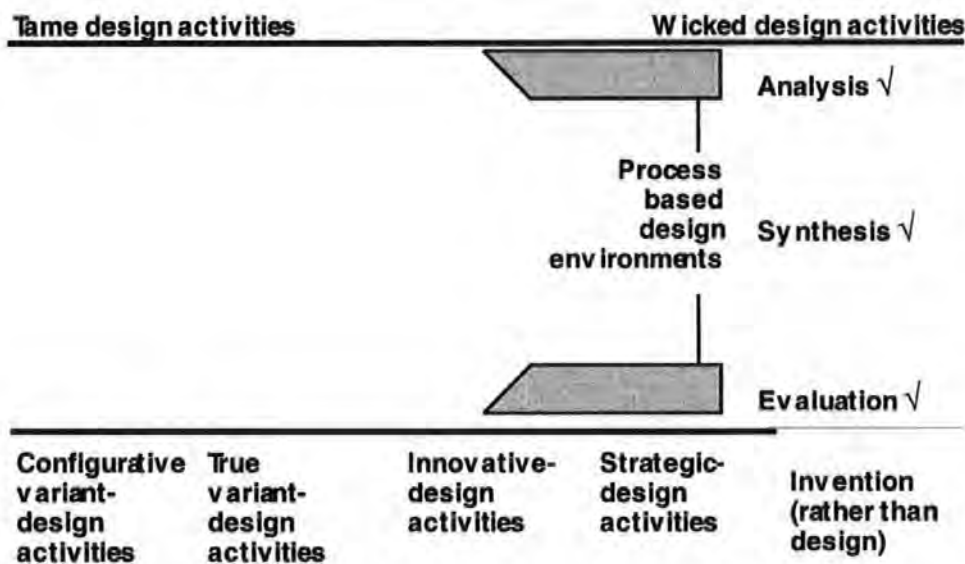


Figure 5.5: This diagram illustrates the estimated range of utility of process based environments along two important axes of the design activity space. The right hand side of the diagram shows that the process based approach appears viable for each of the main stages in the design process.

In summary, then, although process based design environments appear well suited to innovative and wicked projects, there are also good reasons to conclude that they are unsuited to tamer projects in true variant-design and the low-end of innovative design. Two arguments have been presented to support this claim. First, PROSUS is based on a representation of design strategies that appears overly general and hence misses the opportunity to offer specific support to particular prescriptive design methods. Second, because PROSUS is intended to support the entire design process, it is a very complicated definition for a system. It therefore seems likely that designers who are engaged in less demanding projects would appreciate less complicated software. The estimated range of utility of process based environments is illustrated in figure 5.5.

5.5 The identification of a niche in the design activity space

Now that DODEs and the process based approach have been placed in the design activity space it is possible to locate a niche in that space which lies between the ranges of utility of these two approaches. In this section the arguments presented above will be summarised and the location of such a niche on the axes of the design activity space will be described.

The analysis of DODEs found a common element in a number of systems that were implemented by Fischer and co-workers (1992, 1994a, 1994b, 1996) and Silverman (1992). This element is the iconic design-simulators which are the central tool within these DODEs. It was noted that the expert systems based technologies which underlie these design-simulators restrict DODEs to depicting pre-existing components. Since designers generate artefacts by using these design-simulators to arrange such components, these primary tools within DODEs restrict the nature of such artefacts to arrangements of these pre-existing components. Moreover, the only arrangements of such components which are acceptable to the design-simulators are ones that do not contradict the holistic knowledge about these artefacts which is contained within the DODE. These observations imply that the current generation of DODEs are restricted to supporting configurative variant-design.

The second part of the discussion of DODEs investigated whether the paradigms of evolutionary seeding and end-user programming could be applied to extend the range of DODEs toward supporting true variant-design or the low-end of innovative-design. It was suggested that seeding and end-user programming to DODEs might extend their viability of toward true variant-design and the low-end of innovative-design. So an optimistic appraisal of the future potential of DODEs places the upper limit on its range of utility at the low-end of innovative-design. It appears likely that the applicability of such enhanced DODEs would still be restricted to tame design projects. This is because DODEs do not currently include tools for supporting and representing non trivial design strategies.

The unique feature of Blessing's process based environments, and of PROSUS in particular, is that they are inspired by a conjunction of prescriptive design methodologies, second generation design methodologies and Blessing's first hand observations of designers in action. The analysis of these environments notes that process based design environments contain sufficient second generation and ethnomethodological influences to be of great potential benefit to designers who are engaged in wicked projects. Since wickedness and innovation are close cousins, this section will also note that these environments should be beneficial for designers who are engaged in projects that require a good deal of innovative-design or strategic-design.

However, it was argued that if process based design environments appear well suited to innovative and wicked projects, then there are also good reasons to conclude that they are unsuited to tamer projects in true variant-design and the low-end of innovative design. Two arguments were offered to support this claim. First, PROSUS is based on a representation of design strategies that appears overly general

and hence misses the opportunity to offer specific support to particular prescriptive design methods. Second, because PROSUS is intended to support the entire design process, it is a very complicated definition for a system. It therefore seems likely that designers who are engaged in less demanding projects would appreciate less complicated software.

These arguments indicate that a potential niche exists between the areas of the design activity space which are occupied by DODEs and process based approaches. This niche would be of interest to designers who seek support with tame versions of true variant-design and the low-end of innovative-design.

The prior discussions of DODEs and process based design environments have revealed the existence of a niche for an alternative type of knowledge aided design system. This niche is partially located on the axis of the design activity space which measures innovation. Its domain on this axis lies between true variant-design and the low-end of innovative-design.

As some preceding arguments have suggested, this axis is difficult to separate from the axis measuring tameness and wickedness. It will therefore be suggested that the kind of MODEs which are defined in this section also need to be aimed at addressing tame examples of true variant-design and the low-end of innovative-design (figure 5.6).

Since it is hard to predict whether a given project will fall on one side or the other of the divisions between variant-design and the low-end of innovative-design and tame and wicked problems, for practical purposes this niche might extend a small way in either direction on either of the axes.

Turning to the axis that contains the stages of analysis, synthesis and evaluation, it would be desirable if these stages could be supported by any systems that were proposed to offer support to this niche. Since DODEs offer limited support for analysis and evaluation and good support for synthesis and the process based approach appears able to offer support for each of these stages, it would seem that any new class of system should be able to offer support that is at least equivalent to DODEs across this axis and it would be desirable for such a class of system to offer support that will match the promise of the process based approach.

In the next section a class of systems will be proposed to offer support to design activities which lie within the niche that has been defined here.

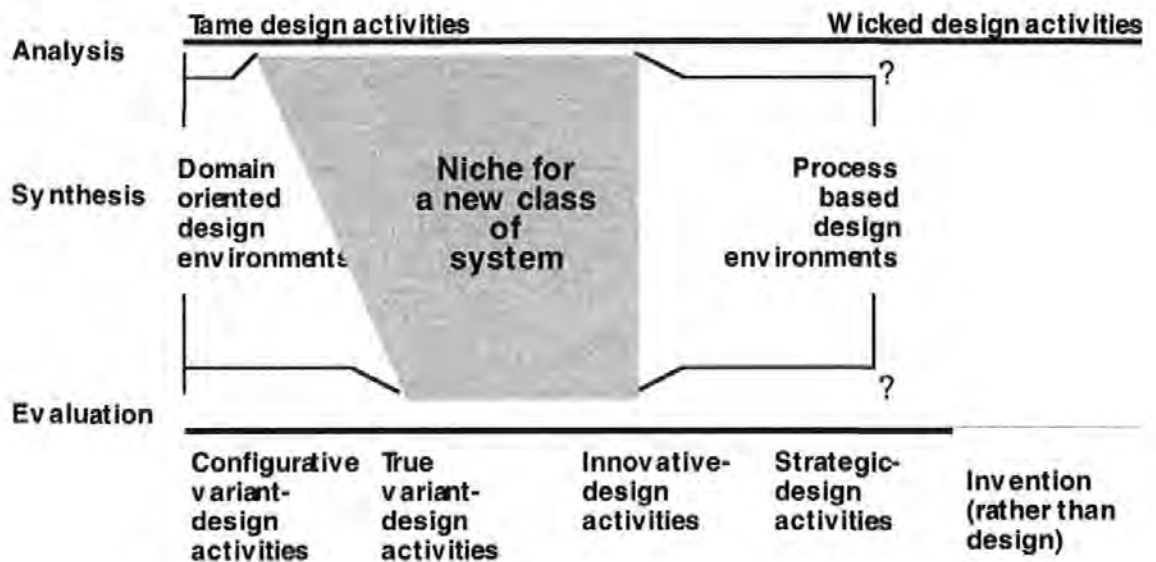


Figure 5.6: This diagram illustrates a niche between the regions in the design activity space that are supported by DODEs and the process based approach.

5.6 The proposal of MODEs

In the previous section a niche within the design activity space was identified. In this section a class of MODEs will be proposed to offer support for activities in this niche. The strategy will be to define MODEs in comparison to DODEs and the process based approach.

The twin criteria for proposing a new class of software are (1) that it will address a need and (2) that it is technically viable. Now that arguments indicating the need for a new class of software have been presented it is time to turn to the issue of inventing a technically viable format.

A technically viable format for MODEs can be supported by referring to the interpretations of the histories of design research, artificial intelligence and knowledge aided design which have been presented in the first half of this thesis. It will be suggested that the way these histories have been presented here allows a slightly more detailed categorisation of the available approaches to developing knowledge aided design systems than those that are currently found in the literature. It will add a third category of system to the existing categories of DODEs and process based environments. This third category can be generated by subdividing the process based approach into its constituent elements of first generation and second generation methods. This finer grained categorisation allows the identification of a

class of MODEs that are domain independent and focused on supporting first generation design methods.

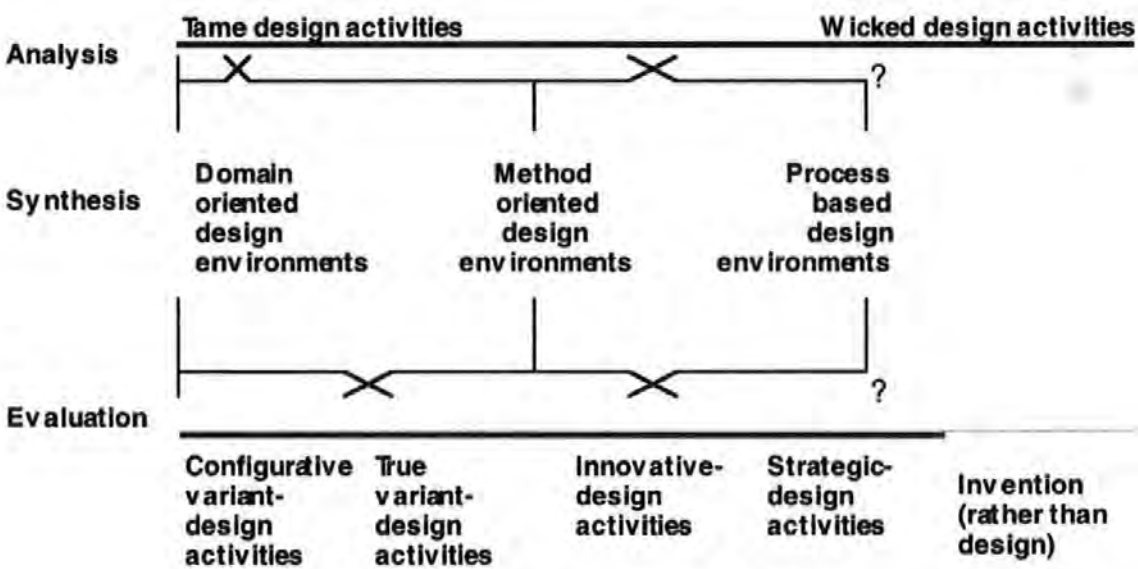


Figure 5.7: This diagram illustrates the suggestion that MODEs should be placed between DODEs and the process based approach on the axes of the design activity space that represent wickedness and innovation.

This section will suggest that these categories should be arranged such that MODEs are placed between DODEs and the process based approach on the axes of the design activity space that represents wickedness and innovation (figure 5.7). The plausibility of such an arrangement is fortunate, then, because it means that MODEs map neatly onto the niche that has emerged throughout this chapter. However, as the conclusion to the section will suggest, MODEs do not appear to be well suited to wicked examples of true variant-design and low-end innovative-design. So MODEs still leave a pressing need for a class of systems which can address the needs of designers working on such design activities.

Since MODEs are being proposed to fill a niche between DODEs and process based design environments, this new class of system will be introduced by comparing and contrasting it to these types of systems.

5.6.1 Comparisons and contrasts between MODEs, DODEs and CITIS

As the term method oriented design environment implies, the key difference between MODEs and DODEs is that the primary source of knowledge within a MODE is derived from a given design method rather than a given design domain. In particular

it is proposed that MODEs should derive this knowledge from first generation prescriptive design methods.

This proposal is made because these first generation methods offer well structured and proven design knowledge which can be depicted as familiar and 'intelligent' interface items. For example, directed and annotated graph are familiar and effective items in both knowledge aided design systems and in design education. Such graphs might therefore replace design-simulators as the central work item in a knowledge aided design system. If they are implemented as passive design-informers or design-aids, then these graphs could act as repositories of strategic design knowledge which designers could either browse through or navigate by. If they are implemented as more interactive design-strategists, then the structures in such graphs could provide logical knowledge to empower processes of mechanical deduction and inference. In either case it appears that the nature and format of the resulting product will not be governed by the knowledge within a MODE to the same extent as it would be by the design-simulator within a DODE. For example, it is plausible that a system to support a method known as parametric analysis (Hollins and Pugh, 1990; Pugh, 1991) could be used to design artefacts, services and systems in multiple domains. This is because the method itself has been proven to be applicable for each of these types of products (Hollins and Pugh *Op Cite*; Pugh *Op Cite*).

The passive design-informers suggested above are quite similar to the approach that Gilleard and Lee applied to the design of the interface to CITIS. However a major difference between CITIS and MODEs would be found in the types of tools which are available at the points in the program that are represented by the nodes within the graphs comprising such passive design-informers. Since CITIS leans heavily toward domain dependence¹ it places domain specific design-experts within the cards representing nodes in its graphs. The applicability of CITIS is hence largely restricted by these tools to a particular type of work item and to configurative variant-design. In a MODE these nodes would be more likely to represent less domain specific tools such as design-visualisers and design-documentors.

Referring to the arguments presented in the previous chapter, in order to support true variant-design and the low-end of innovative-design MODEs would need to include tools from the second group in Green's tool set. In particular both passive and interactive design-documenters appear to be required as well as rudimentary design-secretaries. For example, although passive design-informers have been described as explicit tools it is also possible that they could be made more implicit

¹ CITIS should not be categorised as a DODE under the criteria that have been used in this thesis because it lacks a central design-simulator.

by structuring the interface windows of a system such that each window represents a node in a graph of a prescriptive design method. It is hypothesised that the ability to include such tools within a system that does not restrict innovation to the level of configurative variant-design would help designers to address true variant-design and the low-end of innovative-design.

For these reasons a MODE would be similar in technology to a user-programmable DODE with the obvious difference that the main item being represented in the interface is a method rather than a product.

5.6.2 Comparisons and contrasts between MODEs and the process based approach

The main differences between MODEs and the process based approach are not related to particular design domains nor to the restrictions to creativity caused by design-simulators. Rather these differences reflect the distinctions between first generation and second generation design research. Most specifically they reflect the distinction between the ways that the prescriptions found in first generation research and the advice found in second generation research can be represented using diverse media such as text, graphics or logic.

The previous subsection discussed the graphs that can be derived from prescriptive design methods. However, since these graphs play an even more important role in this subsection the process of their derivation will be rehearsed in slightly more detail.

A definitive feature of prescriptive design methods is that they can be represented as causal networks. In Miles and Huberman's (1994) terminology, a 'causal network' is a data structure that presents a display of the most important independent and dependent variables in a field study (shown in boxes) and of the relationships between them (shown as arrows). It should however be noted that not all first generation design methods are based on field studies (Jones, 1970).

In the context of prescriptive design methods such causal networks show designers how to transform a given set of inputs into a design in a way that is repeatable, goal directed and which offers a reasonable possibility of success. It is immaterial whether such networks are expressed as text, graphs or logical relationships because design researchers can, in principle, translate back and forth between these languages of design.

In order to transform a prescriptive design method into knowledge for a knowledge aided design system, then, it is important that a causal network can be

derived from the description of the method. This means that design researchers and system developers can either express the network as an annotated graph, which would provide the structure and labels for a passive design-informer, or as logical steps and relationships, which would provide the knowledge for a design-strategist.

Second generation methods are much less likely to be expressible as useful causal networks. This is because second generation design methods are addressed towards wicked design processes and it is axiomatic that these are less predictable than first generation methods. Therefore a causal network which describes a second generation method is likely to be less detailed and more contradictory than a causal network depicting a first generation method. It is also far less likely to provide reliable structural design knowledge. This lack of goal directed and structural knowledge implies that it would be hard to equip the interface to a process based design environment with a detailed map of its design process and equally hard to identify rules of inference upon which to base a design-strategist .

Unless new insights into the nature of wicked design projects are uncovered, this argument suggests that Blessing's use of a design matrix in PROSUS can be generalised as the most sensible approach to including structured design knowledge in process based design environments. However, as the arguments presented in the previous section concluded, these kinds of representations seem much too general to be useful to designers who are engaged in tame problems because these can be addressed with first generation design methods.

Given the current understanding of wickedness in design research, then, it can be concluded that different types of knowledge are available for tame and wicked problems. This has the consequence for MODEs that sufficient strategic knowledge should be available in a first generation design method to develop detailed passive design-informers and useful design-strategists. The consequence for process based design environments is that sufficient strategic knowledge seems to be available to build very generalised passive design-informers but insufficient strategic knowledge is available to develop more detailed design-informers or design-strategists.

In theory, then, MODEs and process based environments appear to be fundamentally different types of knowledge aided design system. In practice, however, the tame and wicked projects tend to be hard to distinguish and individual projects tend to have tame and wicked elements. This means that tame and wicked projects can be regarded as extremes on a scale rather than as entirely separate types of projects. It seems likely that if knowledge aided design systems are developed which address tameness and wickedness in design, then each system will be biased toward an extreme on this scale rather than being an example of a fully tame or fully

wicked system. Thus MODEs and process based design environments can be categorised as responses to tameness and wickedness which have more in common with one another than either has in common with DODEs. This raises the possibility that users or developers could embed MODEs within process based design environments and vice versa.

5.6.3 An opportunity for further research

To conclude this proposal of MODEs, it should be stressed that the preceding arguments have an implication for knowledge aided design systems which are aimed at wicked projects in true variant-design and the low-end of innovative-design. If a system aims to support a second generation design method, then its technologies should ideally allow its users to alter any parts of the system that represent that method. Since the proposal for the class of MODEs does not include features that exhibit such functionality, it should be stressed that MODEs are constrained to supporting tame design projects. The task of finding ways to include the specificity of first generation methods and the adaptability found in the proposal for PROSUS can therefore be identified as a future research goal for the subject areas of design research and knowledge aided design.

5.7 Chapter summary and conclusions

The work reported here has addressed the second and third of the hypotheses. In particular the mappings developed in Chapter Three have been used to estimate the range of utility for some recent systems for knowledge aided design. This completes the work performed in response to the second hypothesis. A search has also been performed in the design activity space for a region that is not supported by existing types of system. The class of MODEs has also been proposed. This completes the work performed in response to the third hypothesis.

In the previous chapter it was stressed that many of the arguments presented in the first half of the thesis are motivated by extrapolation from theory rather than empirical observation. This is due to the fact that few knowledge aided design systems have been implemented which explore any regions of Green's model space beyond the familiar territory of the design-expert. Clearly, then, more systems need to be implemented and evaluated in order to test and refine the suggestions that have been made here.

In response to this requirement for knowledge aided design systems that explore new regions of Green's model space, the second half of the thesis will be devoted to developing and implementing a MODE. The development of Patina ~ a MODE for parametric analysis is therefore in Chapters Six and Seven and the work reported in these chapters will address the fourth hypothesis.

Chapter Six

The development of Patina: triggers and analysis

6.0 Overview

The work reported in this chapter begins the process of addressing the fourth hypothesis:

- H4** It is possible to design and implement an example of a MODE by locating a design practice in the design activity space and then applying the mapping to Green's model space in order to propose an aggregate of components that will support this practice.

In this chapter parametric analysis (Hollins and Pugh, 1990; Pugh, 1991; Hollins and Hollins, 1991) is described, located in the design activity space and analysed in search of elements that could benefit from support with a MODE. The process of mapping these elements onto knowledge aided design tools and the implementation of Patina itself is reported in Chapter Seven.

The first section of the chapter describes the triggers which led to the implementation of Patina as a MODE for parametric analysis. These include:

- (1) the need to demonstrate the utility of the design activity space
- (2) the utility of the coupling between the design activity space and Green's model space
- (3) the need to implement an example of the class of MODEs which were defined in Chapter Five.

Parametric analysis has been chosen because it is an example of a design practice that is located in the region of the design activity space that is to be supported by MODEs.

The chapter contains five main sections. In the first section the organisational contexts in which parametric analysis is likely to be used are explained. In the second section the purpose of performing a parametric analysis is motivated. In the third section the kinds of knowledge parametric analysis has been shown to produce and the types of products and services it can survey are discussed. In the fourth section parametric analysis is located within the design activity space to show that it falls within the niche

which MODEs have been proposed to support is. In the fifth section parametric analysis is examined in search of elements which might benefit from knowledge aided support.

6.1 The context of parametric analysis

This section begins the task of introducing parametric analysis by discussing its context of use, which is in structured and proactive product design or service design¹. Product design has been defined to as covering “the spectrum from fashionable and ephemeral consumer items that are generally the subject of discretionary purchases, to essential personal, business and capital purchases that involve longer term commitments of significant sums of money” (British Standards Institution, 1995, p 1). Service design has been defined as covering “the whole spectrum of services. This includes those paid for directly by the recipient, such as accountancy, banking and law, as well as those not paid for directly, such as the Civil Service, social services, emergency services and charitable services” (British Standards Institution, 1995, p 1). The focus here will be upon the proactive design of commercial products.

Commercial products tend to follow a generic cycle of introduction, growth, maturity and decline within their markets (see figure 6.1). The stages in this cycle tend to generate corresponding levels of low, medium, optimal and declining sales-volumes and revenues (Hollins and Hollins, 1991). An organisation which markets a line of products and which aims to remain profitable should strive for a corporate design programme that develops and introduces new products before the existing products reach their saturation points. Approaches to solving this problem can be divided into opportunistic and proactive strategies. These competing strategies will therefore be briefly described in order to set parametric analysis into the context of proactive design.

The opportunistic design strategy trusts that people will experience unforeseen inspirations, i.e., ‘eurekas’, which they then will bring to an organisation to be developed and marketed into products. While it would be foolish for organisations to dismiss the windfalls that can be generated by such ‘eurekas’, a pure opportunistic design strategy is particularly risky in the long term. This is because ‘eurekas’ are unpredictable and therefore cannot be relied upon to provide a steady stream of new products to be developed and brought to market (Druckner, 1985).

¹ For the sake of brevity the term product will refer to products and services in this discussion.

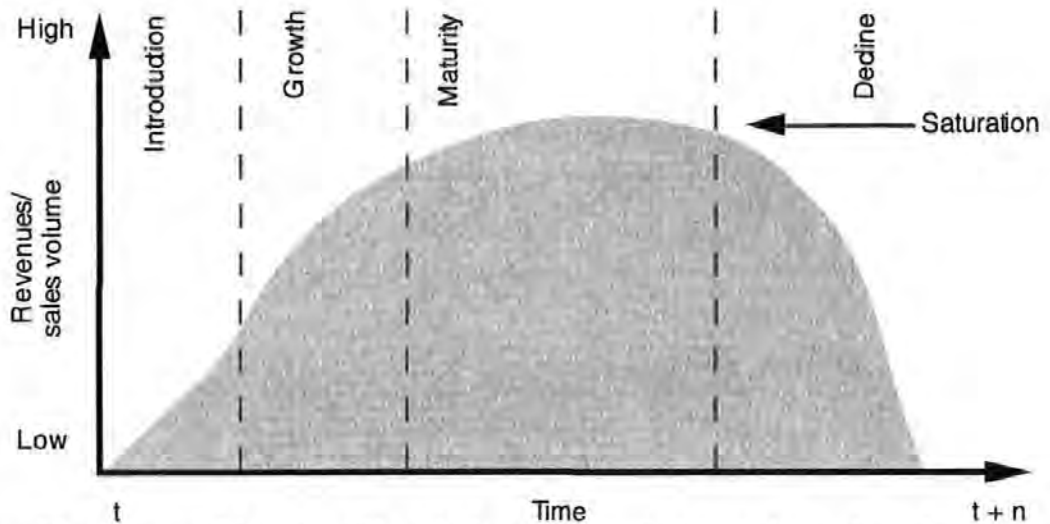


Figure 6.1: A diagram of the product life cycle showing that products have phases of introduction, growth, maturity and decline. The shaded region shows the expected revenues generated during these periods (re-drawn from Hollins and Hollins, 1991, p 93).

Proactive design is an alternative strategy, which is influenced by Edison's concept of an 'invention factory' (Mitcham, 1994). In an organisation that practices proactive design, people are employed to systematically research and develop a 'conveyer belt' of potential products. Although the inherent unpredictability of human innovation and creativity still exposes the sponsor of a proactive design department to some unavoidable risks, this strategy is a rational response to the demands of competitive and volatile markets. With luck, the adoption of proactive design will help an organisation to maintain a product line with representative products in each of the phases of introduction, growth, maturity and decline.

The preceding comments indicate that a mixed strategy of opportunistic and proactive product development is probably desirable. However, the description of parametric analysis presented in this chapter will presume that a proactive approach is being followed.

6.2 The purpose of parametric analysis

Stuart Pugh argued that it is important to begin proactive product design by researching the existing conditions of an intended market, including the set of products that are currently aimed at that market (Pugh, 1991). Such research, he argued, should help to ground specifications for new products on empirical analysis rather than *a priori* assumptions.

Deciding to conduct such research is a prudent strategy considering the high failure rates for new products as well as the high cost of developing and marketing them

(Hollins and Hollins, 1991). As a means to address this goal, Pugh developed a prescriptive design method called parametric analysis. This has been described as an: “inexpensive form of desk research that has proved to be a powerful tool for both marketing and engineering. It is used to identify a product’s place in the market in relation to the competition and also to gain insight into the make up and interrelationships between the parameters inherent in the product under consideration” (Hollins and Pugh, 1990, p 98).

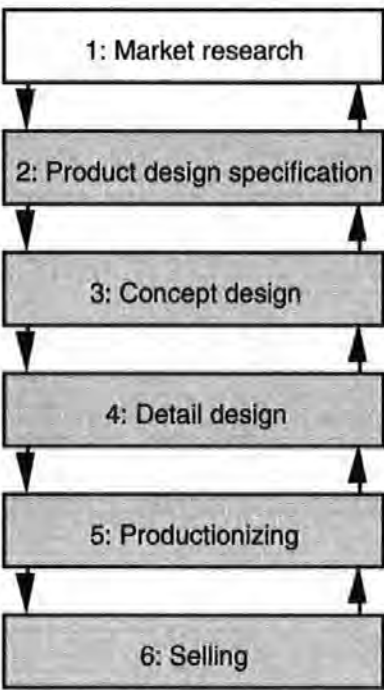


Figure 6.2: A diagram of the Total Design method highlighting the stage of market research.

This description emphasises the fact that parametric analysis is aimed at the development of commercial products which are in competition with a range of comparable products. Incidentally, the term ‘product’ will be used to refer to both goods and services.

Pugh offered parametric analysis to designers as a probe with which to gather information at the outset of the Total Design methodology. This prescriptive methodology consists of a sequence of six iterative stages for: (i) market research, (ii) product design specification, (iii) concept design, (iv) detail design, (v) manufacture or productionizing and (vi) selling (see figure 6.2). Parametric analysis is mainly performed during stage (i) and its results can then referred to during stages (ii) to (vi). The prescribed means of performing an analysis and of storing its results will be described in section 6.3.

Pugh tested his method in the domain of car design during the oil crisis of the 1970s. By using the parametric analysis method Pugh demonstrated that 'miles per gallon' was an increasingly important feature of cars during this period (Dr. William Hollins interviewed by the author in 1995). This result indicates that the method is capable of confirming 'common sense' intuitions involving how political and economic factors can influence engineering design decisions.

Further exploration of the method was performed by Dr William Hollins in his thesis *Product Status and the Management of Design* (1989). Here Hollins investigated various brands of four wheeled land vehicles. These vehicles performed in application domains which included usage in agriculture, industrial and the military. Hollins found that parametric analysis clustered the vehicles in such a fashion that he could readily identify the target market for particular vehicles (Dr. William Hollins interviewed by the author in 1995). He also found that parametric analysis could be used to indicate that a particular brand of vehicle was so statistically unusual that it must have incorporated a new technology. Upon further investigation with the manufacturer of that brand, Hollins found that the product had been withdrawn due to worries about its safety. He has suggested that a parametric analysis performed early in the product's development might have led to similar conclusions before such a significant amount of development cost had been committed to the project (Dr. William Hollins, *Op Cite*).

Hollins has also argued that parametric analysis can be used to track two strategies for developing products. In the first of these strategies, products tend to emulate the success of a market leader by adapting all of their parameters toward its profile. Hollins has emphasised that tracking the second strategy needs to be performed and interpreted in a conservative fashion and periodically re-examined. This is partly because technological trends tend to reach a plateau after a given period of time. Therefore the runaway evolution of certain parameters will tend to give way to the rapid evolution of other parameters. However, given that caveat, such an analysis can be used to spot interesting or 'useless' products. Respectively, such products are probably either occupying a niche away from the market leader or their designers are missing a particularly important feature of the market (Dr. William Hollins, *Op Cite*).

In summary, the work contained in Hollins' thesis shows that parametric analysis can help a designer to uncover technical information which extends beyond immediate common sense perception of a problem.

During the 1990s Hollins has disseminated parametric analysis in works co-authored with Pugh and with Gillian Hollins (Hollins and Pugh, 1990; Hollins and Hollins, 1991). This later work, entitled *Total Design: Managing the Design Process in the Service Sector* (Hollins and Hollins, 1991), extended the domain of parametric analysis from the

engineering sector to the service sector. Hollins currently teaches the method as part of his activities as a consultant. Parametric analysis has also entered the curriculum at various British Universities, such as the University of Brighton (Dr. John Downie, personal communication). In the later 1990s, then, parametric analysis is firmly embedded within the literature and practice of British Engineering education. It is hard to estimate how far the method has penetrated into industrial practice because organisations tend to prefer to keep such information as secret as possible. However, since the method is taught to students of design and to working designers, it seems plausible to assume that it is either currently used in industry or could be given appropriate managerial support.

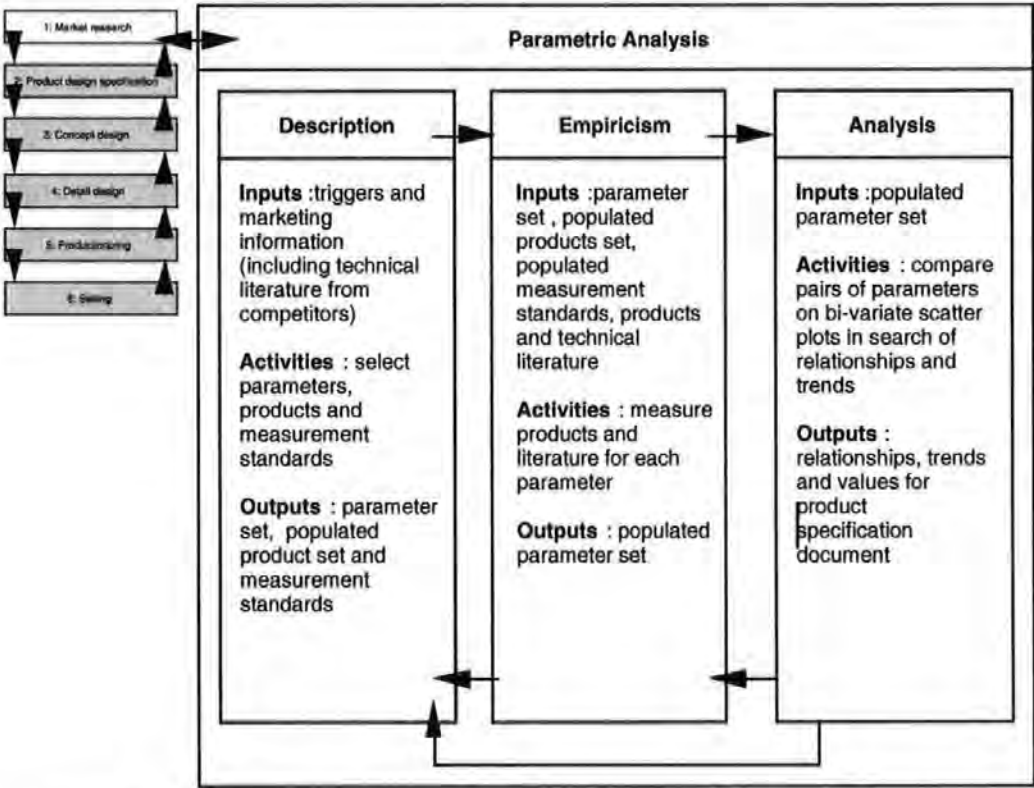


Figure 6.3: A diagram showing the three iterative stages of parametric analysis embedded in the market research stage of the Total Design method.

6.3 The content and output of parametric analysis

The previous section described the purpose of parametric analysis and indicated some of the types of information that it can reveal. This subsection will turn to the subject of how parametric analyses are performed. Its content is largely a rehearsal of the descriptions of the method to be found in Hollins and Pugh (1990), Hollins and Hollins, (1991) and Pugh (1991). These descriptions of the method are also supplemented with reports of

some slightly less formal observations made by Hollins during an interview conducted by this author in 1995.

Pugh’s prescribed method for conducting parametric analyses, can be separated into three iterative stages: description, empiricism and analysis (see figure 6.3). An individual parametric analysis can take two slightly different directions: ‘parameter against parameter’ (PAP) and ‘changes over time’ (COT). The current prototype of Patina supports PAP rather than PAP *and* COT. Hence the discussion in the remaining sections of this chapter will be targeted upon PAP rather than COT¹.

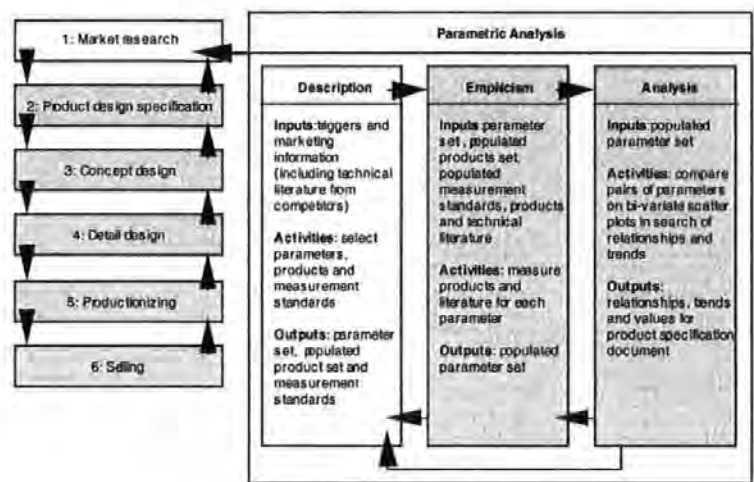


Figure 6.4: A diagram highlighting the prescriptive stage of parametric analysis in the context of the Total Design method and the empirical and analytical stages.

From the perspective of second generation design methods, an important part of the preparation for an analysis involves choosing a designer or team of designers who will perform the analysis. Predicting which areas of expertise will be required and the ideal size for a design team are problematic. Hollins has suggested that in his experience an individual designer is the best choice (Dr. William Hollins interviewed by the author in 1995). However as the wickedness or innovation required by the analysis increases, an individual designer may not be able to bring sufficient cross disciplinary expertise to bear upon each aspect of a product. Moreover as this wickedness or need for innovation increases, it becomes less likely that the areas of expertise that will be eventually be required can be predicted at the outset of an analysis. Goldschmidt’s comparative study of individual designers and design teams has argued that individual designers play most of the roles associated with teams of designers and hence constitute a ‘team of one’ (Goldschmidt, 1995). However it should be noted that working in teams raises issues

¹ Briefly, the COT variant of the method involves graphing values for products across a finite time scale to investigate and predict their responses to market trends. See Hollins and Hollins (1991) for descriptions of COT

associated with organisational design (Tjosvold, 1991) as well as patterns of group communication (Galegher *et al*, 1990). The different ways that groups of designers have been observed to interact during stages of idea generation, evaluation and refinement also bear upon the decision to work alone or in a group (Crickmay and Jones, 1972). For these reasons the following description of parametric analysis will refer to a design team but the reader may assume that this phrase refers to one or more designers.

The first stage of parametric analysis can be called the descriptive stage (see figure 6.4). This consists of inventing and / or selecting sets of products and parameters to model a class of existing products as well as some parts of their environment. To begin an analysis, then, the design team must select a number of existing products in the class they plan to investigate. This can be known as a product set. Once its contents have been chosen it can be thought of as being populated with products. The designers must also identify a set of salient features which determine a class of products. This is known as the parameter set and it may contain a mixture of objective parameters, such as 'temperature' and more subjective parameters, such as 'warmth'. It is important to ensure that an analysis is not missing any of the publicly available information about a class of products. One way to address this goal is to include the types of parameters that are found in manufacturers' catalogues, magazines, and journals devoted to the market under analysis. Table 6.1 presents an array of parameters that Pugh and W. Hollins would associate with a standard engineering product. As a further example, Hollins and Hollins (1991, p 60) add parameters such as 'entertainment / fun' for a service sector product. Similarly, 'investment risk' might be added for a financial product.

Ensuring that a parametric analysis of a product includes generic and subjective characteristics of the products has two immediate advantages. First, it should be easy to find values for these parameters for a selection of existing products without incurring the expense of actually measuring samples of the products themselves (although this process may still be necessary for parameters which are less 'run of the mill'). Second, as Hollins added, these are the parameters which are known to be of interest to customers, magazines and journals (Dr. William Hollins interviewed by the author in 1995). Therefore concentrating at least part of the effort of the analysis on them is likely to enter the designers into a beneficial feedback loop. In theory entering such a feed-back loop can result in products that pander to the known interests of the market. This strategy is not guaranteed to generate innovative products but it might help to avoid what Hollins terms 'useless products' (Dr. William Hollins, *Op Cite*).

Sales potential	Patents	Shelf life
Quality	Competition	Maintenance
Installation	User training	Packing
Weight	Market constraints	Manufacturing facility
Shipping	Size	Process
Customer	Time scale	Product cost
Performance	Ergonomics	Aesthetics
Quantity	Materials	Product life span
Politics	Energy consumption	Disposal
Company constraints	Safety	Legal
Testing	Environment	Etc....

Table 6.1: A set of standard parameters that would be expected in a PDS for an engineering product (from Hollins and Hollins, 1991, p 59).

At this point in the evolution of the analysis it is important to ensure that the products in the product set are comparable across all of the dimensions in the parameter set according to some formal criteria. As Hollins and Pugh put it, designers should “ensure that the data being used relates to comparable products at the same generic level” (Hollins and Pugh, 1990, p 101). For example, it would be problematic to compare a compact disc player against a record deck if one of the parameters is based on the width of a laser beam. On the other hand a set of parameters might be proposed that allow these different technologies to be compared as examples of a product set of ‘entertainment products’.

Once the designers have identified initial parameter and product sets they must populate the parameter set with empirical data derived from the products and their interactions with the market. So the second stage of a parametric analysis can be called the empirical stage (see figure 6.5).

During the empirical stage each product is inspected to determine a value for each parameter. For quantitative parameters, such as ‘temperature’, these values can be measured objectively, using appropriate numeric conventions. However, setting values for qualitative parameters, such as ‘subjective warmth’, requires more human intervention. Therefore, designers might select a group of subjects, such as customers or end users, to subjectively grade the qualitative parameters of each product. This presents the designers with the task of translating those perceptions into quantitative values. Once this stage is completed the parameter set can be thought of as being (initially) populated.

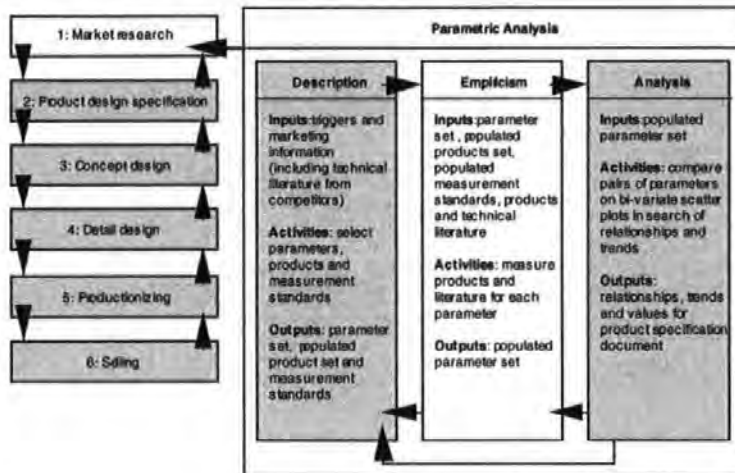


Figure 6.5: A diagram highlighting the empirical stage of parametric analysis in the context of the Total Design method and the descriptive and analytical stages.

The third stage of parametric analysis can be called the analytical stage (see figure 6.6). The purpose of this stage is to uncover relationships between pairs and / or groups of parameters, to name these relationships and to transfer the more important of them to the PDS.

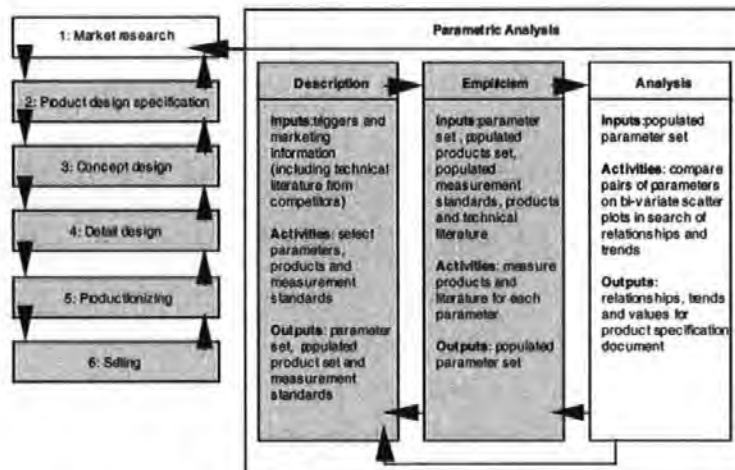


Figure 6.6: A diagram highlighting the analytical stage of parametric analysis in the context of the Total Design method and the descriptive and empirical stages.

To prepare for the analytical stage, the designers must quantify their data and exhaustively graph the populated parameters against one another in bi-variate scatter plots. One bi-variate scatter plot is required for each pair of parameters in the parameter set. During the analytical stage the designers 'mine' the data for 'seams' of knowledge. In other words, they are instructed to apply a combination of statistics, domain knowledge and intuition to find trends and other relationships between the parameters in

their scatter plots. For example, a designer might observe that a set of products nestle along an envisaged line in the graph of two parameters and then interpret this line as the border of a constraint. Designers are also encouraged to “view [the plots] collectively since this usefully establishes further missing logic and gives rise to explanations that are then seen to be obvious” and advised that “grouping of plots, incorporating the same parameters, is always useful and revealing” (Hollins and Pugh, 1990, p 101).

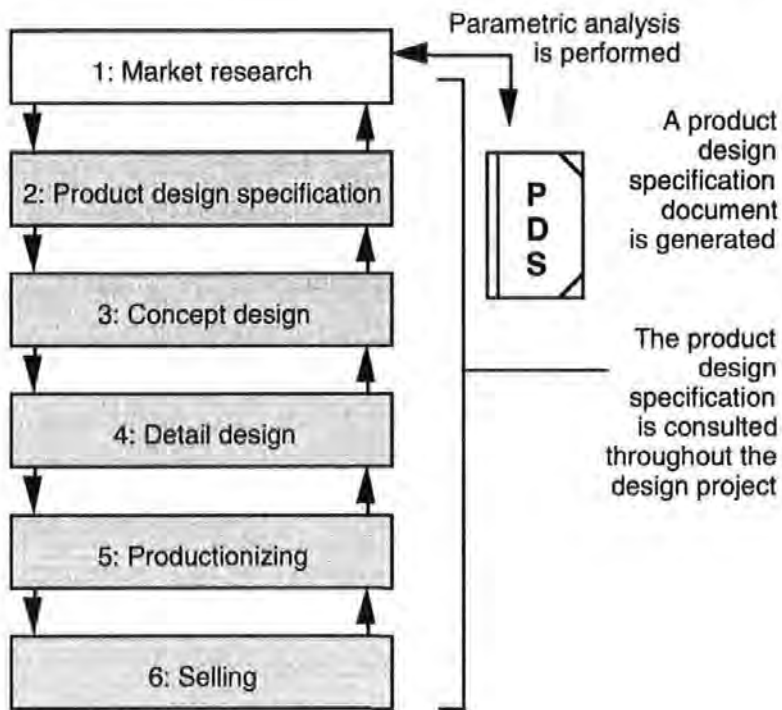


Figure 6.7: A diagram of the Total Design method showing the performance of a parametric analysis, the generation of a product design specification and the stages (in grey) where it is consulted.

Although the official description of parametric analysis encourages designers to plot each of their parameters against one another, Hollins suggests that a certain amount of common sense is required in selecting parameters to plot. This is because an exhaustive cross-plotting of each of the parameters requires *size of parameter set (size of parameter set-1)/2* plots. This requirement for common sense implies that at least one member of the design team performing a particular parametric analysis should be familiar with the type of product they are designing. Given Crickmay and Jones’ previously cited prescription “use many heads to generate a variety of ideas and one head for simplifying insights” (Crickmay and Jones, 1972, p 21) it might be sensible for this domain specialist to be responsible for interpreting the output of this analytical stage.

Once a number of relationships have been found it is necessary to store them in an appropriate format so that they can contribute to the subsequent stages of the Total Design method. As discussed in Chapter Three, prescriptive design methods can contain structured artefacts which help designers to orchestrate their activities (either alone or in teams). One such artefact contained in the Total Design methodology, whose content is influenced by parametric analysis, is a document known as a Product Design Specification (PDS) (see figure 6.7).

A PDS can be described as a ‘note book’ in which, most importantly, each page sets a goal and defines a reasonable objective for a parameter standing for an aspect of a particular product (see figure 6.8). Other ‘fields’ contained on these pages include a best possible value for the parameter, a best existing value (the market leader) and a set of relevant notes.

For example, in the case of a product manual for our imaginary compact disc player, a parameter might stand for ‘readability’, a goal might be ‘understandable by people educated to undergraduate status’, and an objective might be numerically defined by a particular Flesch Reading-Ease Score. This score would then serve as an upper boundary upon the readability of the text. In this example, other columns on the page might contain the ideal readability score and the readability score of a leading product. The notes might mention the fact that a Flesch Reading-Ease Score is an error-prone yardstick and provide examples of the causes of such errors (See Harris, 1996).

Product specification document			
Product Compact disc player manual			
Parameter Readability (Flesch Reading-Ease Score)			
Competition best	Current model (ours)	This design (intent)	World class (target)
10	11	10.5	9
Description			
Readability can be measured by the Flesch Reading-Ease Score. However, this can be misleading because			
Page: 1			

Figure 6.8: Three pages from a hypothetical PDS (adapted from Pugh, 1990).

Taken as a whole, the pages in such a notebook represent a set of goals and objectives that define the envisaged product. Although the parameters found within a PDS will vary according to the type of product being designed, various standard parameters can be expected for standard types of product (see table 6.1).

Once a PDS has been generated it serves as a shared, objective resource which a designer or design team can consult throughout stages (ii) to (vi) of the design project. The PDS can be regarded as a 'virtual' description of the envisaged product. This is because it is first drafted before the conceptual and detail design stages. So it cannot specify how its goals and objectives will be achieved. However, to have value, it should provide relevant and realistic goals to be addressed during these stages; goals and objectives which, as Pugh argued, should have been informed by an empirical and analytical method. Hence, an important feature of parametric analysis is that it can be performed before resources have been wasted on designing, producing or selling concepts or details that address a misconceived opinion of a market's needs. Then, once the analysis has been performed, designers can integrate its output into the PDS.

6.4 Placing parametric analysis in the design activity space

Now that the context, purpose and content of parametric analysis have been introduced, the next task in this analysis is to explore how well the method can be expected to perform for various types of design tasks and activities.

This exploration will discuss the likely performance for two variables that measure the effectiveness and efficacy of the method. This performance will be measured along the three dimensions of the design activity space. To recap, these dimensions stand for: (i) the analytic, synthetic and evaluative stages of a design activity; (ii) the amount of innovation required by a project, and (iii) the amount of tameness and wickedness of a project.

The section has two aims. The first aim is to identify some regions within this space where parametric analysis would appear to offer high values for effectiveness and efficiency¹ (subject to the caveat that the effectiveness and efficiency of a particular parametric analysis can only be assessed by participants and stake-holders). The second aim is to predict the kinds of knowledge that performing a parametric analysis might generate along different axes of the design activity space.

Commercial designers face a default objective of increasing the cost-effectiveness of their activities, i.e., the "ratio of the value received and the cost of resources expended" (British Standards Institution, 1995). In other words they need to constantly maximise

¹ This particular terminology has been adopted from Blessing (1994).

the expected quality of their envisaged product and minimise both the time-to-market and the costs associated with its design (Cross, 1989). Ideally, then, the performance of a parametric analysis could be measured with a quantitative ratio of two variables: efficacy and efficiency (see Blessing, 1994). Efficacy would stand for the benefit of performing the analysis and its contribution to the quality of the product. Efficiency would stand for the costs incurred both in terms of labour and resources (such as the purchase of research materials).

Unfortunately, there is a sound reason why this ratio can only be used as a likely performance indicator, that is as a heuristic, rather than a formally quantifiable metric. With reference to the arguments of Rittel and Webber (1973) and Rittel *et al* (1972) the perceived value of a design decision may conflict for the same or different people at different times of measurement. In the terminology discussed in Chapter Three, then, estimating the efficacy of a particular result from a parametric analysis is a wicked problem in its own right which can involve a broad range of stake holders. Accordingly, it can be dangerous to place undue faith in a unitary cost-benefit analysis. This is because it can be difficult to ensure that each issue which is relevant to each stake holder is well represented and quantified from each perspective. In the following discussion it would be most rational to recognise that each stake holder in a parametric analysis seeks to maximise the ratio between their own conception of the correct content for the efficacy and efficiency of the analysis, and to bear in mind that some of these conceptions might conflict. Unfortunately, given some of the observations about the commercial realities of participatory design explained in Chapter Two, it would also be sensible to conceive of this as an optimistic aim in a commercial environment.

Given this caveat, the type of new knowledge that might be found by a highly efficient parametric analysis can be divided into two broad types, i.e., 'design and manufacturing' knowledge and 'cultural and marketing' knowledge. Since this division is broadly possible, and since these types of knowledge have different values to different types of designers, this analysis will focus on how feasible it will be to discover each type of knowledge along the various dimensions of the design activity space.

6.4.1 Parametric analysis and the first dimension of the design activity space

The first dimension of the design activity space contains the analytic, synthetic and evaluative stages of a design project. Clearly parametric analysis will most often be used during the analytic stage of a project where it can be expected to uncover 'design and manufacturing' knowledge and / or 'cultural and marketing' knowledge depending on the products under consideration. However, as was argued in Chapter Three, each of the

three main stages of design tends to contain elements of the other. Thus it is very likely that parametric analysis might be revisited during the synthetic and evaluative stages of a project. For the current project, then, the main utility of this particular axis of the space of design activities will be for the choice of tools with which to implement a program to support parametric analysis rather than for the type of knowledge that an analysis will generate.

6.4.2 Parametric analysis and the second dimension of the design activity space

The second dimension of the design activity space measures the amount of innovation required by a project. To recap, the content of this dimension was synthesised from prior work by Brown and Chandrasekaran (1989) and Culverhouse (1995). The categories represented on the axis are: repeat orders, configurative variant-design, true variant-design, innovative-design and strategic-design.

Brown and Chandrasekaran (1989) have argued that variant-design is the most frequently attempted, yet least emotionally and intellectually rewarding, class of design problems. They also argued, incidentally, that variant-design is the most promising domain for knowledge-based design. The reason that supports both of these arguments is that in variant-design “all of the design goals and requirements are fully specified, sub components and functions are already known, and knowledge sources already identified” (Brown and Chandrasekaran, *Op Cite*, p 35). This reasoning implies that variant-design can often be regarded as either a glass box domain, a fairly tame domain, or frequently as both.

The problem with this argument is that it is too coarse grained and seems rooted in the cognitive psychology/expert systems view of design (e.g., Simon, 1969), which was criticised in Chapter Two. What Brown and Chandrasekaran treat as a uniform set of problems can be divided into configurative variant-design, which closely matches Brown and Chandrasekaran’s conception, and more realistic true variant-design, which calls for some goal setting, requirements specification, sub-component and function re-design and the identification of new sources of knowledge. In sum both of these forms of variant-design involve ‘tweaking’ rather than full scale innovation but in the former case this ‘tweaking’ applies to parameters within a well defined problem and in the latter it involves ‘tweaking’ parameters that define the problem. True variant-design, then, is where innovation and wickedness start to enter the world of design.

Some implications of this reasoning for the effectiveness of parametric analysis and the type of knowledge it might generate in the setting of variant-design are as follows.

When a design domain is already reduced to a finite set of rules, that are sufficient to enable mechanistic solution of design problems in that domain, designers should not expect further desk-research, in the form of parametric analysis, to uncover significant amounts of new technical knowledge. What they might expect to acquire, however, is new marketing knowledge, especially through performing a COT analysis in a volatile market. In the case of configurative variant-design products, then, it might be sensible to bias the parameter set away from parameters that refer to the product's configuration or construction and toward parameters that represent a product's changing relationship with human culture.

The space on this axis which includes true variant-design and low-end innovative-design is potentially the most promising domain for parametric analysis. There are four reasons to support this assertion. First, designers can confidently assume that a competing set of products exists and they can hope that some of these products will contain some newly introduced features (as the previous subsection suggested, this is an essential pre-requisite for parametric analysis and it is less likely to be true for strategic design). Second, it is possible that some of the interactions between the components and manufacturing processes of these products have not yet been noticed or understood. Third, if the product set contains recently released products it is probable that interactions between these new products and the marketplace have not yet been noticed or understood. Fourth, since the products that are being analysed are in competition, it is also plausible that some of these interactions have been noticed, understood and exploited, as well as kept duly secret by their sponsoring organisations.

In the case of true variant-design to low-end innovative-design, then, it might be sensible to begin with a parameter set that is evenly divided between technical and marketing parameters. Then, as the analysis progresses, the divisions of parameters can be swayed toward whichever type of knowledge is being uncovered.

The expected outcome of performing parametric analysis in strategic-design domains would be similar to that of high-end innovative-design. This observation is supported by the somewhat circular reasoning that the outputs of strategic-design can be regarded as high-end examples of innovative design soon after they are released into the marketplace. However, there is an important reason why the method is not particularly well suited to this domain. This is because if a product is truly strategic then it is unlikely that there will be many competing products with which it can easily be compared.

In the cases of strategic-design and high-end innovative-design, then, it might be sensible to begin with a parameter set that is also evenly divided between technical and marketing parameters. Once again, as the analysis progresses, the divisions of

parameters can be swayed toward whichever type of knowledge is being uncovered. More importantly, though, it is very possible that performing parametric analysis as part of a strategic-design project will require the designers to invent radically new parameters that can be used to unite products with different technological profiles. This appears to be a wicked process in its own right.

6.4.3 Parametric analysis and the third dimension of the design activity space

The issues which are relevant to strategic-design bring this analysis to the third dimension of the design activity space. This axis measures the amount of wickedness involved in designing a particular product and more specifically in designing its design process.

Although the differences between innovation and wickedness are hard to separate, for present purposes innovation can be viewed as a quality of the artefact or service being designed and wickedness can be viewed as a quality of the design method or process that engendered it. It is worth commenting that the style of argument employed here and throughout this subsection is rather teleological, that is, it assumes that the world of design is orderly and rule-governed. This contradicts the findings of studies such as Bucciarelli (1988, 1994), Latour (1996) and Valkenburg and Dorst (1998) which conceive of design as being more empirical and governed by the shifting personal perspectives of designers and other stake holders. Moreover, it would be misleading to suggest that particular projects demand a certain amount of wickedness at their outset. On the contrary, wickedness is probably better viewed as a *post hoc* concept for differentiating past projects. The comments below are therefore presented with the caveat that, for practical reasons, they attempt to summarise a wealth of design activities and ways of characterising them which, for more philosophical reasons, cannot truly be enumerated or described in a neutral language.

Assuming that wickedness in design is almost synonymous with the concept of 'designing designing' (Jones, 1991), then the problems associated with approaching tame design problems via parametric analysis are similar to those associated with configurative variant-design. There seems little design knowledge that is likely to be learned from the products of well-understood design processes. However, as problems begin to involve true variant-design and low-end innovative-design, there might be more design-strategy knowledge embedded in the artefacts which could be recovered. As more wickedness is encountered there is less likelihood that sufficient products will be available to help the design team to deduce how their competitors redesigned their own design processes.

These arguments support the conclusion that parametric analysis might uncover some useful design and manufacture knowledge in an area of the third axis which is quite similar to that of the second axis. Therefore the space which includes the high end of tame problems and the low end of wicked problems seems to be one in which some knowledge about design methods, processes, and strategies could be recovered.

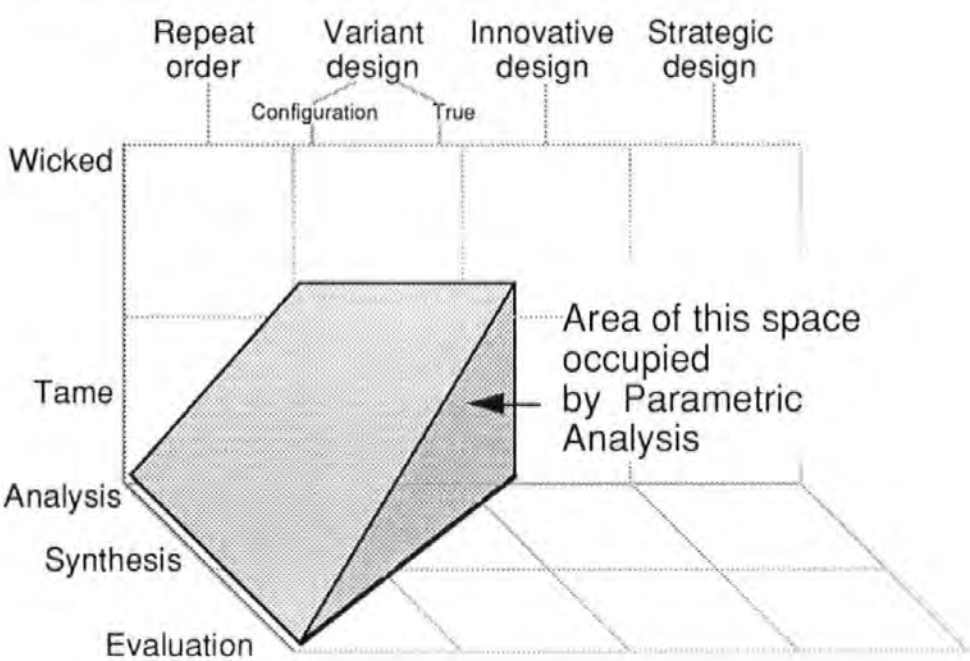


Figure 6.9: The three-dimensional space of design activities which was defined in Chapter Three.

In summary the arguments presented here indicate that parametric analysis could be efficient and effective in the volume of the design activity space shown as a shaded region in figure 6.9. This means the method covers the stages of analysis, synthesis and evaluation (although there is a bias toward analysis), the range from variant-design through to the low end of innovative design, and the range between fairly tame first generation methods and the low end of wicked second generation methods. This is also the region of the space that was identified in the previous chapter as a niche for MODEs.

6.5 Three stages in parametric analysis which might benefit from knowledge aided support

Parametric analysis contains a certain amount of wickedness and complexity which conflict with the default objective of achieving a highly effective and efficient analysis. Later in this section it will be argued that this wickedness and complexity within the method cannot be

eliminated from the method. If this is the case then design researchers cannot tame the designers' work further by supplying vastly improved prescriptive design methods. A rational alternative would therefore be for knowledge aided design systems developers to supply programs that help designers to perform their work and where possible make second-generation knowledge available from within such programs. The purpose of the section, then, is to locate the wicked and complex stages within the method so that Patina can be aimed at supporting them.

The section begins by identifying the wicked stages of the method, examining their costs, and questioning whether further design research might ameliorate them. These tasks are then repeated for the complex stages of the method.

6.5.1 Three points of wickedness within parametric analysis

As Jones (1977) asserted, the points at which a prescriptive design method is least prescriptive, most value laden and most demanding of inventiveness—in other words most wicked—are where a designer must design a way to adapt the method to meet his or her perception of a design problem. Three such points, or black-boxes, can be found within parametric analysis¹. First, the task of inventing or selecting some criteria with which to describe and select a class of products is wicked. Second, the task of inventing or selecting parameters to measure these products is also wicked. Third, the task of inventing or selecting some criteria with which to search bi-variate scatter plots for patterns that might indicate design or marketing knowledge is also wicked.

These assertions can be supported by the following paraphrases of various entries in Rittel and Webber's list of defining features of a wicked problem (Rittel and Webber, 1973):

- The prescriptive design method does not contain a rule to prescribe when to cease making choices about products to include, parameters to define or parameters to compare (paraphrased from Rittel and Webber's second rule).
- On the same theme the prescriptive design method does not contain a rule to prescribe how to evaluate the choices that would bring these processes to a conclusion (paraphrased from Rittel and Webber's third rule).

¹The last stage of wickedness is an execution to Jones' rule because it is located at the final point in the procedural design method where designers invent explanations to describe and exploit these patterns.

- The choice of parameters and products can be evaluated at various stages of the parametric analysis (or even after the parametric analysis is deemed complete) by varying stake holders with varying results (paraphrased from Rittel and Webber's fourth rule).
- Each process of describing, selecting and evaluating choices of parameters and products will consume some portion of the available time to market and will thus contribute to the social and economic cost of developing a product (paraphrased from Rittel and Webber's fifth rule).
- There might be an alternative and unseen way to invent or select the criteria for choosing the parameters and products or for comparing parameters and naming relationships (paraphrased from Rittel and Webber's sixth rule).
- The choice of parameters and products that is made at the descriptive stage will determine the universe of data that is available for subsequent analysis (paraphrased from Rittel and Webber's ninth rule).

To summarise these paraphrased excerpts from Rittel and Webber's list, although the results of a particular parametric analysis are partially determined by the inventiveness and relevance of its own design the method does not specify how to adapt itself to meet individual circumstances that designers will face. For example, it does not prescribe which parameters to chose or which products to examine, it does not prescribe optimum sizes for the parameter and products sets, nor does it advise designers when to stop analysing a populated parameter set.

Having noted that the method is wicked in these three of its stages it becomes important to ask whether the wickedness is intrinsic or whether design methodologists could, in principle, find or invent domain specific ways to eliminate it.

Inventing a general purpose prescriptive design method to provide answers to each of the decisions that need to be made at these stages would be an unfeasible task. It would be unfeasible for reasons that are comparable in scope and philosophical naiveté to attempt to write a look-up table to do duty for general intelligence. This is because the circumstances which designers are liable to encounter, the content of their background knowledge and the values that they bring to design projects cannot be predicted or enumerated. Moreover, the languages of design which would be necessary storage media for such methods are likely to be value-laden and open to the same problems of

translation between specialities that were identified for design-secretaries in Chapter Four (see also Bond, 1992).

Inventing specific prescriptive design methods for specific types of products and value systems might be considered viable, in the same way that writing expert systems is considered viable for certain 'closed world' domains. However, in each case, some of the wickedness associated with a design task will merely be moved up-stream from a designer to a design researcher rather than eliminated. Moreover, such custom prescriptive design methods would probably suffer the same kind of 'brittleness', i.e., context sensitivity, as expert systems.

Both of the above arguments can be accused of being directed at 'straw people'. However they serve the purpose of showing that if the path of lowering the wickedness of parametric analysis is beyond the technical scope of the design methodology community then both domain-independent and domain-specific expert systems approaches are respectively flawed and limited. This leaves a more knowledge aided approach as the only currently viable technical approach toward dealing with this wickedness.

6.5.2 An area of complexity within parametric analysis

The complexity within parametric analysis is found within a glass-box surrounded by the previously mentioned black-box activity of choosing parameters to graph against one another. The term 'complexity' will be used in this context in a way that is derived from the computational subject area of algorithmics and which is applied to people. The subsection will hence begin by describing the concept of complexity in its native context of computer science and then translate its effects to the interaction of designers and parametric analysis.

In computer science an algorithm is described as computationally complex when the number of steps necessary to complete that algorithm increase at a rate that is greater than a polynomial of the size of its input (Harel, 1987). The complexity of parametric analysis is chiefly found in the analytical stage which demands a lot of inventive and context sensitive search and analysis. So designers who follow the method must expect to incur the costs of a lot of repetitive and time-to-market consuming analyses.

The instructions for parametric analysis prescribe that designers should analyse a bi-variate plot for each parameter pair in a parameter set. This prescription requires a search space of at least *size of parameter set (size of parameter set - 1) / 2* nodes. However, since the procedural design method is both iterative and wicked, the complexity may be far worse than this in practice.

So, if the analytical stage of parametric analysis could be reduced to an algorithm, a complexity theorist should not consider that algorithm to be too complex for a computer to perform because the algorithm requires a polynomial number of steps. However, it will be remembered that this stage of parametric analysis cannot be reduced to a predictable mechanical algorithm because analysing and interpreting each of the plots in this search space requires human intuition. In other words the search space should be regarded as the structural component of a prescriptive design method and each node in this space should be regarded as a black box process which requires valuable human design skills. So the prescribed method for the analytical stage of parametric analysis can be considered to be complex in terms of human resources because even a modestly sized parameter set of, say, fifty parameters will require designers perform one thousand two hundred and twenty five analyses. As the fifth paraphrase from Rittel and Webber's list of aspects of a wicked design process reminds us: each process of describing, selecting and evaluating choices of parameters and products will consume some portion of the available time to market and will thus contribute to the social and economic cost of developing a product.

Hence the demands of this activity conflict with the inherent demands of performing an efficient analysis which would, in ideal circumstances, only compare viable pairings of parameters.

The problem with trying to eliminate the complexity from the prescriptive design method for parametric analysis is more straightforward than that of trying to eliminate its wickedness. The instructions for parametric analysis warn that "it is essential that the participants are willing and able to do the many cross-plots necessary, without in the first instance being able to deduce the reason for so doing...[indeed]...the reason for doing a particular cross-plot may appear to be illogical...this is not to say that traditional, logical relationships should not be plotted" (Pugh, 1991, p 36-37). In other words, since the point of analysing the relationships between pairs of variables is to uncover unsuspected patterns between any and all parameter pairs design researchers can offer no *a priori* and generally applicable method with which to prune the search space. So it is worth reiterating Hollins' advice that common sense *has* to be applied here (Dr. William Hollins interviewed by the author in 1995).

Given the power of computers to display graphs quickly and accurately this leaves knowledge aided design research as a possible means to deliver designers a tool with which to engage in this glass-box (though extremely complex) task. The goal, as Fischer has suggested, would be to "divide the exponent of such a problem" (1992, p 207).

6.6 Summary

The introduction to the chapter described the triggers which led to the choice of parametric analysis as a source for a MODE for parametric analysis. The core of the chapter presented descriptions of the organisational contexts in which parametric analysis is likely to be used, the purpose of performing a parametric analysis, together with the kinds of knowledge which it has been shown to produce and the types of products and services it can survey. In preparation for designing a MODE to support parametric analysis the various stages of the method were described in detail alongside the artefacts and other materials which are used during an analysis. Finally parametric analysis was located along in the design activity space, and the stages in the method which might benefit from knowledge aided support were highlighted.

In Chapter Seven the implementation of Patina ~ a MODE for parametric analysis will be reported. This feasibility prototype of a MODE will be designed in accordance with the mappings and theory presented in part one of the thesis to support the wicked and complex elements of parametric analysis which were highlighted in this chapter.

Chapter Seven

The development of Patina: a MODE for parametric analysis

7.0 Overview

The work reported in this chapter completes the process of addressing the fourth hypothesis:

- H4** It is possible to design and implement an example of a MODE by locating a design practice in the design activity space and then applying the mapping to Green's model space in order to propose an aggregate of components that will support this practice.

The chapter describes how the mapping between the design activity space and Green's model space guided the development of Patina ~ a MODE for parametric analysis.

The chapter contains five main sections. The first section motivates the choice of Connell and Shafer's object oriented rapid prototyping method (1995) to implement a high fidelity feasibility prototype of a MODE. The second section describes the program's concept design with guidance from the mapping between the design activity space and Green's model space. The third section describes the implementation of a detail design in the Small Talk programming language. It also describes the 'rug plot' a novel interface component which was adapted from the data graphics literature (Tufte, 1983). The fourth section describes the program's performance in the context of an informal trial with a group of target users. The fifth section discusses what can be inferred about the viability of implementing MODEs according to the theories presented in previous chapters on the basis of developing Patina.

7.1 Development method

Prototypes are an important tool for software developers because they "often reveal that what we wish for is unrealistic or ill conceived. Conversely, prototypes can reveal that the designer's wishes were not sufficiently imaginative" (Schrage, 1996, p 194). Patina was developed as a high fidelity feasibility prototype of a MODE using Connell and Shafer's object oriented rapid prototyping method (Connell and

Shafer, 1995). In this section the definitions of 'high fidelity' and 'feasibility' prototypes will be discussed, Connell and Shafer's method will be described and the decision to implement Patina in this fashion will be motivated.

7.1.1 Reasons for implementing a high fidelity feasibility prototype

The fidelity of a prototype may be low or high and each of these approaches has its own strengths and weaknesses. Low fidelity prototypes range in complexity from paper based illustrations of a system to animate story boards, implemented in an environment such as Macromind Director, which illustrate the functionality of a user interface. The main advantage of low fidelity prototyping is swift development time which allows for multiple iterations of the development cycle. However this must be weighed against the disadvantage that domain specialists who are not well versed in human computer interaction can find it difficult to envisage a story board as a working system. High fidelity prototypes range in complexity from basic systems written in a scripting language, such as Hypercard, to more realistic systems written in a fourth generation language, such as Small Talk. The advantage of high fidelity prototyping is an increased level of realism in the resulting product and the disadvantage is a vastly extended development time which means less approaches can be tested. However, the improved interactivity and response found in a high fidelity prototype can make it easier for potential users to understand the program design's strength and weaknesses.

A feasibility prototype is a system that is used to demonstrate that something is possible and to discover whether it will meet the needs of an organisation or profession (Goldberg and Rubin, 1995). The opposite of a feasibility prototype is an analysis prototype, which is a penultimate stage in system development that is implemented after several iterations of feasibility prototyping have proven the worth of a concept.

As Patina is being developed to demonstrate the potential of MODEs to designers, design researchers and developers of systems for knowledge aided design, it was reasoned that a high fidelity feasibility prototype would enable the whole target audience to envisage the utility of an example of a MODE and, by induction, the potential utility of this class of system.

7.1.2 Reasons for following an object oriented methodology

Once the decision had been made to implement a high fidelity feasibility prototype it was prudent to choose a well tested prototyping methodology. Connell and Shafer's object oriented rapid prototyping method can be used to develop high fidelity feasibility prototypes.

One major advantage of performing object oriented prototyping is that a successful prototype can produce classes that can be reused when a production version of the program is implemented in an efficient object oriented language. This potential advantage is lost when implementing a prototype in an application such as Macromind Director. This is because such applications do not provide access to the underlying data-representation, hence the desirability of coding in a comprehensive object oriented development environment. In summary, this method affords an opportunity to build the first stage of a 'robust design' that is "created with the intention or possibility of future evolution" (British Standards Institute, 1995).

The task of building a high fidelity feasibility prototype using Connell and Shafer's method is addressed in three stages:

- (1) some initial requirements are gathered by interviewing potential users of the program and experts in the task domain and then reviewing literature from the subject area
- (2) a prototype is implemented in what Connell and Shafer call a 'comprehensive object oriented development environment'
- (3) the prototype is presented to a panel of experts and users in the context of a structured scenario to elicit their opinions about the software's design and utility.

Connell and Shafer present three prescriptions that capture the essence of object oriented rapid prototyping: "Build something that really works and contains actual user-familiar data, instead of just a pretty screen prototype. Build it with tools that allow for performance tuning ... so that it is not just a requirements model and you will not have to start over again. Pre-specify only a tiny portion of the requirements for which you are sure there is complete understanding and agreement between you and the users." (Connell and Shafer, 1995, p 19). The third of these prescriptions seemed very important to Patina as the genre of software being developed was new

and the process being supported was likely to be unfamiliar to some of its target users.

The feasibility prototype whose design is reported here was based upon a literature review of parametric analysis, an interview with Dr William Hollins, the theory developed and cited in the first part of this thesis, an implementation in Small Talk, and an initial informal evaluation with a team of designers and students of design¹.

The work presented in this chapter therefore represents one iteration of Connell and Shafer's methodology.

7.2 Concept design

The concept design phase reported in this section will describe the high level knowledge aided components that address the opportunities for knowledge aided support for parametric analysis which were identified in the previous chapter. The choice of these components was guided by the mapping between the design activity space and the various tools from Green's tool set which was presented in Chapter Four.

7.2.1 Suitability of parametric analysis for implementation as a MODE

In the previous chapter it was argued that parametric analysis seems suited to design projects that are located at: the analytical stage of design, the range between variant-design and the beginning of innovative-design, and the range between tame design activities and the low-end of wickedness. It was therefore suggested that parametric analysis is located in a particular volume of the space of design activity space which matches with the niche for MODEs. So the theory presented in the previous chapter indicated that a MODE would be an appropriate style of program to develop for parametric analysis.

7.2.2 Reasons to base the structure of Patina's interface on the structure of parametric analysis

One of the defining features of a MODE is that its interface should represent the method being followed rather than the object being designed. It therefore seemed

¹This evaluation is only summarised here as it is concerned with evaluating Patina as a program rather than in its role as a MODE.

appropriate to base the structure of Patina's command interface on the structure of the method for parametric analysis. This meant that the system's command structure and layout of windows could be viewed as a kind of implicit design-informer which might tend to lead the program's users through the method's constituent stages in a logical order.

One consequence of this design decision was that the following set of windows were considered necessary: a window to represent the product set, a window to represent the parameter set, a window in which parametric paintings could be viewed and, for reasons that will be clarified below, a window in which a PDS document could be constructed. This PDS window was added to the specification during the implementation of the first prototype of Patina. Two other windows were also added: an introductory screen and a 'home page'. The reasons for the inclusion of these windows will be discussed below.

At this point in the program's conceptual design a broad framework for the program was in place and the focus of the decisions became smaller because individual tools needed to be matched to the tasks of parametric analysis. This meant that the mappings between the design activity space and Green's model space became relevant.

7.2.3 Reasons to add a passive design-informer in the format of a flow chart as a main interface metaphor with embedded design-informers and secretaries

The first action when designing Patina's central metaphors was to seek mappings between the design activity space and Green's model space for the stage of analysis, true variant-design and tame design activities. Then the mappings were adapted to meet the specific needs of the problem at hand.

The mappings that applied to the stage of analysis were presented in section 4.3.2. These are relevant to the overarching design of Patina because analysis is the stage in design that Patina supports most closely (although synthesis and evaluation play roles within a parametric analysis). The suggestion that deep and wide knowledge should be made available indicated that passive design-informers would be appropriate throughout the program. This is because passive design-informers can present wide knowledge in a visual format to orient readers, and they can present more specific knowledge in the form of browsable text that is embedded in larger diagrams.

The mappings that applied to true-variant design, which were presented in section 4.3.5, also suggested that design-informers would be appropriate. However, these

mappings also suggested that design-secretaries would be appropriate, particularly if more than one designer was involved in an analysis and any form of user seeding were being performed. For this reason 'simple' design-secretaries were made available throughout the program.

The mappings that applied to tame design strategies, which were presented in section 4.3.8, recommend using design-experts when the task is domain-dependent and when opportunities for design rational capture are available. Since parametric analysis is domain independent it seemed prudent to begin by implementing design-secretaries so that design strategies and rational might be captured for use in design-experts and design-strategists in future versions of the program.

Since Patina's target user group include students and professional designers who might be inexperienced in using parametric analysis, it was reasoned that the way in which the program's windows served as an implicit design-informer might need to be reinforced by placing more explicit maps of method within the windows themselves. Such maps would illustrate the logical ordering of these stages and be annotated with text that would indicate the logic behind these orderings. The main inspiration for using a flow chart for this purpose was the proven viability of the interactive flow chart from the interface to Gilleard and Lee's CITIS system (Gilleard and Lee, 1996).

A passive design-informer appeared to be the simplest tool available for such tasks. It was therefore decided that a passive-informer in the format of a flow chart of the method should be included as a repeating motif throughout the program. It was also decided that this graphical representation of the method could be supported by quoted text from the original description of the method. At this stage in the development cycle it seemed prudent to refrain from implementing the map as a hyper linked object unless, and until, users of the program expressed a wish for such a feature.

It was presumed that most of Patina's potential users would either be engaged in higher education or would have already earned higher national diplomas or degrees. It was therefore decided to adopt academic conventions in the way that text was presented and references were cited. It was thought that an introductory screen which made the program's influences clear via citation would address this issue.

One consequence of these design decisions was that an introductory screen would be needed to make the program's influences clear. It was decided to introduce the concept of the passive-informer at this point. It was reasoned that the knowledge presented by the design-informer should be broad as it introduced the method and also narrow as it gave feedback on the progress to date of each analysis.

A second consequence of these design decisions was that a window to house a passive-informer, with sufficient room for in-depth annotation, was considered necessary. A 'home page' was therefore added to the set of windows to be included in the program.

In later screens it was reasoned that design-secretaries and design-informers should be located within the overarching metaphor of a design informer based on a flow chart.

7.2.4 Reasons to address the wickedness of designing a parameter set by including an interactive design-informer

The previous chapter argued that parametric analysis contains three especially wicked elements. The first is the task of inventing or selecting some criteria with which to describe and select a class of products. The second is the task of inventing or selecting methods by which to measure and represent values for the various parameters of these products. The third is the task of inventing or selecting some criteria with which to search bi-variate scatter plots for patterns that might indicate design or marketing knowledge.

The problem here is that wickedness represents a research agenda for knowledge aided design. So the mappings based on reasoning about prior research were unlikely to offer dependable guidelines. What they did suggest was adapting design-informers, design-documenters and design-simulators to the task. The design-informer was selected and it was augmented with a design-visualiser (see section 7.3.10). For these reasons this aspect of the design of Patina can be seen as a development of the guidelines in the mapping presented in Chapter Four.

It was decided that the wickedness found in the task of designing a parameter set would make a suitable candidate for support that could be adapted to the other wicked elements of the method. This task was chosen because it was considered at least equivalent in wickedness to that of designing a product set and it was analogous to planning a set of analyses of the parameter paintings.

It was therefore decided to concentrate initially on developing a design-informer for the parameter set window. Concentrating on this stage of the method would mean that some of its positive results could be transferred to the design of the windows for the product set and the parametric paintings during subsequent iterations of the program's design.

In design activities which range from routine design to creative design, the tool from Green's space of tools which seemed best-suited to addressing this type of

wicked task appeared to be an interactive design-informer. Such a design-informer could invite designers to consider important aspects of a product that should be considered as a part of the parameter set. It could also invite users to browse design knowledge about these aspects in the form of text sourced from the original design method. The knowledge that seemed to be required related to the standard set of aspects of a product that one would expect to find in a PDS document (see table 6.1).

One consequence of this design decision was that the window to house the parameter set would need ample space for an interactive design-informer.

7.2.5 Reasons to address the wickedness and complexity of analysing a parameter set by including a rug plot

This third of the element of wickedness discussed above also contributes to the complexity of parametric analysis. This complexity is chiefly found in the analytical stage of the method which demands a lot of inventive and context sensitive search and analysis of scatter plots.

The arguments about wickedness and the mapping between the design activity space and Green's model space presented in section 7.2.4 also applied to supporting this part of the method. However, since this part of the method clearly contained a black box element surrounded by a glass box element it seemed appropriate to use a domain independent design-visualiser. So the arguments developed in Chapter Three and 4.1.4 were germane to the concept design of this area of the program.

It was hypothesised that the complexity found in the task of analysing parameters might be reduced if users could investigate groups of related parameters. This insight was partially inspired by the instruction within the method for parametric analysis that: "Grouping of plots, incorporating the same parameters, is always useful and revealing" (Pugh, 1991, p 37).

The type of tool from the newly enhanced version of Green's space which seemed best-suited to addressing this type of complex task appeared to be a design-visualiser. However, a suitable metaphor needed to be devised at the concept design stage before detail design could begin.

There were two reasons to infer that a format for data display known as the rug plot (Tufte, 1983) would be a suitable metaphor upon which to base the graphical user interface for such a design-visualiser. First, a rug plot displays statistical data and cues in a very sparse format. It uses a minimal amount of what Tufte calls 'data ink', i.e., "the non-erasable core of a graphic, the non-redundant ink arranged in response to variation in the numbers represented [in a graph]" (Tufte, 1983, p 93).

Using a minimal amount of data ink allows the use of a minimal amount of pixels. This is an important consideration because users will need to display a lot of data upon the confined space of a computer monitor. Second, a rug plot can be used to project an n-dimensional space onto a flat surface. This is achieved by 'knitting' bi-variate scatter plots into a cohesive matrix, or 'rug'. This seems beneficial because it allows designers to analyse groups of parameters on screen without recourse to data compression.

In principle a rug plot could be used to examine any number of parameters. However, there are two factors which are likely to limit the number of parameters which designers can realistically examine. The first of these is the limited size of computer monitors. This factor could be addressed by adding extra artefacts, such as scroll bars, to the user interface. However, engineering psychologists, such as Wickens (1992), recommend against such practice because when possible all the data being analysed at one time should be visible. The second factor is that engineering psychologists, such as Wickens (1992), prescribe that only seven (plus minus two) 'chunks' of data should be presented at one time—this prescription is based on Miller's evidence that human short term memory can only hold seven-plus-or-minus-two meaningful chunks of data at one time (Miller, 1956).

During the interview which was conducted as part of the requirements analysis for this program (Dr. William Hollins interviewed by the author in 1995), Dr. Hollins made it clear that the ability to draw lines on scatter plots was very important. In fact, Dr Hollins went further and asked whether the program could generate the lines *for* the users. At this point it was considered prudent to follow Connell and Shafer's previously cited advice about only concentrating on well-understood functionality in the first prototype of a system (Connell and Shafer, 1995). It was therefore decided to implement the rug plot itself and to subsequently gather users' requests for tools that would discover trends during the initial evaluation.

One consequence of this design decision was that the window to house the parametric paintings needed to be large enough to contain a matrix of nine bi-variate scatter plots (presenting up to six parameters). A second was that each of these plots should contain a tool for drawing lines and indicating clusters. A third was that the underlying data representations should have interfaces that could accommodate sensible requests from an array of artificially intelligent tools. The window which housed these multiple parametric paintings was called the parametric easel. A final consequence was the decision to add a miniature version of this tool to the parameter set.

7.2.6 Reasons to make design-secretaries and design-documenters available in the product set, the parameter set and the parametric paintings

In Chapter Four a design-documenter was described as a tool that would be embedded in a larger system where it would prompt designers to record the reasoning behind their design decisions. It was also explained that Green advocated embedding such design-documenters in graphic representations of artefacts and systems so that designers could investigate how a particular item came to achieve a current state. It was hypothesised that Patina could benefit from the inclusion of a design-documenter in which users could record design rationale about the product set, the parameter set and the relations between the parameters. This appeared to be beneficial to the sponsoring organisation as it would mean that design rationale could be captured for subsequent seeding of the program and to contribute to the PDS document, which is a natural venue for the output of a parametric analysis.

The provision of design-secretaries and design-documenters available is supported by the mappings presented in Chapter Four for the reasons given in section 7.2.3. One consequence of this design decision was that the design team appeared to require a part of the program where they could describe themselves so that authorship of design rationale could be attributed.

7.2.7 Reasons to include a representation of the PDS document

The previously discussed need for a design-documenter rested on the perception that the PDS document is a natural venue for the output of a parametric analysis. It was therefore hypothesised that performing an isolated parametric analysis would not be very productive and that it would not provide users with a sense of closure. It was therefore decided that Patina should venture beyond the confines of the method for parametric analysis and include the PDS document, which is an artefact of the next stage in the total design method. This meant that the flow charts that would serve as passive design-informers throughout the program should make clear the place of parametric analysis in the larger method. One consequence of this design decision was that the program would include a window that represented an interactive PDS document.

7.2.8 A summary of the conceptual design decisions

To summarise this subsection, Patina's key concepts include: a command structure and window layouts that map onto the stages and artefacts in parametric analysis, flow charts which make the method's stages and connections explicit, guidelines from the original instructions for the method, an interactive design-informer to help users to design a parameter set, a design-visualiser to help them to interact with groups of parametric paintings, a design-documenter to capture design rationale, and a PDS document to provide an outlet for results and a sense of closure for the users.

Each of the above concepts was subsequently designed in detail. The output of this detail design, which was concurrently implemented in an object oriented programming environment, is illustrated in the following section.

7.3 Detail design

The previous section documented Patina's conceptual design decisions. This section turns to Patina's detail design. It will begin by explaining the choice of programming language and environment. It will then describe the program's graphical user interface (GUI) by illustrating the various commands and windows in a depth-first fashion using screen shots from a run of the program.

7.3.1 The choice of programming language and development environment

The prototype of Patina was implemented in the Small Talk programming language using the Visual Works 2.0 development environment on a Power Macintosh 8100 with 24 MB of RAM running under System 7.5. This combination of Small Talk and Visual Works 2.0 was chosen for four reasons.

First, Patina was developed via the object oriented rapid prototyping method and the Visual Works implementation of Small Talk contains sufficient generic classes and GUI development tools to qualify as a 'comprehensive object oriented development environment' (Connell and Shafer, 1995). This type of development environment is recommended by Connell and Shafer as a prerequisite tool for object oriented rapid prototyping.

Second, since a Small Talk application has access to its development environment, end-users could in principle customise Patina to meet these unforeseen needs and ways of working. This point has implications for other paradigms which

might come into play during subsequent iterations of the program's design, such as evolutionary re-seeding by professional knowledge engineers or designers.

Third, a range of third party vendors and academic institutions sell additional components which implement expert systems from within the Visual Works environment. This means that many of Green's expert system based tools could be added to the program if they appear to be needed on the basis of outcome of the program's evaluation. This provides a second potential upgrade path for the program.

Finally, Visual Works is a cross-platform development environment and the most recent implementations of the environment include the facility for web-based programming and delivery. So Patina could be adapted such that non co-located designers could use it across a variety of operating systems and the world wide web. This provides a third potential upgrade path.

7.3.2 The graphical user interface

The following subsections will describe the detail design of Patina's GUI in a depth-first fashion. The interface is platform independent and it incorporates the concept of a home page from web-based programs.

7.3.3 The detail design of the introductory screen

When Patina is invoked from the Visual Works command window it presents its users with an introductory screen in the form of a modal dialog window. This screen describes the program and sets it into its expected context of use. The screen contains two items. The first is a brief textual description of the program's purpose (together with some references to useful literature about the method). The second is a flow chart of the total design method.

The text is intended to inform users of the method's origin in order to reassure designers that the program is inspired by the prescriptions of design methodologists rather than computer scientists. It also functions as an acknowledgement and formal reference to the works of Hollins and Pugh, which are cited throughout the program.

The flow chart is intended as a navigational aid and a motif that is repeated and expanded throughout the program. Adopting and adapting Green's terminology this chart can be described as a passive design-informer, whose purpose is to make the program's expected context of use explicit and to guide designers into the method they are performing.

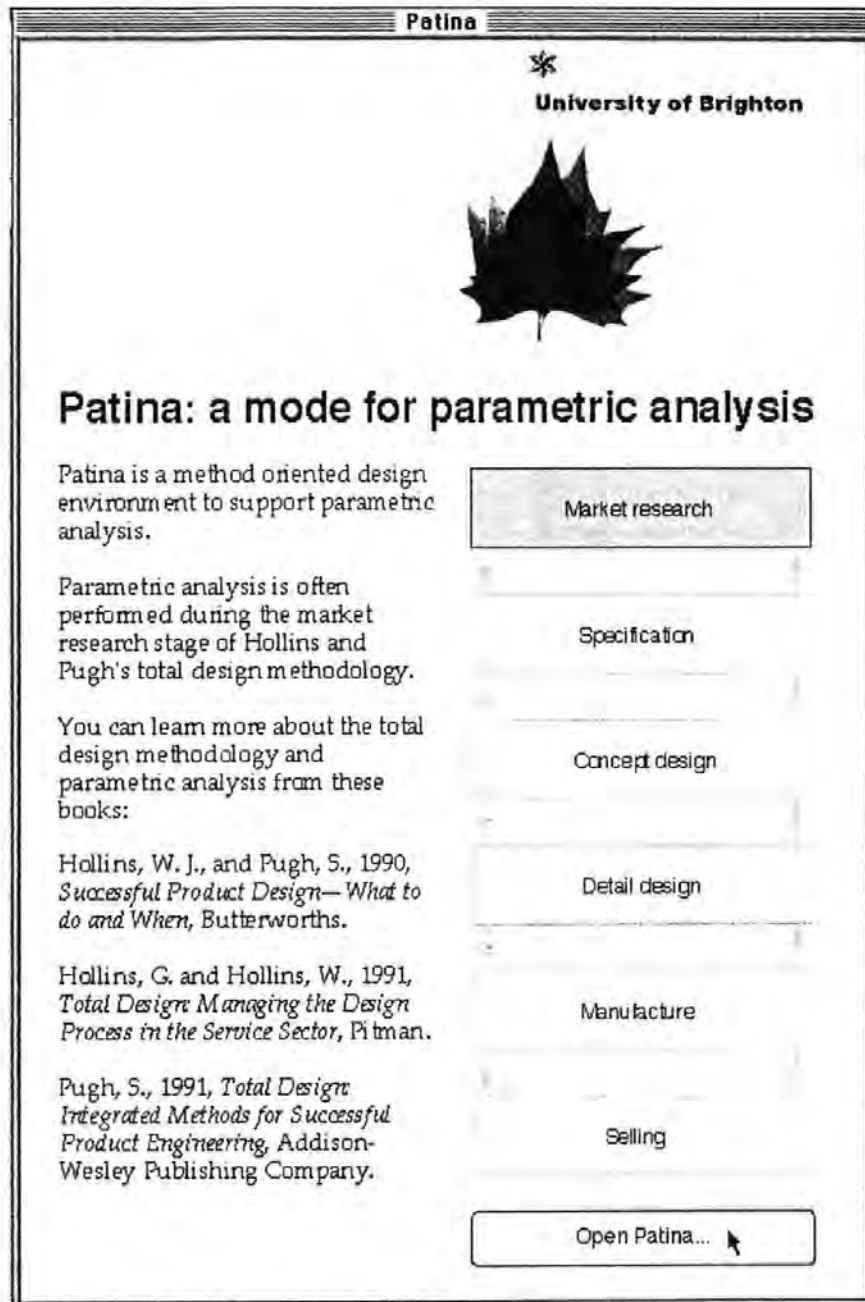


Figure 7.1: Patina's introductory screen.

The first stage of the chart represents the point in the total design method where parametric analysis is performed. It is therefore highlighted in a fashion that is repeated for various stages of similar charts throughout the program.

Potential upgrade paths for this chart include linking its sections as buttons to windows or textual-panes which describe corresponding parts of the total design method in detail. At this point in the program's development, however, it would not appear sensible to recast the highlighted stage as a hyper link to subsequent parts of

the program. This is because Patina does not support the whole process of market research. An alternative would be to create a hyper link to a version of the diagram of the constituent processes of market research on page 32 of Pugh's *Total Design* (1991).

Once Patina's users have read this introductory screen, clicking on the button labelled 'Open Patina' leads them to a modeless window representing a 'home page' with Patina's top-level menu-bar.

7.3.4 The detail design of the combined home page and menu bar

This modeless window contains four main elements. These will be described by working down from the upper-most part of the screen to the lower-most part. The first element is a menu bar which is local to the window and which provides access to the other windows in Patina's GUI. The commands on these menus enable Patina's end-users to: control file management; build the content of a knowledge base containing the sets which will contain the information for the analysis; configure and examine parametric paintings, and to view and update the content of a PDS document based upon the results of the analysis. The second element is a text pane which describes the purpose of parametric analysis via a substantial quote from Hollins and Pugh (1990). The third element is a chart of parametric analysis which contains instructions for performing the method. This chart emerges from a version of the chart of the total design method which Patina's users will already have encountered in the introductory screen. It serves as part of the passive design-informer that is distributed across the program. The fourth element is a pair of labels giving feedback on the current size of the product and parameter sets.

This window functions as part of an implicit of design informer because the structure of the menus follows the intended flow of stages in the method, i.e., assemble a team, design product and parameter sets, perform the analysis and record the results in the PDS document, and the chart presents a visual representation of this order by emerging from the marketing section of total design and arriving at its specification section.

Once Patina's users have had the opportunity to read the instructions in this window they can begin to explore the menus on the menu bar. These will now be described, working top-down from left to right.

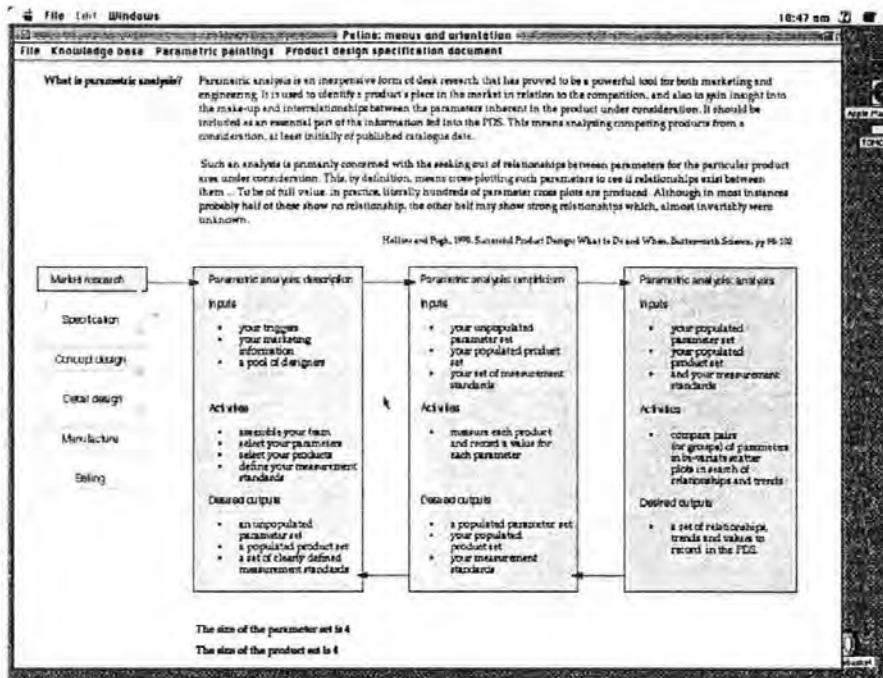


Figure 7.2: Patina's home page and menu bar.

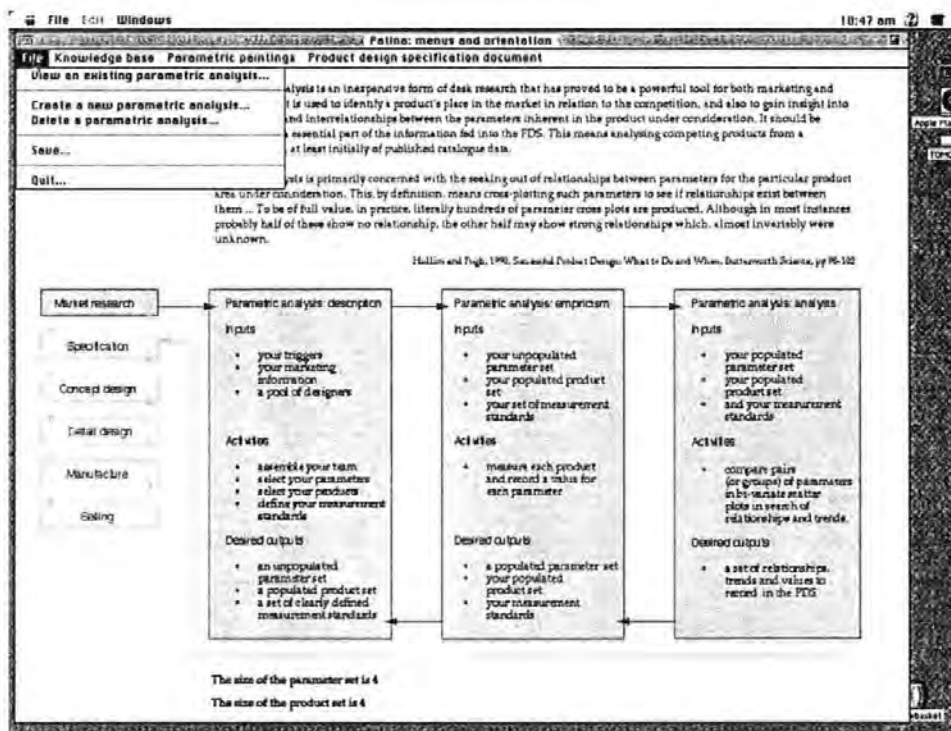


Figure 7.3: Patina's file menu.

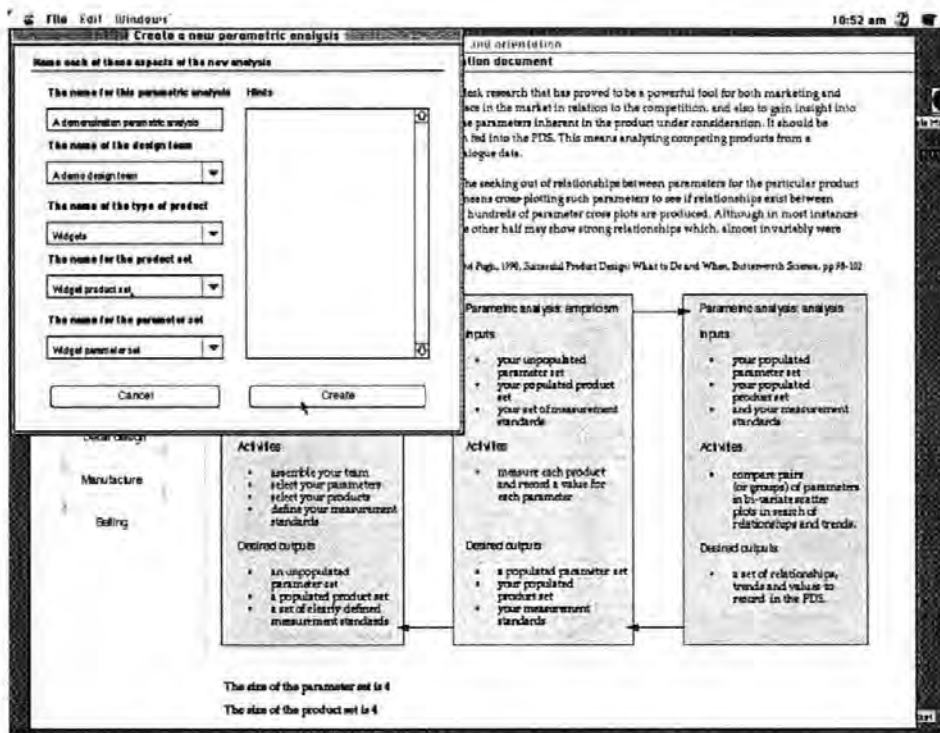


Figure 7.4: Patina creates a parametric analysis dialog window.

7.3.5 The detail design of the dialog window for creating a parametric analysis

The commands on Patina's file menu allow Patina's users to perform standard file opening, creation saving and destruction actions, i.e., view an existing parametric analysis, create a new parametric analysis, delete a parametric analysis, save and quit (figure 7.3). Each of these commands has an associated modal dialog window which offers some useful feedback.

7.3.6 The detail design of the dialog window for creating a parametric analysis

The most important of the windows engendered by the file menu is the modal dialog window for creating a parametric analysis (figure 7.4). This prompts Patina's users for the names of: the analysis itself, the design team, the type of product, the product set and the parameter set. Feedback is provided about names of existing sets and pull-down menus are provided for re-usable names. For example, a parameter set might be re-used for a new analysis, in which case its existing attributes would be cloned for the new analysis. One consistent feature of the interface which is introduced here is that text areas which either collect or present knowledge to and

from users are coloured yellow. Creating or opening an analysis causes the view held on the knowledge base menu (see below) to be opened. Incidentally, creating a parametric analysis causes a 'set' representing a new analysis to be created in Patina's model layer. This set contains other sets which will represent the design team, the product set, the parameter set and the PDS document.

7.3.7 The detail design of the commands on the knowledge base menu

The commands on Patina's knowledge base menu (figure 7.5) allow Patina's users to view windows which show them the current state of the design team, the product set and the parameter set. These windows have their own menu bars, which create associated dialog windows, i.e., 'children'. In these dialog windows users can create, update and delete representations of designers, products and parameters. Each of these windows and sets of children will now be described individually.

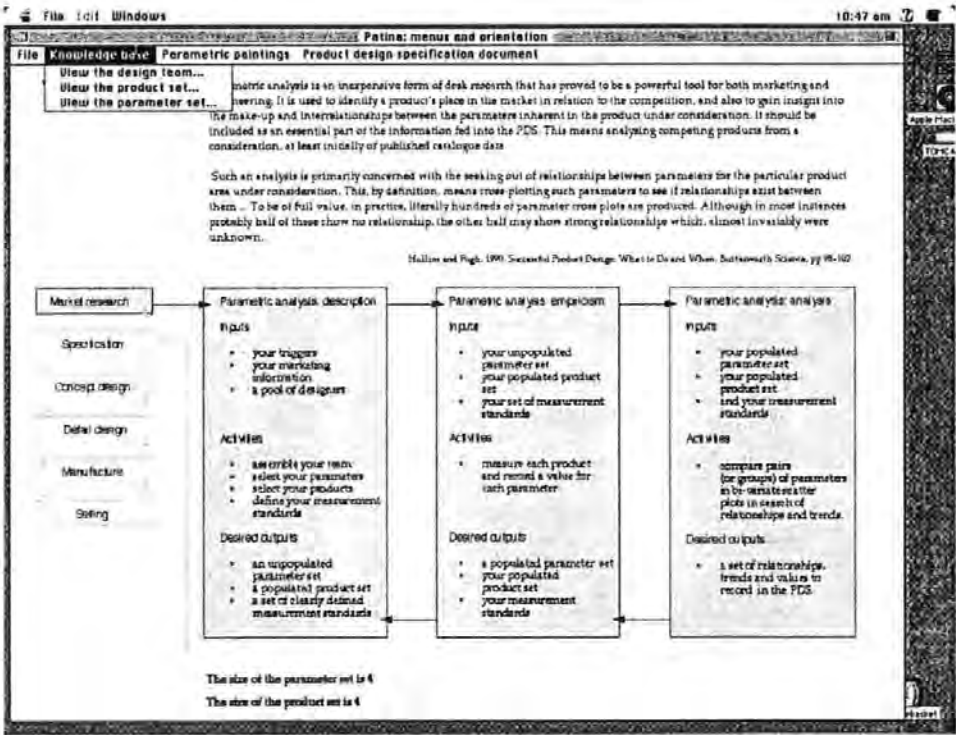


Figure 7.5: Patina's 'knowledge base' menu.

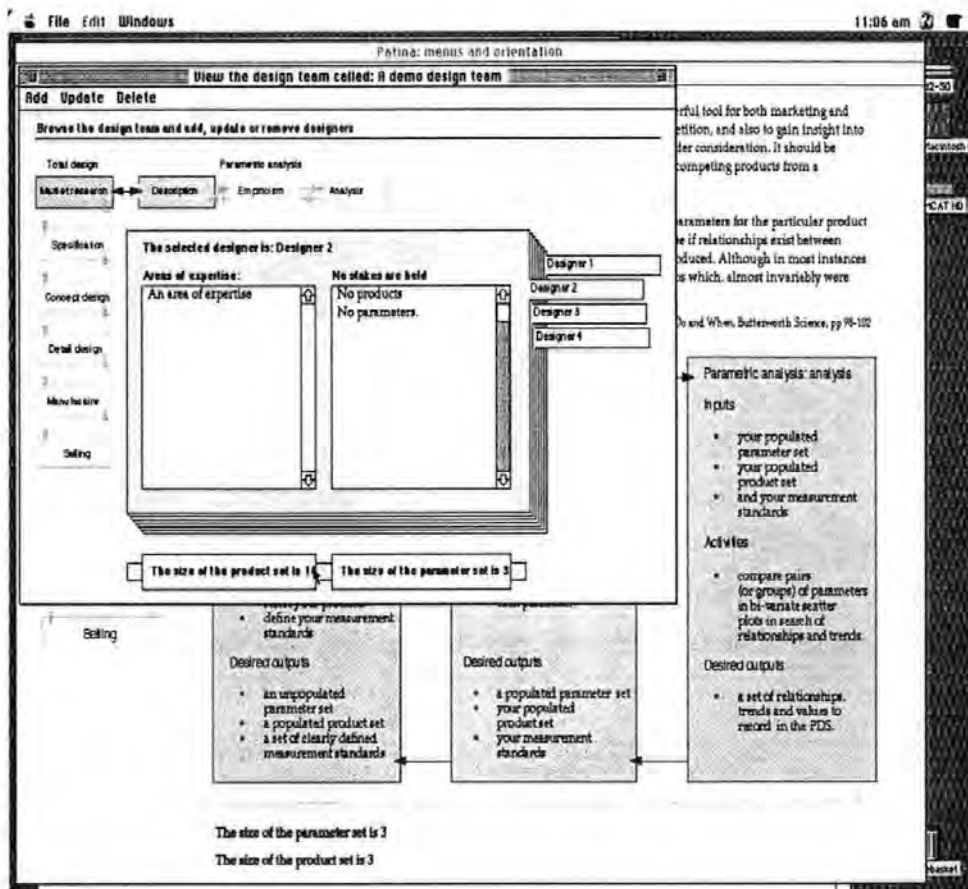


Figure 7.6: Patina's design team window.

7.3.8 The detail design of the design team window and its children

The design team window serves three purposes. First, to help the design team to become aware of one another's abilities (particularly if they are not co-located). Second, to help designers working at the later stages of design to locate the authors of design rationale that has been generated throughout a project and recorded in the PDS document. Third, to help knowledge engineers or designers to re-seed their design environments using knowledge in the form of design rationale that can be attributed to a designer or group of designers working in a design team.

The knowledge referred to in the second and third of these points is written by designers during the analysis. It may refer to either specific products or parameters. Once a designer has added some design rationale to a product or parameter of that design, they become a 'stake holder' in that particular product or parameter (Rittel and Webber, 1973).

To these ends, the window embeds its functions in a chart showing that the designers are engaged at the 'description' stage of parametric analysis and allows

designers to add, update and delete descriptions of themselves. These descriptions are presented in a note book whose format is repeated throughout the rest of the program (figure 7.6). In this note book individual designer profiles can be accessed by clicking on the relevant named tab. Each profile shows: the designer's name, their areas of expertise and any 'stakes' that they hold in particular products or parameters. The lower-most element of the window shows more of the context that the designers are working in by displaying the sizes of the product and parameter sets.

Designers can be added to the note book via a command on the menu bar of this window which causes a child dialog window labelled 'add a designer profile' to appear (figure 7.7).

This modal dialog window allows the designers to name a new designer and to add brief descriptions of their areas of expertise. Appropriate feedback is provided about the names of existing designers and the areas of expertise for this particular designer. Various tests are performed to ensure that duplicate designer profiles are not created (similar tests are performed in the windows for adding and updating products and profiles). Once the designer's profile has been created the designer's name can be selected as an author of items of design rationale throughout the rest of the program.

Descriptions of designers can be updated via another command on the menu of the design team window. This causes a modal dialog window to appear in which a designer's name can be changed, new areas of expertise can be added and existing areas can be deleted (figure 7.8).

A final command allows designer profiles to be deleted via a third modeless dialog window (for brevity the dialog windows associated with deleting designer profiles, product profiles and parameter profiles are not illustrated). Once a designer's profile has been deleted the entry for that designer is removed from the note book in the design team window.

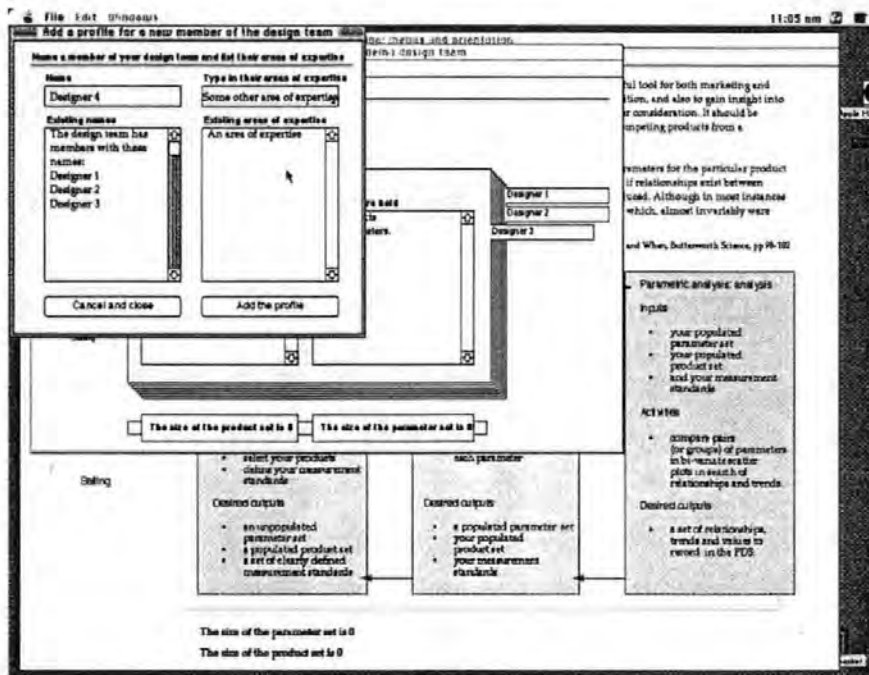


Figure 7.7: Patina's add a designer profile dialog window.

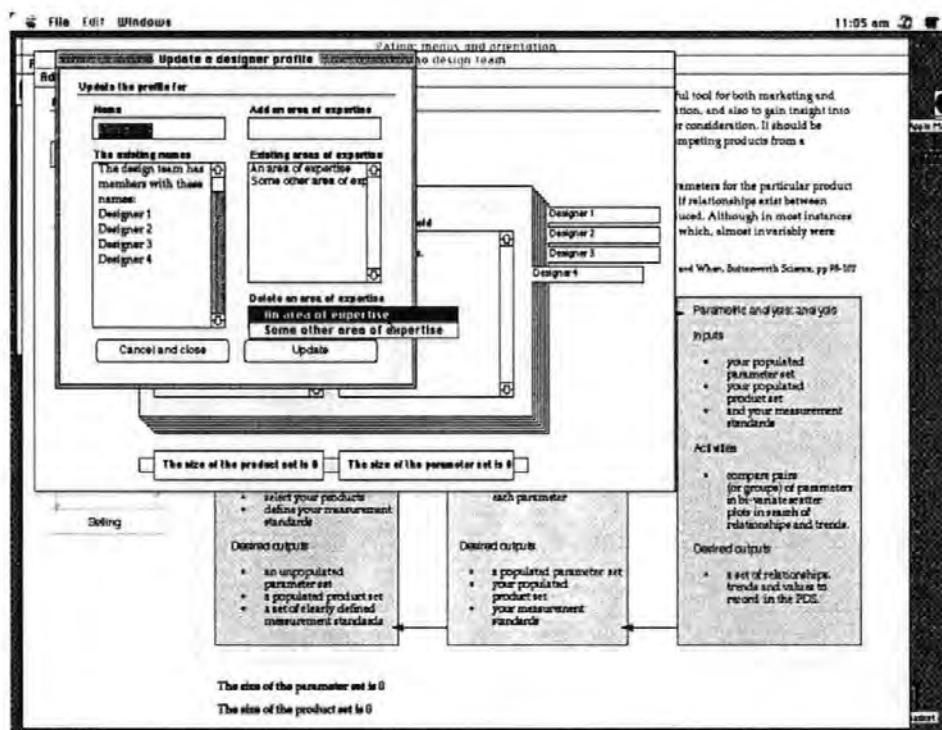


Figure 7.8: Patina's delete a designer profile dialog window.

7.3.9 The detail design of the product set window and its children

The product set window is a modeless window that allows the designers to build a set of products which they will explore via the parametric analysis.

The window's structure is similar to that of the design team window: it features a central note book and a menu bar that allows designers to add, update and delete products and which is surrounded by a chart showing its place in the design method. The additional command on this menu is one that allows designers to capture a picture of each product (figure 7.9).

The note book represents the product set with pages that each contain: the values associated with the various parameters for the product, the parameters that lack a value, the stake holders, a large coloured area which shows the hue which will be used to graph the product¹, and a button that opens a window containing a list of design rationale associated with the product. Selecting some design rationale from this list allows the designers to read or modify that information using the program's design-documenter.

The command for adding a product, which is housed in the window's menu, leads to a modal dialog window (figure 7.9). This window contains fields and pull-down menus that represent: the products name, its type (this knowledge is inherited from dialog window for creating a parametric analysis), its manufacturer (this can be typed in or selected from manufacturers that have already been entered) and whether the manufacturer is associated with the design team (this information becomes important at the stage of completing the PDS document).

The command for updating a product leads to another modal dialog window with similar fields to those mentioned above (figure 7.10). The additional feature of this window is that it contains a set of buttons that allow the design team to add, update and delete design rationale about the product. Such design rationale is entered in the form of optional notes (as illustrated in the right-most portion of figure 7.10).

An optional note contains text fields for typing in the subject and content of the note. It also contains a list of members of the design team in which multiple authors can be selected. As mentioned above, once a designer becomes an author of such a note they also become a stake holder in the product. The optional notes are recorded in windows created by the design-documenter whose three button interface is repeated in the windows associated with the parameter set and the parametric easel.

¹One of the important prescriptions in the definition of parametric analysis is that colour should be used where possible to distinguish between the products on the scatter plots (Hollins and Pugh, 1990; Pugh 1991). See also the description of the parametric easel below.

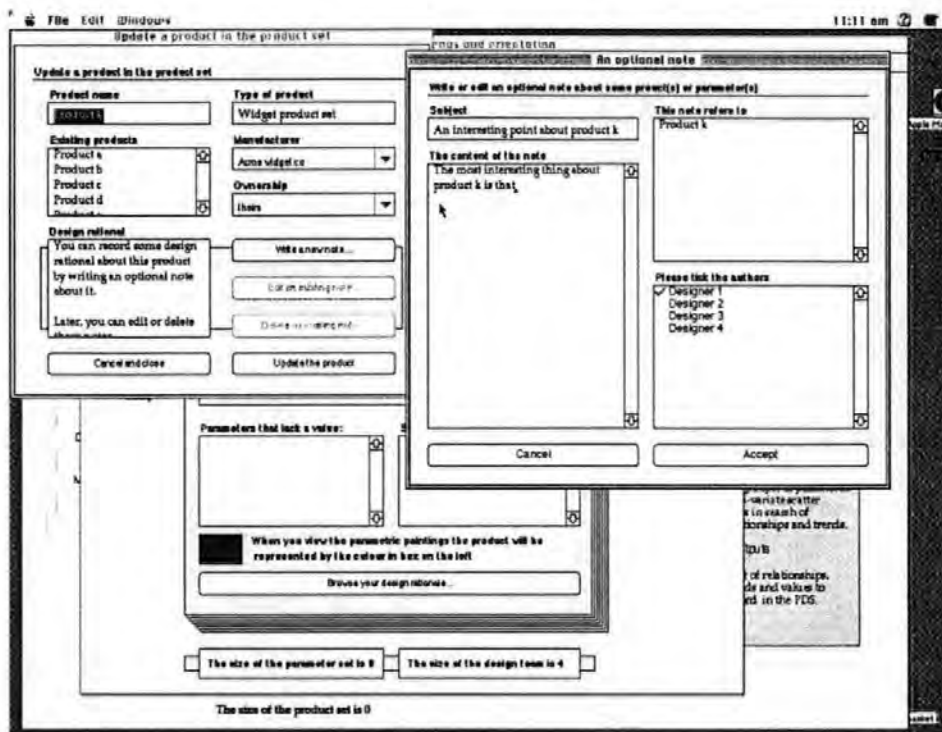


Figure 7.10: The update a product window showing the process of adding an optional note.

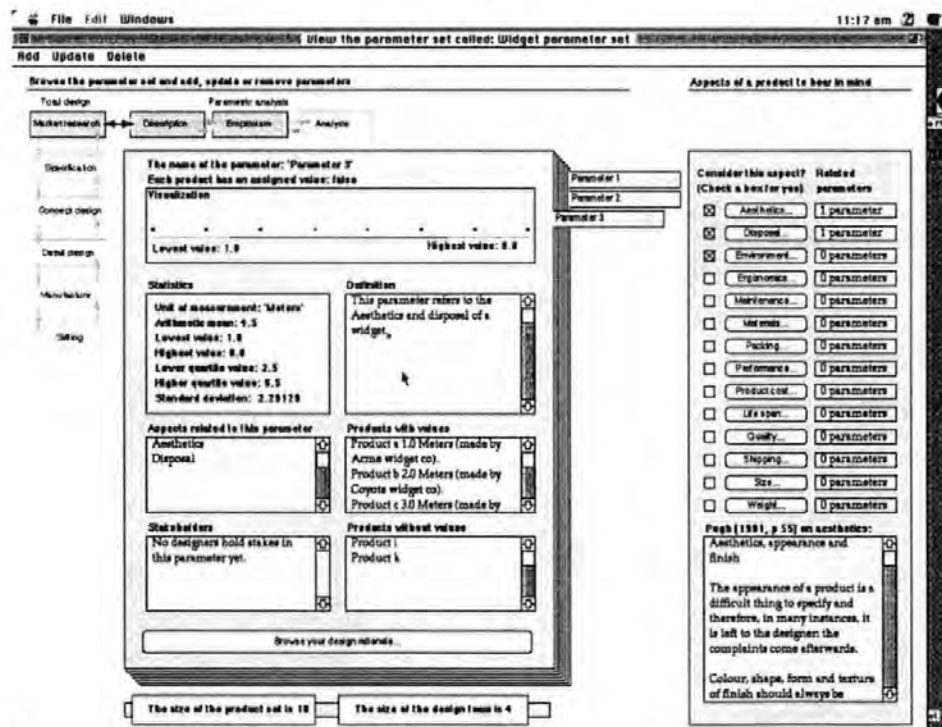


Figure 7.11: The parameter set window.

A final command allows product profiles to be deleted via a third modeless dialog window. Once a product's profile has been deleted, the entry for that product is removed from the note book in the product set window.

7.3.10 The detail design of the parameter set window and its children

The parameter set modeless window allows the designers to build and populate a parameter set. Hence it contains a chart which indicates that the parameter set is referred to throughout a parametric analysis (figure 7.11).

Once again the structure is similar to that of the design team window: it features a central note book and a menu bar that allows designers to add, update and delete parameters. However it also contains a passive design-informer that contains Pugh's advice about a number of standard aspects of a product to consider. Since this window contains two major work items, these will be discussed separately and they are also illustrated individually in figures 7.12 (a and b), which show the note book associated with the parameter set and a visualisation for the parameters, and figure 7.13, which shows the passive design-informer containing Pugh's advice and various other items of feedback.

The note book associated with the parameter set contains seven fields that present feedback to the designers and one button which allows them access to the design rationale associated with the parameter (figure 7.12 a).

The upper-most of these fields is illustrated in figure 7.12 (b). In this example, the field presents a visualisation of: the values associated with eight products for this parameter (the evenly spaced coloured regions), a range frame with a raised and a darkened central section which shows the upper and lower quartiles for the parameter, a bright point on the range frame showing the location of the arithmetic mean for these values. This field serves as a design visualiser (see Chapter Four) which provides a means for the designers to become familiar with the layout of the data for a parameter. Its design is sourced from Tufte's *The Visual Design of Quantitative Information* (1983) in which this particular arrangement of a range frame is proposed as a way to pack a maximal amount of information into as few pixels as possible.

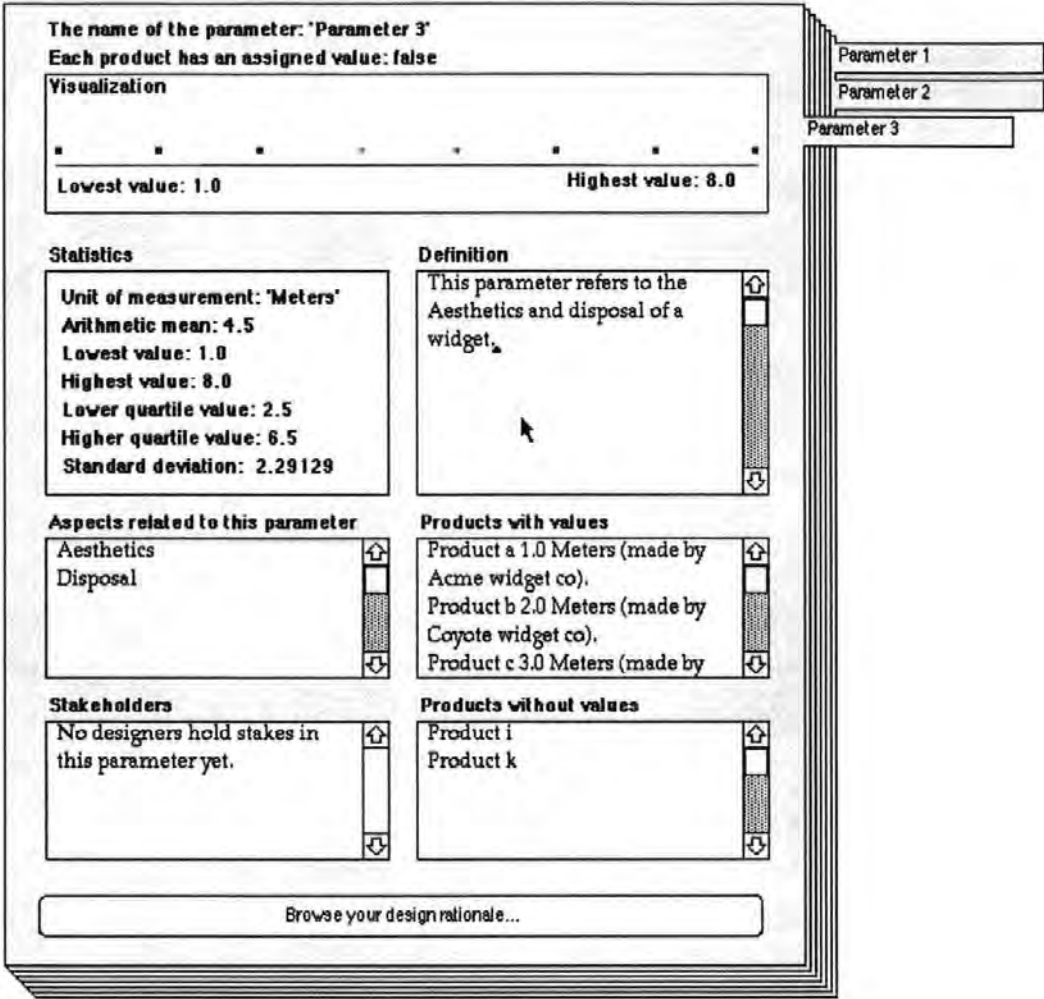


Figure 7.12 (a): The note book representing the parameter set.

Visualization



Figure 7.12 (b): The visualisation of the values and statistics for a particular parameter.

The subsequent fields present: the statistics which contribute to the visualisation; the definition of the parameter; the various standard aspects of a product which could be related to this parameter (see table 6.1, Hollins and Hollins, 1991, p 59; Pugh, 1991, pp 47-48); an alphabetical listing of the values of the products; a list of the stake holders for the parameter, and a list of the products without a value.

The second work item is an interactive design-informer. This is concerned with presenting design knowledge to help designers chose parameters and relate them to the content of the set as a whole (figure 7.13).

The screen contains fourteen rows. Each of these rows represents a particular aspect of a product which the total design method advocates that designers should consider (see table 6.1, Hollins and Hollins, 1991, p 59; Pugh, 1991, pp 47-48). The first item in each row is a check box to show whether the design team consider this aspect to be relevant to the current analysis. The second item is a button labelled with the name of the aspect. Pushing this button causes the text pane to display advice on this particular aspect of a product. The advice is quoted from Pugh's *Total Design* (1991). The text field on the right of the button indicates the number of parameters in the current parameter set which currently refer to this aspect of a product.

The first command on the menu associated with the parameter set window allows designers to create parameters in a modal dialog window (figure 7.14).

The first column in the window contains three fields that allow the designers to name the parameter, to view the names of existing parameters and to decide upon a numeric format (e.g., a real number, an integer, a fixed point, a currency or a year). Appropriate error checking is applied once a format is specified. For example, integers can be automatically cast in real numbers (and *vice versa*).

Consider this aspect? (Check a box for yes)	Related parameters
<input checked="" type="checkbox"/> Aesthetics...	1 parameter
<input checked="" type="checkbox"/> Disposal...	1 parameter
<input checked="" type="checkbox"/> Environment...	0 parameters
<input type="checkbox"/> Ergonomics...	0 parameters
<input type="checkbox"/> Maintenance...	0 parameters
<input type="checkbox"/> Materials...	0 parameters
<input type="checkbox"/> Packing...	0 parameters
<input type="checkbox"/> Performance...	0 parameters
<input type="checkbox"/> Product cost...	0 parameters
<input type="checkbox"/> Life span...	0 parameters
<input type="checkbox"/> Quality...	0 parameters
<input type="checkbox"/> Shipping...	0 parameters
<input type="checkbox"/> Size...	0 parameters
<input type="checkbox"/> Weight...	0 parameters

Pugh [1991, p 55] on aesthetics:

Aesthetics, appearance and finish

The appearance of a product is a difficult thing to specify and therefore, in many instances, it is left to the designer; the complaints come afterwards.

Colour, shape, form and texture of finish should always be

Figure 7.13: An interactive design-informer which helps the design team to select their parameters.

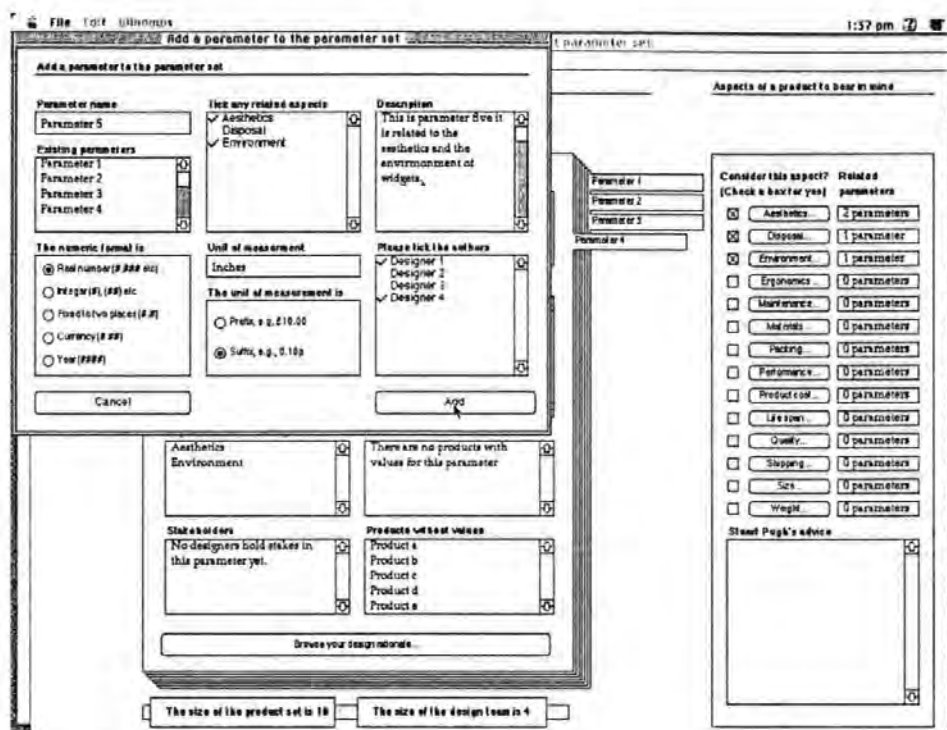


Figure 7.14: The add a parameter to the parameter set window.

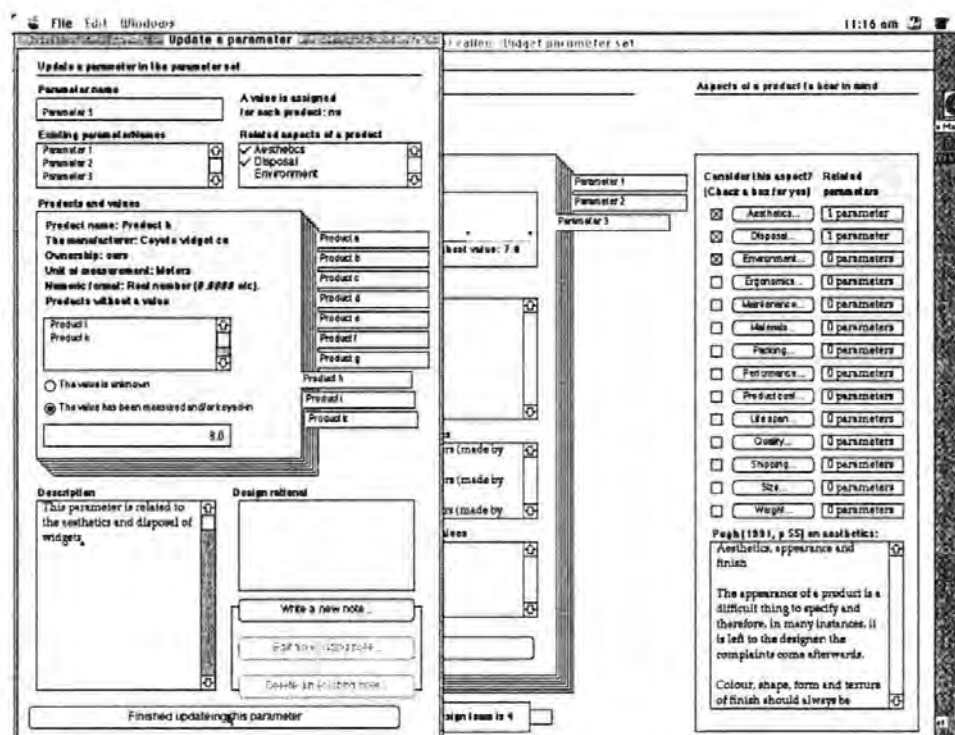


Figure 7.15: The update a parameter window.

The second column also contains three fields. The first field provides a list of the aspects of a parameter that the design team have decided to consider. The designers can select those aspects which they consider to be relevant to this parameter. Selecting such an aspect will mean that the parameter being defined will contribute to the tally of parameters relating to the aspect which is presented in the passive design-informer on the left of the parameter set window. The second field allows the designers to choose a unit of measurement for the parameter. The third field allows the design team to specify whether this unit is a prefix or a suffix.

The final column contains two fields which allow the designers to describe the parameter and to select authors for that description. This description will carry forward to the parameter set window and to the PDS document window (see below). Since the description is stored within the parameter alongside the other design rationale this part of the screen can be thought of as a specialised part of the design-documenter. A weakness in the current design, however, is that there is no provision for displaying the authorship of the description in the parameter set.

The second command on the menu associated with the parameter set window allows designers to update the definitions of their parameters and to associate values for the products from the product set (figure 7.15).

The values are entered at this stage in the program as it was envisaged that the design team would define an initial parameter set before populating it with values. A case was considered for repeating this mechanism in the product set. However, on the grounds of parsimony, it was decided not to implement such a feature unless and until it was requested by users.

The screen provides feedback on those products which currently lack a value and provides a note book of products where the design team can associate values for the parameter. The fields within this note book perform error checking on the values that are input and ensure that they are of the correct numeric type.

The final command on the menu associated with the parameter set window allows designers to delete parameters from the parameter set.

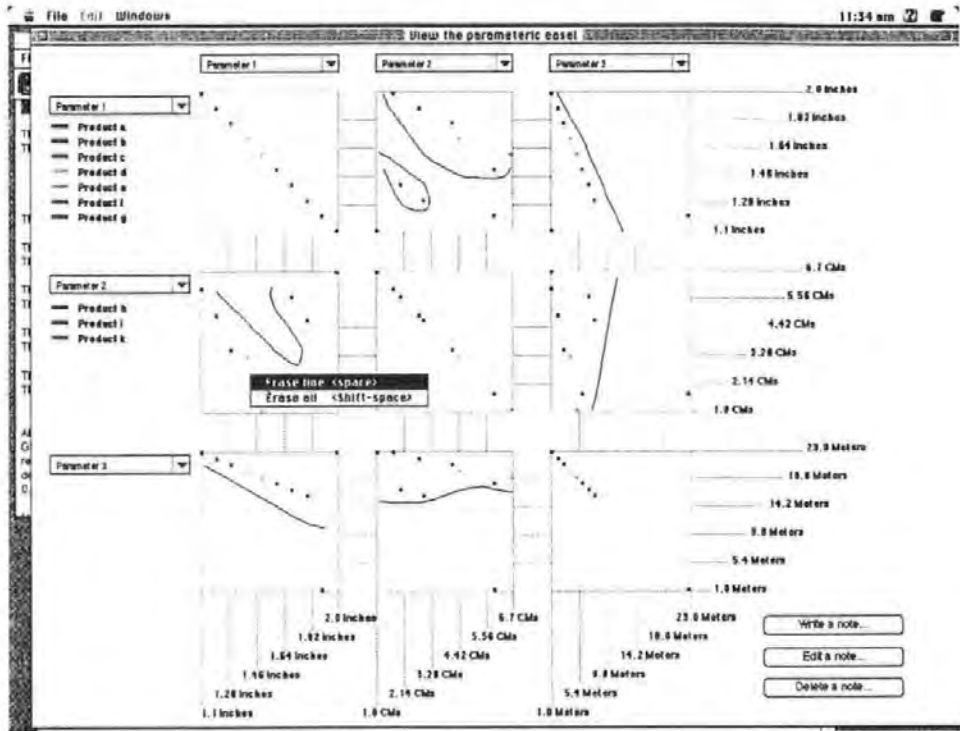


Figure 7.16a: The parametric easel.

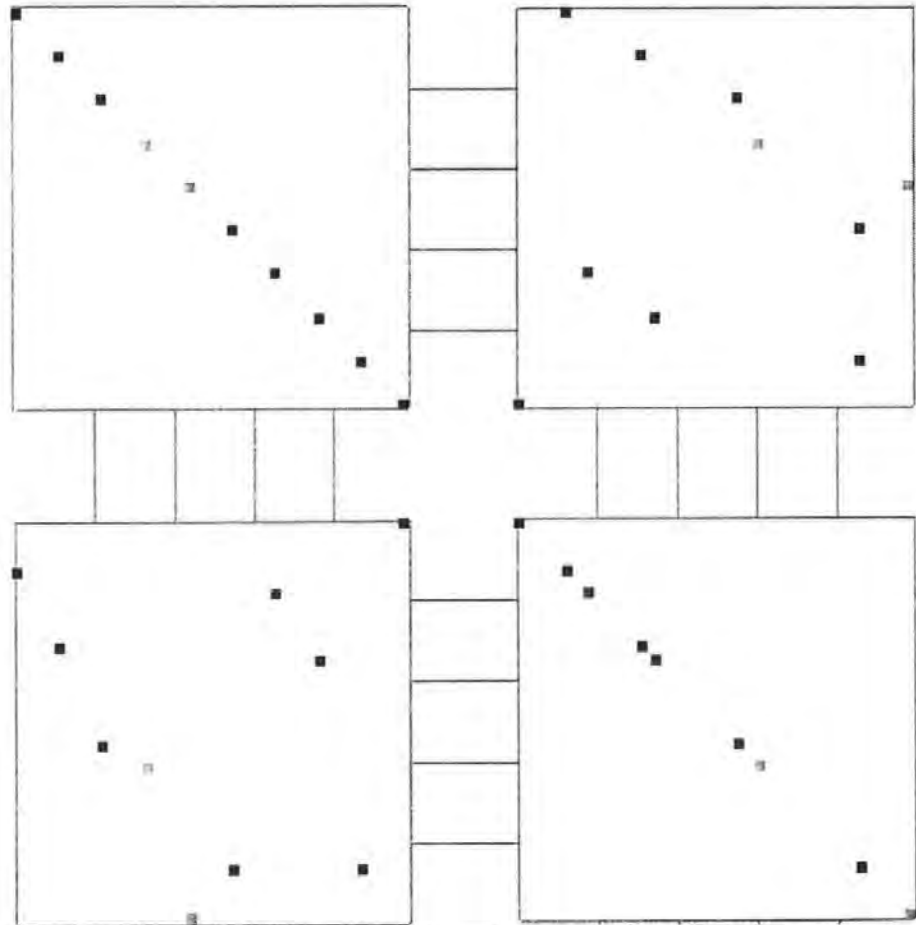


Figure 7.16b: A detail from the parametric easel showing how four scatter plots are 'knitted' together.

7.3.11 The detail design of the parametric easel window

The parameter easel modeless window allows the design team to view up to six parameters at once and to search them for clusters, correlations and other relationships (figure 7.16a). The screen's central work item simplifies Tufte's concept of a rug plot (Tufte, 1983). It was designed in order to address a point made in the original instructions for performing the method: "Grouping of plots, incorporating the same parameters, is always useful and revealing"(Pugh, 1991, p 37). The rug plot is composed of six scatter plots which are 'knitted' together by horizontal and vertical lines. These lines relate to the values listed on the right-most and lower-most sections of the screen. Figure 7.16 (b) shows four scatter plots to illustrate the concept. In each case individual 'product points' can be traced horizontally and vertically across the plots.

The contents of the individual scatter plots can be controlled by the pull-down menus on the left-most and upper-most sections of the screens. Each of these pull-down menus provides access to each of the parameters in the parameter set. Selecting a parameter in left-most pull-down menus controls the parameter that will be displayed across each scatter plot on the horizontally adjacent y-axis. Similarly, each of the upper-most pull-down menus controls the parameter that will be displayed on all three scatter plots on the horizontally adjacent x-axis. For example, in the top row of scatter plots in figure 7.16c the upper-most and left-most pull-down menus set each of the y axes to display 'parameter 1' and the pull-down menus in the upper-most row set the three x-axes to 'parameter 1' and 'parameter 2'.

The design team can elect to draw lines, curves and groupings to indicate trends, co-relations or other relationships in each scatter plot (figure 7.16d). This is achieved by placing the mouse within the boundaries of a particular plot and holding down the mouse button. These lines can be deleted using a menu associated with the 'alt' key and the mouse button.

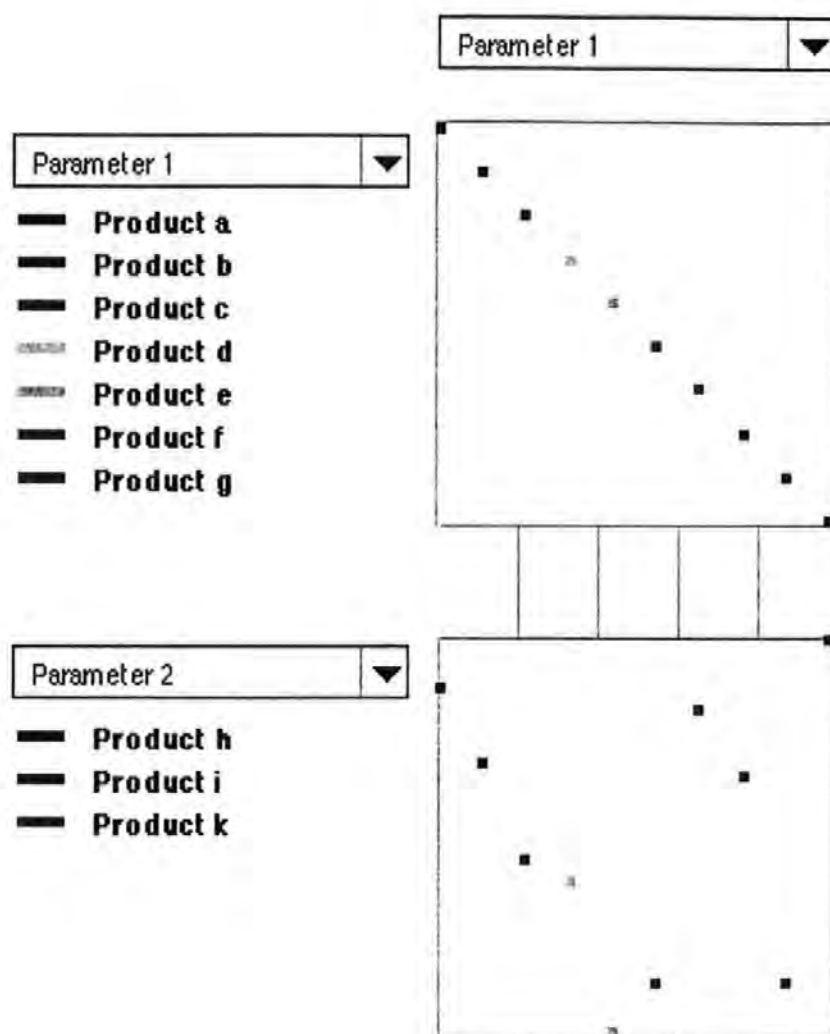


Figure 7.16c: Three pull-down menus causing 'parameter 1' and 'parameter 2' to be graphed against 'parameter 1'.

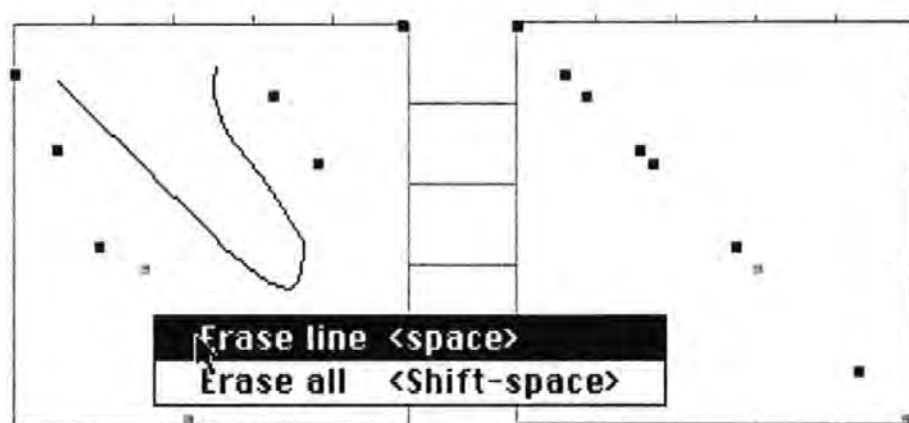


Figure 7.16d: Erasing a line on a scatter plot within the parametric easel.

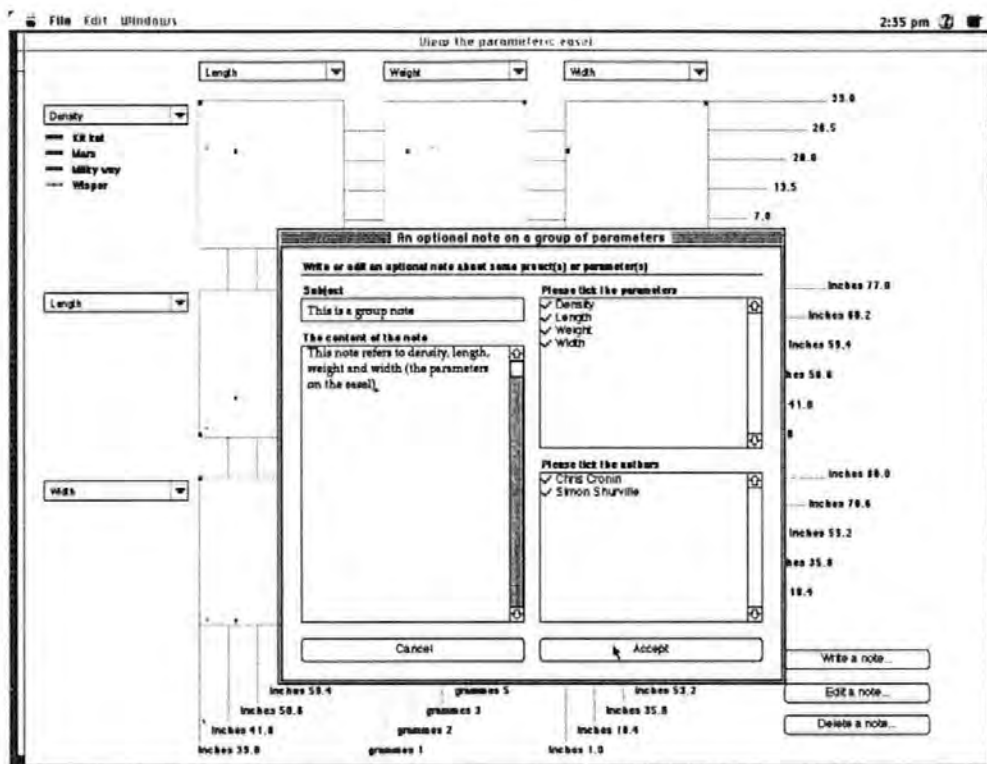


Figure 7.17: The special design rationale dialog for writing about multiple parameters.

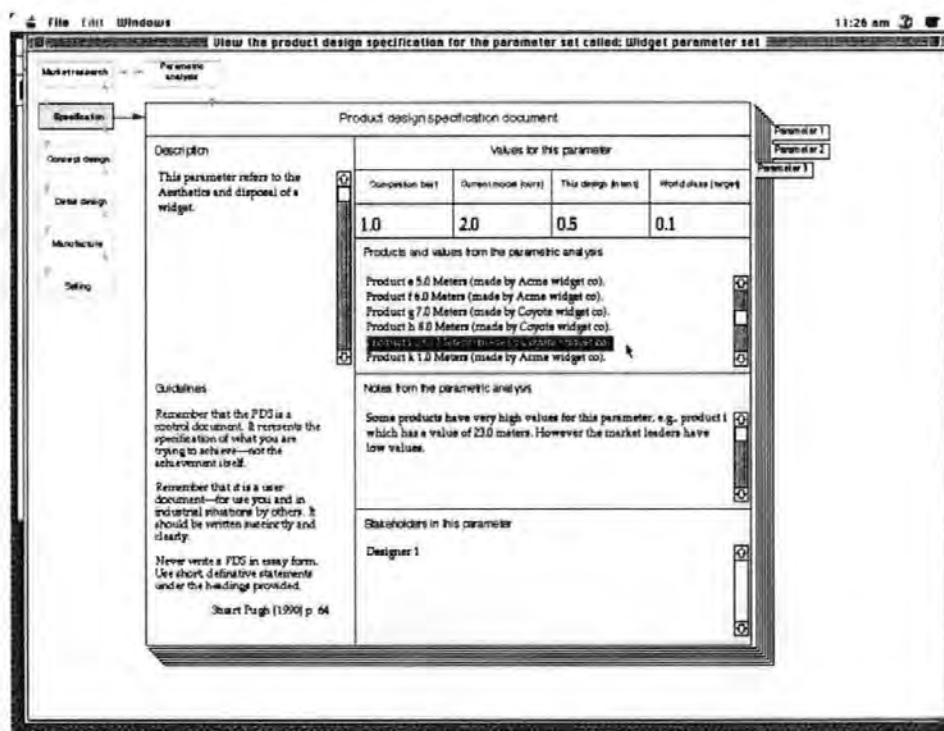


Figure 7.18: The PDS document window

When the design team have found a relationship between two or more parameters they can record the fact in a special design rationale dialog window (figure 7.17). Such a note can be related to any of the currently displayed parameters by selecting the parameter's name in a list box in the upper-most and right-most corner of the window. In this fictional example two designers have discovered a relationship between parameters representing a products' density, length, weight and width. In due course the content of the note will be attached to each of the selected parameters and it will therefore appear in the relevant entries in the program's representation of the PDS document.

7.3.12 The detail design of the PDS document window

The PDS document modeless window allows the design team to view a virtual PDS document (See the discussion in Chapter Six) presented on screen as a note book with a page for each parameter in the parameter set (figure 7.18).

This windows serves as an external representation in which the design team can review the outputs of the parametric analysis and construct elements of the PDS. The team can review the design rationale that they have created, e.g., descriptions of parameters and relationships between parameters. They can also inspect the product values, manufacturer information, and ownership information for each product and parameter. Finally, they can enter values for: their competitor's best product, the current model marketed by the sponsoring organisation, the current design, and a world class target. The note book is set in context by a flow chart which shows that the team have now left the analytic stage of the total design method and are now engaged in its specification stage.

7.4 An informal evaluation

The final stage in each iteration of Connell and Shafer's method is an informal evaluation. The main purpose is to ensure the program is within its target area. A secondary purpose is to discover users needs for new features and to delete features that users do not require. Since the evaluation was aimed at assessing the design of Patina as a *program* to support parametric analysis rather than at Patina as a *MODE*, it will be very briefly summarised here for the sake of completeness¹.

The trial was scenario-based with a subsequent questionnaire and a semi-structured group interview. A team of three evaluators was assembled consisting of a

¹ A full text reporting the evaluation is available from the author.

member of staff and two students from an MSc in product development at the University of Brighton. A small scale scenario involving cereal design was used in order to allow the team to interact with as much of the method and Patina as possible.

The team spent four hours role-playing the scenario with Patina. This involved describing the design team, adding values for products, selecting and adding parameters, discussing the relationships between parameters, discussing the relationships between parameters and the aspects of a product contained on the design-informer, writing design rationale about products and parameters with the design-documenter, studying parametric paintings to find relationships between parameters, drawing lines to indicate groups and trends on the parametric paintings, recording design rationale about such relationships, and starting to complete the PDS document. The participants tended to propose parameters to examine and then either deleted or adapted them on the basis of their 'experiments' on the parametric-easel, or of advice they found in the design-informers. This process was repeated in an iterative fashion until a parameter set emerged with which the team felt comfortable and which included a wide range of aspects of the product. This confirmed that the parametric-easel and the design-informers were useful tools which helped the team to tame the wickedness involved in designing an informative parameter set.

From the perspective of concept and detail design the participants' reactions to Patina were favourable. The team were well disposed toward the inclusion and implementation of MODE based features such as the passive design-informers expressing a preference for these to become even more interactive with the addition of hyper links to detailed knowledge about parametric analysis and its context. The team also asked for a domain oriented design-informer to proactively suggest aspects of the products under consideration that should be added to the analysis. This theme was repeated in the arena of the parametric easel: the participants requested additional knowledge based tools to automatically seek relationships with criteria set by the users.

In summary, the team requested that (1) more knowledge about parametric analysis and total design should be added to the existing design-informers, (2) some domain specific expert system style tools should be added that could be prepped with knowledge about particular types of product via by a user seeding process, and (3) some intelligent search facilities should be added to the parametric-easel. No requests were made for existing features to be removed. This confirms that Patina's present design functions well within its target area and suggests that it would be

worthwhile to prototype additional knowledge based and domain specific tools in a second version of the program.

7.5 Discussion

This section addresses the question of what can be inferred about the viability of implementing further MODEs from developing Patina.

Patina was implemented as a high fidelity object oriented prototype by following Connell and Shafer's advice to "Build something that really works and contains actual user-familiar data, instead of just a pretty screen prototype. Build it with tools that allow for performance tuning ... so that it is not just a requirements model and you will not have to start over again." (Connell and Shafer, 1995, p 19).

Consequently, the user interface described above communicates with underlying classes that store knowledge in a structure inspired by the structure of parametric analysis itself. For this reason Patina represents a working system which can be further enhanced with additional knowledge based tools that can share knowledge with the existing tools.

Since developing design-experts is a well developed technique, there are no immediately identifiable technical barriers to expanding Patina in the directions indicated by the informal evaluation. However, adding tools to transfer the content of the design-documenters to tools such as design-strategists remains a research area for Patina and MODEs in general as well as for DODEs and the process based approach.

Parametric analysis seems quite typical in terms of size, complexity and wickedness to the other methods described in the design research literature (see Chapter's Two and Three). So it appears justifiable to infer that similar programs could be built to support comparable methods. Ideally, a suite of MODEs could be implemented to support appropriate stages in a complete strategy for product design, such as the total design methodology.

There are two reasons to assume that this would be economically viable. One of the strengths of MODEs is that they can be quite self-contained systems which can interact with one another in the same way that prescriptive design methods can interact. Moreover, on the basis of the evidence of implementing Patina, which required six months of programming time, they are quite inexpensive to implement. So, providing common methods for accessing knowledge are employed, it appears that such a suite could be produced by a set of independent developers.

In summary, Patina provides an initial indication that developing other full size MODEs on the basis of the theory presented here is a technically and economically viable proposition.

7.6 Summary

The work reported here described the implementation of a Patina as a feasibility prototype of a MODE. This addressed the fourth hypothesis, showing that it was possible to implement an example of a MODE that was guided by the mapping between the design activity space and Green's model space. The following aspects of Patina's implementation were reported:

- The choice of Connell and Shafer's object oriented rapid prototyping method (1995) was motivated and a high fidelity prototype was chosen as it was felt this would lead to a realistic example of a MODE.
- Patina's concept design was guided by the mapping between the design activity space and Green's model space.
- This concept design was implemented as a detail design in the Small Talk programming language.
- The program's performance in the context of an informal trial with a group of target users confirmed that Patina's initial design was within its target area.
- The results of this small scale trial has set an agenda for further development which involves prepping more knowledge into the existing design-informers and adding some more domain specific and knowledge based tools to the program.

7.6.1 Suggestions for further work

Patina has demonstrated that the concept of a MODE can be implemented with available technologies. It has also been shown that a MODE can fulfil the needs of users engaged in a design activity located within the niche which was identified for MODEs in Chapter Five.

However, Patina needs further testing that concentrates upon its role as a MODE rather than as a program to support parametric analysis. Such testing should first

investigate whether a MODE works well in comparison with the use of prescriptive design methods without computer support. It should also compare the use of prescriptive design methods that are supported by other computer tools and systems, such as spreadsheets, DODEs and the process based approach, with support from a MODE. Further research should also address cognitive issues, such as whether the external representation of a method helps designers to concentrate on design rather than methodology.

Moreover, further examples of MODEs should be implemented and evaluated in order to fully support the proposal of this class of system for knowledge aided design.

However, as the discussion in section 7.5 concluded, this initial result is sufficiently encouraging to justify a programme of further research which could be pursued in a technically and economically viable way.

Chapter Eight

Conclusion

8.0 Overview

The main original outcomes of this research were:

- a description of design activities that included (1) the stages in the design process, (2) the required degree of innovation with components, and (3) the required degree of innovation with design strategies
- a classification of existing tools and systems for knowledge aided design in terms of this description of design activities
- the identification of a niche of design activities that present systems do not support, encompassing each stage in the design process and requiring an intermediate amount of innovation with components and design strategies
- the technical proposal of the class of MODEs to support activities in this niche in which the central tools represent a design strategy requiring an intermediate level of innovation
- a demonstration of this class of MODEs via the implementation of Patina
~ a MODE for parametric analysis.

This chapter contains five main sections. The first four sections describe the outcomes of the research in more detail, relating each outcome to a specific hypotheses that was presented in Chapter One. The fifth section suggests a programme of further research to develop the model of design, extend and evaluate Patina itself, and evaluate the potential of MODEs as a class of system for knowledge aided design.

8.1 Outcomes of describing design using knowledge from design research

Green (1992a) highlighted the problem that knowledge aided design has been largely restricted to implementing expert systems that solve previously formulated, well-defined design tasks. An original argument was presented attributing this problem to an overly-limited description of design that knowledge aided design has inherited from cognitive psychology and in particular from Simon (1969) (see Chapter Two). This suggested that a more detailed description would help clarify the range of existing tools and systems

for knowledge aided design and might highlight tasks for which new classes of system are required. This suggestion generated the following hypothesis

H 1 It is possible to construct a design activity space whose axes include strategies sourced from design research for characterising and addressing design projects. This new design activity space should enable new insights to be generated into the classes of knowledge aided design systems that can and should be created.

The outcome of addressing this hypothesis was a description of design which contained three relevant factors (see Chapter Three):

- (1) the main stages in design, i.e., analysis, synthesis and evaluation,
- (2) the amount of innovation with components, i.e., repeat orders, configurative variant-design, true variant-design, innovative-design, and strategic-design,
- (3) the amount of innovation with design strategies, i.e., well-defined first generation design strategies and more second generation design strategies where innovation is required.

This description of design differs from previous descriptions that were favoured in the field of knowledge aided design. It includes the stages of analysis and evaluation that were lacking from the descriptions of Simon (1969) and Chandrasekaren (1992) and includes a more detailed model of innovation with components than Chandrasekaran's description (1992). It also includes the notion of innovation with design strategies which is omitted by Simon and Chandrasekaren but which Goel and Pirolli (1989, 1992) included in a description aimed at studying problem solving within cognitive psychology. The description of innovation with design strategies presented here is also more detailed than that used by Goel and Pirolli. Although Blessing (1994) had included this notion within her proposal for process based design environments, she did not propose a description of design for general use within knowledge aided design that included this feature.

The strengths of this description of design become apparent in the work reported here for the following reasons:

- the description did enable new insights to be gained into existing tools and systems for knowledge aided design (see Chapters Four and Five)

- these insights were centred on features that had been lacking from previous descriptions, i.e., the stages of analysis and evaluation, the amount of innovation with components ranging between configurative variant-design and true variant-design, and the difference between supporting straightforward tame design strategies and more inventive wicked design strategies (see Chapters Four and Five)
- these insights led to the identification of these types of design as being in need of support which motivated the proposal of a class of system to offer such support (see Chapter Five).

In summary, the description of design that was developed to address the first hypothesis has included a set of features from design research that are new to the theory of knowledge aided design. As the remaining sections of this chapter will testify, these features have helped to generate new insights into the classes of knowledge aided design systems that can and should be created.

8.2 Outcomes of using the description of design tasks to classify tools and systems for knowledge aided design

The description of design has been used to classify a catalogue of tools for knowledge aided design in order to (1) understand why existing tools are limited in range, (2) estimate the range of new tools, and (3) show that two classes of system configured from such tools are not suited to as wide a range of design activities as might first appear. This analysis addressed the second hypothesis:

H 2 It is possible to develop mappings between this design activity space and Green's model space of knowledge aided design systems. These mappings should enable the approximate range of applicability of individual tools and classes of systems to be indicated. These mappings should also make it possible to re-characterise the accepted range of applicability for some existing classes of system.

The analysis of tools focused on a set of tools for knowledge aided design (Green, 1992a). These tools included design-aides, design-demons, design-documenters, design-experts, design-informers, design-secretaries, design-strategists, and design-translators, and design-simulators. Descriptions of these tools were presented (see Chapter Four) together with a description of a new tool called a design-visualiser that is implemented in Patina (see Chapter Seven).

To prepare for the main categorisation the tools were first analysed in terms of seven features which Green had proposed (1992a). A second ingredient of the categorisation was an original observation of a match between glass- and black-box design activities and the types of automatic and interactive tools for knowledge aided design (see Chapter Three).

The tools for knowledge aided design were classified in relation to the categories of the description of design (see Chapter Four). Those findings which are key to the categorisation of systems for knowledge aided design are indicated in table 8.1. This table highlights the tools that faired best in terms of the seven features identified by Green for each factor in the description of design.

Stages of design

	Analysis	Synthesis	Evaluation
Design-demon	-	+	N/A
Design-documenter	+	+	+
Design-expert	-	+	-
Design-informer	+	+	N/A
Design-secretary	+	+	+
Design-simulator	-	+	-

Innovation with products

	Low	Med	High
Design-demon	+	-	-
Design-documenter	+	+	+
Design-expert	+	-	-
Design-informer	+	+	+
Design-secretary	+	+	+
Design-simulator	+	-	-

Innovation with design strategies

	Low	Med	High
Design-demon	+	-	-
Design-documenter	+	+	+
Design-expert	+	-	-
Design-informer	N/A	+	+
Design-secretary	+	+	+
Design-simulator	+	-	-

Table 8.1: An indication of the suitability of various tools to design activities. A plus sign indicates that the tool is suitable to a value for a factor, a minus sign indicate that the tool is unsuitable to a value for a factor and an N/A indicates that the tool is neither especially suited nor unsuited to a value for a factor. Arguments to support these indications of suitability can be found in Chapter Four.

Since the main goal was to expand the range of systems for knowledge aided design, it was necessary to estimate the design activities to which traditional systems and their more recent alternatives were suited. DODEs (Fischer and co-workers 1992, 1994a,

1994b, 1996; Silverman, 1992) and the process based approach (Blessing, 1994) were selected as they represent alternatives to traditional systems while still adapting the tools that have already been discussed.

In the case of knowledge based systems the necessary facts for the estimation had largely been provided by Green (1992a). Green argued that these systems are suited to synthesising specific types of product via heuristic search within the boundaries of well-specified and pre-defined problems. So the following estimations could be extrapolated from established arguments:

Stages of design: knowledge based systems are well-suited to synthesis and less well-suited to analysis and evaluation (particularly as the degree of innovation with components increases).

Innovation with products: knowledge based systems are suited to configuring pre-existing components but are not suited to innovation with components.

Innovation with design strategies: knowledge based systems are suited to very simple and repetitive design strategies but are not suited to strategies that require innovation.

DODEs were selected to represent a frequently implemented type of system, which have been claimed to 'enhance creativity' (Fischer, 1993) or, more specifically, innovation with products. DODEs have not previously been categorised using a description of design with the factors in use here. So the categorisation revealed some new knowledge about this class of systems. A full report of the design tasks for which DODEs appear to be applicable was presented in section 5.2. The key findings are reported below:

Stages of design: DODEs appear well-suited to synthesis and less well-suited to analysis and evaluation in the same way as traditional knowledge based systems. This is because the coupling of design-simulators and design-demons means that it is difficult to analyse a product in terms of new components and difficult to evaluate new components or existing components in new ways.

Innovation with products: DODEs appear well suited to design tasks where the level of innovation required consists of the entirely routine configuration of existing components. This is because design-simulators support the

synthesis of existing components into a 'plan' representing an evolving product and design-demons bring pre-existing knowledge to bear to critique the plan as it evolves. For this reason DODEs do not appear well-suited to tasks that require either an intermediate or high level of innovation with components.

Innovation with design strategies: DODEs appear suited to supporting the strategy of heuristic search which is performed within design tasks where a low level of innovation is required. This is because their design-demons are based on technologies inherited from automated 'planning'. They do not appear suited to supporting tasks where an intermediate or high level of innovation with design strategy is required.

Although Blessing's process based approach has yet to be implemented, it was selected for categorisation because the system design set out to support both straightforward and innovative design strategies. A full report of the factors for which Blessing's proposal for a process based environment appears to be applicable was presented in section 5.4. The key findings are reported below:

Stages of design: The central tools in Blessing's proposal for process based environments are a design matrix, which functions as a passive design-informer containing design-secretaries and design-documenters, that depicts each stage of design within a generic representation. So process based environments appear well suited to supporting the stages of analysis, synthesis and evaluation.

Innovation with products: The design-secretaries and design-documenters that are central to a process based environment are well suited to product design that requires high degrees of innovation. This is because they allow designers to describe new products and share this information with one another. Although such tools can be used for less innovative projects they appear less suitable than automated tools that have been prepped with appropriate knowledge.

Innovation with design strategies: Process based environments appear well suited to tasks where a high degree of innovation with design strategies is required. This is because the design matrix, which functions as a passive design-informer, depicts a

very general design strategy that does not predict the details of the actual strategy that will be used. However, this configuration of a passive design-informer appears less suited to tasks where a well structured strategy is adapted to meet the demands of a particular project.

In summary, the categorisation of tools and systems that was developed to address the second hypothesis have generated two main original insights. First, it was argued that DODEs are not as suited to supporting creativity with products as had previously been claimed (c.f., Fischer, 1993). Second, it was argued that a process based approach which is centred upon a passive-design informer that depicts Blessing’s design matrix might be too general a representation for projects requiring an intermediate degree of innovation with design strategies. In the next section these conclusions will underpin the definition of a class of systems to support projects that require an intermediate amount of innovation with products and design strategies.

Stages of design

	Analysis	Synthesis	Evaluation
Knowledge based system	-	+	-
DODE	-	+	-
Process based approach	+	+	+
In need of support	√	√	√

Innovation with products

	Low	Med	High
Knowledge based system	+	-	-
DODE	+	-	-
Process based approach	+	+	+
In need of support	√		

Innovation with design strategies

	Low	Med	High
Knowledge based system	+	-	-
DODE	+	-	-
Process based approach	-	-	+
In need of support	√		

Table 8.2: An indication of the suitability of various systems to design activities. A plus sign indicates that the tool is suitable to a value for a factor, a minus sign indicates that the tool is unsuitable to a value for a factor. A tick indicates that a type of design activity is in need of support from a new type of system. Arguments to support these indications of suitability can be found in Chapter Five.

8.3 Outcomes of identifying a need for a new class of system and of proposing MODEs

This section reports upon the identification of a coherent niche of design tasks that are not supported by existing classes of system and therefore require support from a new type of system. This work was performed to address the third hypothesis:

H 3 It is possible to search the design activity space for at least one niche that is unsupported by existing classes of system and to propose a new class of system, composed of tools located in Green's model space, that appears suited to supporting activities in this niche. The use of the design activity space should indicate that the particular class that has been proposed is suited to supporting activities in the niche and that the surrounding classes of system are unsuited to offering such support.

The classifications of systems for knowledge aided design that have been summarised are collated in table 8.2. This table also indicates a set of design activities within the description of design that are not supported by current classes of system. These design activities appear to form a coherent niche in need of support from a new class of system. The types of activities which are located in this niche are described in terms of the factors in the description of design tasks below:

Stages of design: The stages of analysis and evaluation are only supported by process based design environments, which have yet to be implemented and tested. Although the stage of synthesis is supported by knowledge based systems and DODEs, as Lawson (1990) indicates, synthesis requires some analysis and evaluation. So support for all three stages is desirable in any system aimed at innovation with products or strategies.

Innovation with products: Knowledge based systems and DODEs support low levels of innovation with products. Process based environments appear to support low, intermediate and high levels of innovation. However it has been argued that innovation with products is often coupled with innovation in design strategy (see Chapter Five). It has also been argued that the current design for process based environments appears unsuited to low levels of innovation. So support with intermediate levels of innovation with products appears to be required.

Innovation with design strategies: Knowledge based systems and DODEs support design strategies where no innovation is required. Although Blessing's proposed process based environment sets out to support straightforward design strategies and strategies where innovation is required, it has been argued that this approach may not be as effective for strategies requiring an intermediate amount of innovation. So support for intermediate levels of innovation with design strategies appears to be required.

The arguments presented here indicate that a niche of tasks that require support from a new class of system exists between the areas of the design activity space which are supported by DODEs and process based approaches. It has been shown that at least one design method can be located within this niche (see Chapter Six).

Turning now to the proposal of a class of system to support such activities, there are two criteria for proposing a new class of software (1) that it will address a need, and (2) that it is technically viable. Now that arguments indicating the need for a new class of software have been presented it is time to turn to the issue of inventing a technically viable format.

This fine-grained description of design activities and tools and systems for knowledge aided design that have been developed allow a third class of system to be created 'between' the existing categories of DODEs and process based environments. This third category can be generated by subdividing the process based approach into its constituent elements of first generation and second generation design methods. That is, into well structured approaches with an intermediate level of innovation and more loosely structured approaches with a high level of innovation. For this reason the new class of system is named method oriented design environments to show that the method, rather than the domain, is the central concept and that a structured 'method', rather than a less-structured 'process', is the main practice being supported.

This proposal has been made because first generation methods offer well structured and proven design knowledge which can be depicted as familiar and 'intelligent' interface items. For example, directed and annotated graphs are familiar and effective items in both knowledge aided design systems and in design education. Such graphs might therefore replace design-simulators as the central work item in a knowledge aided design system. If they are implemented as passive design-informers or design-aids, then these graphs could act as repositories of strategic design

knowledge which designers could either browse through or navigate by. If they are implemented as more interactive design-strategists, then the structures in such graphs could provide logical knowledge to empower processes of mechanical deduction and inference. In either case it appears that the nature and format of the resulting product will not be governed by the knowledge within a MODE to the same extent as it would be by the design-simulator within a DODE.

This definition of MODEs appears viable from a technical perspective because it adapts technologies that have been successfully implemented as DODEs and concepts that have been shown to be plausible by Blessing (1994). A selection of tools that are suited to the various factors of design that have been considered are summarised below:

Stages of design: The stages that stand in special need of support are analysis and evaluation. The proposal is that MODEs include design-documenters and design-secretaries to provide external representations where designers can define their terms of analyses and evaluation and, indeed, the components they are investigating. Passive design-informers are suited to providing textual and graphical guidance through these stages. The stage of synthesis can be supported with abstract design-simulators, such as the design-visualiser (see Chapter Four) which are not tied to representing particular pre-defined products or components.

Innovation with products: The level of innovation that stands in need of support is intermediate. The proposal is that MODEs include design-documenters and design-secretaries to record information about new products and components and access information about familiar products and components through the interface to these tools. This implements the kind of user seeding and evaluation that was discussed in the context of DODEs in section 5.3.

Innovation with design strategies: The level of innovation with design strategies in need of support is intermediate. The proposal is that passive design-informers depict the structure of an existing method and provide a textual explanation of the method and its stages. These design-informers should also provide feedback about the progress through the method and the values of its key variables throughout the interface to the MODE. If the deviation from the method is small then the designers may be able to keep

track of this without external support. However, a means to represent significant changes needs to be developed (see section 8.5).

In summary, the work presented here has identified a set of design tasks that are not supported by existing classes of system and proposed MODEs as a new class of system that appears suited to supporting activities in this niche. A demonstration of a MODE will be discussed in the next section.

8.4 Outcomes of implementing Patina ~ a MODE for parametric analysis

This section reports how the technical viability of developing a MODE was confirmed through the implementation of Patina ~ a MODE for Parametric Analysis. The process of implementing Patina also demonstrates the viability of using the description of design activities presented above to orient system design. This shows that analysis is located within the niche of design activities that has been identified for support via MODEs. This work addressed the fourth hypothesis:

H 4 It is possible to design and implement an example of this new class of system by locating a design practice in the design activity space and then applying the mapping to Green's model space in order to propose an aggregate of components that will support this practice.

The first step in implementing a MODE was to select a design method which matched the niche that had been identified. Parametric analysis (Hollins, 1989; Hollins and Pugh, 1990; Pugh, 1991; Hollins and Hollins, 1991) was chosen because: it met this criteria (see section 6.4); it is used by students of design and professional designers; and its strategy and usage are well documented (Hollins, 1989; Hollins and Pugh, 1990; Pugh, 1991; Hollins and Hollins, 1991).

An analysis of the method itself revealed areas that were complicated and labour intensive, which appeared amenable to support with the design-visualiser (see Chapters Four and Six), and where innovation was required in the selection of products and ways to describe them, which appeared amenable to support via design-informers.

Patina was implemented as a MODE to support parametric analysis using Connell and Shafers' object oriented prototyping method (1995). The task of building a high fidelity feasibility prototype of a MODE to support this method was addressed in three stages: (1) initial requirements were gathered by interviewing potential users of the program and interviewing Dr William Hollins, an expert in the history, design and use

of parametric analysis, and then reviewing literature from the subject area, (2) a prototype was implemented in the Small Talk programming language, (3) the prototype was presented for an informal evaluation to a group of potential users in the context of a structured scenario to elicit their opinions about the software's design and utility.

Patina set out to support the complicated and labour intensive areas in the method and to help designers to choose products and parameters that would provide useful insights. The fact that parametric analysis could be located in the niche which led to the proposal of MODEs indicated that a MODE was a viable starting point to orient system design.

Patina was developed with reference to the arguments about the suitability of various tools for the various factors in the description of design discussed above (see Chapter Seven). These led to an architecture in which the central work item was a passive design-informer that represented a graphic and textual description of the method itself and provided feedback on the users' progress through a particular analysis. Design-secretaries were used to collect information about products and ways to describe them which were stored in a central knowledge base that was designed to reflect the structure of the method itself. Design-visualisers were added to these design-secretaries to provide immediate feedback about the relationships between parameters and products. Other design-informers provided feedback on the way that the products were currently described and standard aspects of a product that might need to be included within an analysis. One innovative tool was a design-visualiser, adapted from Tufte's rug plot (1983), which allowed designers to view multiple relationships between numeric representations of products.

In an informal evaluation of the program, whose purpose was to ensure that its features fit within the target area, a team of three evaluators used the system to conduct a miniature parametric analysis. From the perspective of concept and detail design the participants' reactions to Patina were favourable. The team were well disposed toward the inclusion and implementation of MODE based features such as the passive design-informers. However they requested that these become even more interactive with the addition of hyper links to detailed knowledge about parametric analysis and its context. The team requested a domain oriented design-informer to proactively suggest aspects of the products under consideration that should be added to the analysis and additional knowledge based tools to automatically seek relationships with criteria set by the users. No requests were made for existing features to be removed.

Although an informal evaluation can never be totally conclusive, the opinions of these evaluators tended to confirm that Patina's present design functions well within its target area. The evaluators requests suggest that it would be worthwhile to prototype

additional knowledge based and domain specific tools in a second version of the program. In section 8.5 the kinds of more formal evaluations of Patina and of MODEs in general that are needed to advance the theory that has been presented here will be indicated.

The discussion of Patina in its role as a MODE suggested that since parametric analysis is comparable to other design methods (see Chapter's Two and Three, it can be inferred that MODEs are a feasible way to support comparable methods. It was suggested that MODEs appear to be a cost-effective type of system to develop. This is because MODEs are quite self-contained systems yet they can in principle interact with one another in the same way that prescriptive design methods can interact in a larger design-strategy. The successful implementation of Patina therefore indicates that other MODEs could be implemented.

In Summary, developing Patina as a MODE for parametric analysis has shown that it is possible to design and implement an example of a MODE by locating a design practice in the design activity space and then applying the theoretical conclusions presented in Chapters Three, Four, and Five to design a system to support this practice. Since this system is an example of the new class of MODEs, this implementation shows that the scope of knowledge aided design can be extended. Moreover, since problem setting / analysis are central to parametric analysis, the implementation of the program indicates that the range of systems for knowledge aided design can be extended in this direction which lies beyond the scope of traditional knowledge based technologies.

8.5 Further research

The suggestions for further research are as follows:

- (1) extending the description of design tasks that has been developed here,
- (2) investigating whether MODEs can be implemented to support other design methods,
- (3) investigating whether the external representations of design strategies that are central to the definition of MODEs enable designers to free their mental resources from representing strategies and thereby concentrate on the task itself,

- (4) investigating whether an adaptable interface could allow users to represent adaptations that they are making to the prescriptions and structure of a given design method,
- (5) performing long term evaluations with Patina, and
- (6) extending Patina to include some domain specific design-experts and design-informers.

Bibliography

- Agre P. (1997) *Computation and Human Experience*, Cambridge: Cambridge University Press.
- Akin (1979) *An Explanation of the Design Process*, Design Methods and Theories, number 13, pp 115 – 119.
- Alexander C. (1962) *Notes on the Synthesis of Form*, Cambridge, MA: Harvard University Press.
- Alexander C. (1963) *The Determination of Components for an Indian Village*, in Jones J. and Thornley D., (eds.), 1963.
- Alexander C. (1971) *Notes on the Synthesis of Form* (Second Edition), Cambridge, MA: Harvard University Press.
- Alexander C. and Jacobson M. (1971) *The State of the Art in Design Methods*, DMG Newsletter, volume 5, number 3, pp 3 – 7.
- Alexander C., Silverstein M., Angel S., Ishikawa S. and Abrams D. (1975) *The Oregon Experiment*, New York, NY: Oxford University Press.
- Alexander C., Ishikawa S., Silverstein M., Jacobson M., Fiksdahl-King I. and Angel S. (1977) *A Pattern Language*, New York, NY: Oxford University Press.
- Allen J., Hendler J. and Tate A., (eds.) (1990) *Readings in Planning*, San Mateo, CA: Morgan Kaufmann Publishers.
- Archer B. (1965) *A Systematic Method For Designers*, London: Council for Industrial Design.
- Archer B. (1969) *The Structure of the Design Process*, in Broadbent G. and Ward A. (eds.), 1969
- Archer B. (1979) *Whatever Became of Design Methodology?*, Design Studies, volume 1, number 1, pp 17 – 18.
- Asimow W. (1962) *Introduction to Design*, Englewood-Cliffs, NJ: Prentice-Hall.
- Barley N. (1992) *Design in a Tribal Context*, Journal of Design History, volume 5, number 2, pp 103 – 111.
- Barry A. (1996) *The European Network*, New Formations, number 29, pp 26 – 38.
- Baudrillard J. (1981) *Simulacra and Simulation*, Ann Arbor, Michigan: University of Michigan Press.
- Beardon C., Gollifer S., Rose C. and Worden S. (1995) *Artists and Designers as Users*. Proceedings of Computers in Context: Joining Forces in Design, Third Decennial Conference, Denmark, 14 – 18 August 1995, pp. 90 – 98.
- Biggs L. (1995) *The Engineered Factory*, Technology and Culture, volume 36, number 2, pp 174 – 188.
- Blessing¹ L. (1994) *A Process-Based Approach to Computer-Supported Engineering Design*, Thesis prepared at the University of Twente, Enschede, Netherlands.
- Blessing L. (1996) *Design Process Capture and Support*, in Tichem M. (ed.) (1996).
- Boden M. (1988) *Computer Models of Mind: Computational Approaches in Theoretical Psychology*, Cambridge: Cambridge University Press.
- Boden M. (1990) *The Creative Mind: Myths and Mechanisms*, London: Weidenfeld and Nicolson.
- Boden M. (1993) *Creative Constraints and Unpredictability*, in Edmonds E. (1993) (no page numbers are given in these pre-prints).
- Boden M. (ed.) (1996) *The Philosophy of Artificial Life*, Oxford; Oxford University Press.
- Bond A. (1992) *The Co-operation of Experts in Engineering Design*, in Gasser and Huhns (eds.) (1992).
- Booker P. (1963) *A History of Engineering Drawing*. London: Chatto and Windus.

¹See also L. Stomph-Blessing.

- Brachman R. (ed.) (1985) *Readings in Knowledge Representation*: Los Altos, CA: Morgan Kaufmann Publishers.
- British Standards Institution (1995) *Design Management Systems: Part Ten, Glossary of Terms Used in Design Management*, Chiswick: British Standards Institution.
- Broadbent G. (1967) *Notes on Design Method*, in Broadbent G. and Ward A. (eds.) (1969).
- Broadbent G. and Ward A. (eds.) (1969) *Design Methods in Architecture*, London: Lund Humphries Publishers Limited.
- Brodie K., Carpenter L., Earnshaw J., Gallop R., Hubbard R., Mumford A., Osland C. and Quarendon (eds.) (1991) *Scientific Visualisation—Techniques and Applications*, Berlin: Springer-Verlag.
- Brown D. and Chandrasekaran B. (1989) *Design Problem Solving: Knowledge Structures and Control Strategies*, London: Pitman.
- Bucciarelli L. (1988) *An Ethnographic Perspective on Engineering Design*, Design Studies, volume 9, number 3, pp 159 – 168.
- Bucciarelli L. (1994) *Designing Engineers*, Cambridge, MA: MIT Press.
- Buchanan R. (1995) *Wicked Problems in Design Thinking*, in Margolin V. and Buchanan R. (eds.) (1995).
- Budgen D. (1995) *Design Models from Software Design Methods*, Design Studies, volume 16, number 3.
- Burnett M. and Baker M. (1994) *A Classification System for Visual Programming Languages*, Journal of Visual Languages and Computing, September, 1994, pp 287 – 300.
- Candy L. and Edmonds E. (1996) *Creative Design of the Lotus Bicycle: Implications for Knowledge Support Systems Research*, Design Studies, volume 17, number 1, pp 71 – 90.
- Chandrasekaran B. (1992) *Design Problem Solving*, in Green (ed.) (1992).
- Chapman D. (1987) *Planning for Conjunctive Goals*, Artificial Intelligence, number 32, pp 333 – 337.
- Clark A. (1989) *Microcognition: Philosophy, Cognitive Science and Parallel Distributed Processing*, Cambridge, MA: MIT Press.
- Clark P. (1988) *Organisation Transitions and Innovation-Design*, London: Pinter.
- Cleveland W. (1994) *A Science of Graphical Data Display*, in Penman R. and Sless D. (eds.) (1994).
- Colman W. (1994) *Presenting Mathematical Information*, in Penman R. and Sless D. (eds.) (1994).
- Connell J. and Shafer L. (1995) *Object Oriented Rapid Prototyping*, Englewood Cliffs, NJ: Yourdon Press Computing Series.
- Coyne R., Rosenman M., Radford A., Balachandran M. and Gero J. (1990) *Knowledge-Based Design Systems*, Reading, MA: Addison-Wesley.
- Coyne R. and Snodgrass A. (1993) *Rescuing CAD from Rationalism*, Design Studies, volume 14, number 2, pp 100 – 123.
- Crickmay C. and Jones J. (1972) *Imagination and Method: Designing as a Response to Life as a Whole*, Bletchley, Buckinghamshire: Open University Press.
- Cross N. (1975) *Design Methods Manual*, Bletchley, Buckinghamshire: Open University Press.
- Cross N. (1984) (ed.), *Developments in Design Methodology*, New York, NY: John Wiley and Sons.
- Cross N. (1989) *Engineering Design Methods*, Bletchley, Buckinghamshire: Open University Press.
- Cross N. (1994) *Strategies for Product Design*, Chichester: John Wiley and Sons.
- Cross N., Dorst K. and Roozenburg N. (1992) *Research in Design Thinking, Workshop 29 – 31 May, 1991*, Technical University Delft: Delft University Press.

- Cross N., Naughton J. and Walkers D. (1981) *Design Method and Scientific Method*, Design Studies, volume 2, number 4, pp 195 – 201.
- Culverhouse P. (1995) *Constraining Designers and their CAD Tools*, Design Studies, volume 16, number 1, pp 81 – 101.
- de Camp (1963) *The Ancient Engineers*, New York, NY: Doubleday.
- de Landa M. (1991) *War in the Age of Intelligent Machines*, New York, NY: Zone Books.
- de Vries M., Cross N. and Grant D. (1993) *Design Methodology and Relationships with Science*, Boston, MA: Kluwer Academic Publishers.
- Draper S. and Norman D. (1986) (eds.), *User Centred System Design: New Perspectives on Human-Computer Interaction*, Hillsdale, NJ: Lawrence Erlbaum Associates Inc.
- Dreyfus H. (1992) *What Computers Still Can't Do: A Critique of Artificial Reason*, Cambridge, MA: MIT Press.
- Druckner P. (1985) *Innovation and Entrepreneurship*, London: Heinmann.
- Earnshaw R. and Wiseman N. (1992) *An Introduction to Scientific Visualisation*, Berlin: Springer-Verlag.
- Elton M. (1993) *Towards Artificial Creativity*, in Edmonds E. (ed.) (1993).
- Everitt B. (1978) *Graphical Techniques for Multivariate Data*, London: Heinemann Educational Books.
- Ferguson E. (1992) *Engineering the Mind's Eye*, Cambridge, MA: MIT Press.
- Fischer G. (1992) *Domain Oriented Design Environments*, Proceedings of the Seventh Knowledge-Based Software Engineering Conference, IEE Computer Society, pp 204 – 213.
- Fischer G. (1993) *Creativity Enhancing Design Environments*, in Gero J. and Maher M. (eds.) (1993).
- Fischer G. (1994a) *Domain Oriented Design Environments*, Automated Software Engineering, pp 177 – 203.
- Fischer G. (1994b) *Turning Breakdown into Opportunities for Creativity*, Knowledge Based Systems, volume 7, number 4, pp 221 – 232.
- Fischer G., Lemke A., McCall R. and Morch A. (1991) *Making Argumentation Serve Design*, Human Computer Interaction Journal, volume 6, numbers 3 and 4, pp 393 – 419.
- Fischer G., McCall R., Ostwald J., Reeves B. and Shipman F. (1996) *Seeding, Evolutionary Growth and Reseeding: the Incremental Development of Collaborative Design Environments*, in Olson G., Malone T. and Smith. J. (eds.) (1996).
- Fisher R. (1936) *Statistical Methods for Research Workers*, London: Penguin.
- Ford K., Glymour C. and Hayes P. (eds.) (1995) *Android Epistemology*, Cambridge, MA: MIT Press.
- Fry T. (1995) *A Geography of Power: Design History and Marginality*, in Margolin and Buchanan (eds.) (1995).
- Galegher J., Kraut R. and Egido C. (eds.) (1990) *Intellectual Teamwork: Social and Technological Foundations of Co-operative Work*, Hillsdale, N.J: Lawrence Erlbaum Associates Inc.
- Gamma E. (1995) *Design Patterns*, Wokingham: Addison-Wesley.
- Gero J. and Maher M. (eds.) (1993) *Modelling Creativity and Knowledge- Based Creative Design*, Hillsdale, NJ: Lawrence Erlbaum Associates Inc.
- Gero J. and Sudweeks F. (1994) *AI in Design*, Dordrecht: Kluwer Academic.
- Gero J. and Sudweeks F. (1996) *AI in Design*, Dordrecht: Kluwer Academic.
- Gero J. and Sudweeks F. (eds.) (1996) , *Artificial Intelligence in Design 96*, New York, NY: Kluwer Academic Publishers (1996).
- Ghose R. (1995) *Design, Development, Culture, and Cultural Legacies in Asia*, in Margolin and Buchanan (eds.) 1995.

- Gilleard J. and Lee Y. (1993) *Building Services Design in a Hypertext Environment*, in Pohl J. (ed.) (1993).
- Gilleard J. and Lee Y. (1994) *Hypermedia: an Aid for Sprinkler Layout and Installation Design*, *The International Journal of Construction Information Technology*, volume 2, number 4 (1994) pp 81 – 95.
- Gilleard J. and Lee Y. (1996) *Hypermedia and Engineering Design*, *The New Review of Hypermedia and Multimedia*, volume 2, pp 43 – 57.
- Gloag J. (1942) *What About Business?*, London: Penguin.
- Gloag J. (1947) *The English Tradition in Design*, London: Penguin.
- Goel V. and Pirolli P. (1989) *Motivating the Notion of Generic Design within Information-Processing theory: The Design Problem space*, *AI Magazine*, number 10, pp 19 – 36.
- Goel V. and Pirolli P. (1992) *The Structure of Design Problem Spaces*, *Cognitive Science*, number 16, pp 395 – 429.
- Goldberg A. and Rubin K. (1995) *Succeeding with Objects: Decision Frameworks for Project Management*, Reading, MA: Addison-Wesley.
- Goldschmidt G. (1995) *The Designer as a Team of One*, *Design Studies*, volume 16, number 2, pp 189 – 209.
- Goldstine H. (1972) *The Computer from Pascal to von Neumann*, Princeton, NJ: Princeton University Press.
- Gordon W. (1961) *Synectics: the Development of Creative Capacity*, New York, NY: Harper and Row.
- Green M. (1992a) *Conceptions and Misconceptions of Knowledge-Aided Design*, in Green M. (ed.) (1992).
- Green M. (ed.) (1992) *Knowledge-Aided Design*, London: Academic Press.
- Grief I. (ed.) (1988) *Computer-Supported Co-operative Work: a Book of Readings*, San Mateo, CA: Morgan Kaufmann Publishers, Inc.
- Hall D. (1984) *A History of Engineering in Classical and Medieval Times*, La Salle, IL: Open Court.
- Harel D. (1987) *Algorithmics: the Spirit of Computing*, Wokingham: Addison-Wesley Publishing.
- Harris R. (1996) *Variation Among Style Checkers in Sentence Measurement*, *Text Technology*, volume 6, number 2, pp 80 – 90.
- Hars A. (1996) *3-D Meets Manufacturing: The Latest 3-D CAD Systems Improve Manufacturing and Design Processes*, *Byte*, December, 1996, pp 40IS 13 – 40IS 18.
- Haugeland J. (1985) *Artificial Intelligence: the Very Idea*, Cambridge, MA: MIT Press.
- Hayes-Roth F., Waterman A. and Lenat D. (eds.) (1983) *Building Expert Systems*, London: Addison-Wesley.
- Heath T. (1984) *Method in Architecture*, Chichester: John Wiley and Sons.
- Heath T. (1993) *Social Aspects of Creativity and their Impact on Creativity Modelling*, in Gero and Maher (eds.) (1993).
- Heims S. (1992) *The Cybernetics Group: the making of a Post-war Social Science*, Cambridge, MA: MIT Press.
- Hertz K. (1992) *A Coherent Description of the Process of Design*, *Design Studies*, volume 13, number 4, October, 1992, pp 393 – 411.
- Hirano T. (1995) *The Development of Modern Japanese Design: a Personal Account*, in Margolin V. and Buchanan R. (eds.) (1995).
- Hodges H. (1992) *Technology in the Ancient World*, New York, NY: Barnes and Nobel.
- Hofland P. and Utsler J. (1997) *Data Mining at Your Desk*, *Byte*, volume 22, number 7, July 1997, pp 32IS 7 – 10.
- Hollins G. and Hollins W. (1991) *Total Design: Managing the Design Process in the Service Sector*, London: Pitman.

- Hollins W. (1989) *Product Status and the Management of Product Design*, PhD Thesis, University of Strathclyde.
- Hollins W. J. and Pugh S. (1990) *Successful Product Design—What to do and When*, London: Butterworths.
- Jakle J. and Sculle K. (1994) *The Gas Station in America*, Baltimore: Johns Hopkins University Press.
- Jefferies R., Turner A., Polson P. and Atwood M. (1981) *The Processes Involved in Designing Software*, in Anderson J. (ed.) (1981).
- Jones J. (1963) *A Method of Systematic Design*, in Jones J. and Thornley D. (eds.) (1963).
- Jones J. (1969) *The State of the Art in Design Methods*, in Broadbent G. and Ward A. (eds.) (1969).
- Jones J. (1970) *Design Methods: Seeds of Human Futures*, New York: Van Nostrand Reinhold.
- Jones J. (1977) *How My Thoughts about Design Methods have Changed During the Years*, *Design Methods and Theories*, volume 11, number 1, pp 50 – 62.
- Jones J. (1984) *Essays in Design*, Chichester: John Wiley and Sons.
- Jones J. (1988) *Softtechnica*, in Thackera J. (ed.) (1988) *Design After Modernism*, London: Thames and Hudson.
- Jones J. (1991) *Designing Designing*, London: Architecture Design and Technology Press.
- Jones J. (1992) *Design Methods: Seeds of Human Futures* (second edition), New York: Van Nostrand Reinhold.
- Jones J. and Thornley D. (eds.) (1963) *Conference on Design Methods*, Oxford: Pergamon Press Ltd.
- Karatani, K. (1995) *Architecture as Metaphor: Language, Number, Money*, Cambridge, MA: MIT Press.
- Kuhn T. (1962) *The Structure of Scientific Revolutions*, Chicago, IL: University of Chicago Press.
- Kunz W. and Rittel H. (1970) *Issues as Elements of Information Systems*, Working Paper 131, Berkeley, CA: Centre for Planning and Development Research, University of California.
- Landauer T. (1995) *The Trouble with Computers: Usefulness, Usability and Productivity*, Cambridge, MA: MIT Press.
- Landels J. (1978) *Engineering in the Ancient World*, Berkley, CA: University of California Press.
- Larkin J. and Simon H. (1987) *Why a Diagram is (Sometimes) Worth Ten Thousand Words*, *Cognitive Science*, number 11.
- Latour B. (1996) *Aramis or the Love of Technology*, Harvard: Harvard University Press.
- Laurel B. (1990a) *The Computer as Theatre*, New York, NY: Academic Press.
- Laurel B. (ed.) (1990b) *The Art of Human-Computer Interface Design*, Reading MA: Addison-Wesley Publishing Company Inc.
- Laurillard D. (1987) *Computers and the Emancipation of Students: Giving Control to the Learner*, *Instructional Science*, number 16, pp 3 – 18.
- Lawson B. (1980) *How Designers Think: the Design Process Demystified*, Oxford: Butterworth Architecture.
- Lawson B. (1990) *How Designers Think: the Design Process Demystified (Second Edition)*, Oxford: Butterworth Architecture.
- Lawson B. (1993) *Parallel Lines of Thought*, in Edmonds E. (1993) (no page numbers are given in these pre-prints).
- Lowe R. (1993) *Diagrammatic Representation: Techniques for Exploring its Mental Representations and Processing*, *Information Design Journal*, volume 7, number 1.

- Luckman J. (1967) *An Approach to the Management of Design*, Operational Research Quarterly, volume 18, number 4, pp 345 – 358.
- Macrae N. (1992) *John von Neumann*, New York, NY: Pantheon Books.
- Maguire P. (1991) *Designs on Reconstruction: British Business, Market Structures and the Role of Design in Post-War Recovery*, Journal of Design History, volume 4, number 1, pp 15 – 30.
- Margolin V. (ed.) (1989) *Design Discourse: History | Theory | Criticism*, Chicago and London: The University of Chicago Press.
- Margolin V. (1989) *Post-war Design Literature: a Preliminary Mapping*. In Margolin V. (ed.) (1989).
- Margolin V. and Buchanan R. (eds.) (1995) *The Idea of Design: a Design Issues Reader*, Cambridge, MA: MIT Press.
- Martin C. (1988) *User-Centred Requirements Analysis*, London: Prentice-Hall.
- Miles M and Huberman A. (1994) *Qualitative Data Analysis (Second Edition)*, Thousand Oaks, CA: Sage Publications.
- Miller R. (1956) *The Magic Number Seven Plus or Minus Two: Some Limits on Our Capacity for Processing Information*, volume 63, pp 81 – 97.
- Minsky M. (1986) *Society of Mind*, New York, NY: Simon and Shuster.
- Mitcham C. (1994) *Thinking through Technology: the Path between Engineering and Psychology*, Chicago, IS: University of Chicago Press.
- Mitchell W. (1995) *City of Bits: Space, Place and the Infobahn*, Cambridge, MA: MIT Press.
- Moravec H. (1988) *Mind Children: The Future of Robot and Human Intelligence*, Harvard, MA: Harvard University Press.
- Morley D. and Robins K. (1992) *Techno-Orientalism*, New Formations, number 16, pp 136 – 156.
- Newell A. and Simon H. (1972) *Human Problem Solving*, Englewood Cliffs: Prentice-Hall.
- Norberg A and O'Neil J. (1995) *Transforming Computer Technology: Information Processing for the Pentagon*, Baltimore, MR: Johns Hopkins University Press.
- Onians J. (1991) *Idea and Product: Potter and Philosopher in Classical Athens*, Journal of Design History, volume 4, number 2, pp 65 – 73.
- Open University, Arts Foundation Course Team (1974) *William Morris and industrialisation* (an Arts foundation Course, Programme 34), Milton Keynes: Open University.
- Owen C. (1991) *Design Education in the Information Age*, Design Issues, volume VII, number 2, pp 25 – 33.
- Pacey P. (1992) 'Anyone Designing Anything?' *Non Professional Designers and the History of Design*, Journal of Design History, volume 5, number 3, pp 217 – 225.
- Page B. (1997) *Promises Unfulfilled*, Byte, volume 22, number 7, July 1997. pp 32IS 11 – 14.
- Page J. (1963) *A Review of the Papers Presented at the Conference on Design Methods*, in Jones and Thornley (1963).
- Petroski H. (1992) *To Engineer is Human: The Role of Failure in Design*, New York, NY: Vintage Books.
- Petroski H. (1995) *Soft Graphics*, American Scientist, volume 83, number 1.
- Petroski H. (1996) *Invention by Design: How Engineers Get from Thought to Thing*, Cambridge MA: Harvard University Press.
- Peters T. (1993) *Liberation Management: Necessary Disorganisation for the Nanosecond Nineties*, London: Pan.
- Pham D. (ed.) (1988) *Expert Systems in Engineering*, London: IFS.
- Potter N. (1980) *What is a Designer: Things, Places, Messages*, Reading: Hyphen.
- Pugh S. (1989) *Knowledge-based Systems in the Design Activity*, Design Studies, volume 10, number, 4, pp 219 – 227.

- Pugh S. (1991) *Total Design: Integrated Methods for Successful Product Engineering*, Wokingham: Addison-Wesley Publishing Company.
- Purcell P. (1988) *Computer Environments for Design and Designers*, Design Studies, volume 9, number 3.
- Putnam H. (1975) *Mind, Language and Reality*, Cambridge: Cambridge University Press.
- Pye D. (1964) *The Nature of Design*, London: Studio Vista.
- Pye D. (1978) *The Nature and Aesthetics of Design*, New York, NY: Van Nostrand Reinhold.
- Rittel H. and Webber M. (1973) *Planning Problems are Wicked Problems*, Policy Studies, number 4, pp 155 – 169.
- Rittel H., Grant D. P. and Protzen J. (1972) *Second Generation Design Methods* (Rittel interviewed by Grant and Protzen), The DMG 5th Anniversary Report: DMG Occasional Paper No 1, pp 5 – 10.
- Rodgers P. (1995) *Product Performance Assessment*, PhD Thesis, London: University of Westminster.
- Rodgers P. and Huxor A. (1998) *The Role of Artificial Intelligence as 'Text' within Design*, Design Studies, vol 19, number 2, April, pp 14 – 161.
- Rolt L. (1974) *Victorian Engineering*, Harmondsworth, Middlesex: Penguin Books.
- Rothenberg A. and Hausmann C. (1976) *The Creativity Question*, Duke University Press.
- Rudd J., Stern K. and Isensee S. (1996) *Low vs. High fidelity Prototyping Debate*, Interactions: January 1996, pp 75—86.
- Rychener M. (ed.) (1988) *Expert Systems for Engineering Design*, New York, NY: Academic Press.
- Ryle G. (1949) *The Concept of Mind*, London: Hutchinson.
- Sacerdoti E. (1975) *The Non-Linear Nature of Plans*, in the Proceedings of the International Joint Council of Artificial Intelligence 1975, pp 206 – 214. Palo Alto, CA: Morgan Kauffman.
- Scaife M., Curtis E. and Hill C. (1994) *Inter-disciplinary Collaboration: A Case Study of Software Development for Fashion Designers*, Cognitive Science Research Paper number 334, University of Sussex.
- Schrage M. (1996) *Cultures of Prototyping*, in Winograd T. (ed.) (1996).
- Silverman B. (1992) *Critiquing Human Error*, London: Academic Press.
- Simon H. (1969) *The Sciences of the Artificial*, Cambridge MA: MIT Press.
- Sriram D., Stephanopoulos G., Logcher R., Gossard D., Groleau N., Serrano D. and Navinchandra D. (1989) *Knowledge-based System Applications in Engineering Design: Research at MIT*, AI Magazine, p 79 (Fall 1989).
- Snodgrass A. and Coyne R. (1992) *Models, Metaphors and the Hermeneutic of Designing*, Design Issues, volume IX, number 1, pp 56 – 74.
- Stier D. (1992) *Creating a Scientific Community at the Interface Between Engineering Design and AI: a Workshop Report*, in Tong and Sriram (eds.) (1992) volume III.
- Stomph-Blessing L. (1992) *Engineering Design and Artificial Intelligence: a Promising Marriage?* In Cross et al (eds.) (1992) pp 177 – 212.
- Sturt G. (1923) *The Wheelwright's Shop*, London: Cambridge University Press.
- Suchman L. (1987) *Plans and Situated Actions*, Cambridge: Cambridge University Press.
- Suchman L. (1994) *Do Categories have Politics? The Language/Action Perspective Reconsidered*, Computer Supported Cooperative Work, volume 2, number 3, pp 177 – 190.
- Suchman L. (1995a) *Making Work Visible*, Communications of the ACM, volume 38, number 5, pp 57 – 65.
- Suchman L. (ed.) (1995b) *Special Issue of the Communications of the ACM: Making Work Visible*, Communications of the ACM, volume 38, number 5.

- Swales J. (1991) *Genre Analysis*, Cambridge: Cambridge University Press.
- Tjosvold D. (1991) *Team Organisation: an Enduring Competitive Advantage*, Chichester: John Wiley and Sons,
- Tong C. and Sriram D. (eds.) (1992) *Artificial Intelligence in Engineering Design (volume I)*, Boston, MA: Academic Press Inc.
- Tong C. and Sriram D. (eds.) (1992) *Artificial Intelligence in Engineering Design (volume II)*, Boston, MA: Academic Press Inc.
- Tong C. and Sriram D. (eds.) (1992) *Artificial Intelligence in Engineering Design (volume III)*, Boston, MA: Academic Press Inc.
- Towers G. (1995) *Building Democracy: Community Architecture in the Inner Cities*, London: UCL Press.
- Tunncliffe A. and Scrivner S. (1991) *Knowledge Elicitation in Design*, Design Studies, volume 12, number 2.
- Tufte E. (1983) *The Visual Design of Quantitative Information*, Cheshire, CN: Graphics Press.
- Tufte E. (1992) *Envisioning Information*, Cheshire, CN: Graphics Press.
- Tufte E. (1997) *Visual Explanations: Images and Quantities, Evidence and Narrative*, Cheshire, CN: Graphics Press.
- Valkenburg R. and Dorst K. (1998) *The Reflective Practice of Design Teams*, Design Studies, vol 19, number 3, July 1998, pp 249 – 271.
- von Neumann J. (1958) *The Computer and the Brain*, London: Yale University Press.
- Whitehead A. N. (1925) *Science and the Modern World*, New York, NY: Free Press.
- Whitney D. (1990) *Designing the Design Process*, Research in Engineering Design, number 2, pp 3 – 13.
- Wickens C. (1992) *Engineering Psychology and Human Performance*, New York, NY: Harper Collins.
- Winograd T. and Flores A. (1987) *Understanding Computers and Cognition: a New Foundation for Design*, Reading, MA: Addison-Wesley.
- Winstanley G. (ed.) (1990) *Artificial Intelligence in Engineering*, Chichester: Wiley.
- Winston P. and Pendergrass K. (eds.) (1984) *The AI Business: Commercial Uses of Artificial Intelligence*, Cambridge, MA: MIT Press.
- Wooley T. (1995) *Architectural Education and Participatory Design—the Use of Live Projects to Break Down Barriers*, in Pemberton, Downie and Peng (eds.) (1995).
- Zullighoven H. (ed.) (1992) *Special Issue: Practical Experiences in Prototyping, Information Technology and People*, volume 6, numbers 2 and 3.