University of Plymouth

PEARL **https://pearl.plymouth.ac.uk**

04 University of Plymouth Research Theses 01 Research Theses Main Collection

1990

COGNITIVE PROCESSES IN ENGINEERING DESIGN

Ball, Linden John

http://hdl.handle.net/10026.1/674

http://dx.doi.org/10.24382/4475
University of Plymouth

All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.

COGNITIVE PROCESSES IN ENGINEERING DESIGN

LINDEN JOHN BALL

A thesis submitted in partial fulfilment of the
requirements of the Council for National Academic Awards
for the degree of Doctor of Philosophy

August 1990

Department of Psychology, Polytechnic South West
in collaboration with the Department of Computer Science,
University of Reading and Plessey Semiconductors, Roborough.

ACKNOWLEDGEMENTS

I am indebted to my Director of Studies, Professor J.St.B.T Evans for his expert supervision and technical assistance throughout the undertaking of this research as well as the continuous encouragement and friendship he has always provided. His unflagging patience during the write-up of the thesis should also not go without mention. I would additionally like to express my many thanks to Ian Dennis for the considerable guidance, advice and help that he provided in his role as my second supervisor.

Thanks are due also to all other members of the PEDA project team, including Professor Keith Baker, Phil Culverhouse, Pat Pearce, Peter Jagodzinski, Dean Scothern and the late Gill Venner. The continued advice and stimulation of these colleagues has proved invaluable and the dedicated help of Phil and Dean with, amongst other things, the development of design problems and the assessment of subjects' work, has been especially appreciated. I am thankful, too, for the assistance of Kenn Lamb (formerly of Plessey Semiconductors) who helped with the recruitment of professional subjects and who provided useful advice and ideas.

I would also like to acknowledge the financial support given by the National Advisory Body. Without the research grant awarded to the larger multi-disciplinary project the present programme of psychological research would not have been possible.

Finally, my thanks go out to Penny and Liz.

DECLARATIONS

(1) While registered for this degree, I have not been a registered candidate for another award of the CNAA or of a University.

(2) The present research project was funded by a National Advisory Body grant originally awarded to Professor K.D. Baker and Dr G. Sullivan in 1986. Upon their departure to Reading University at the end of 1986, however, the grant was taken over by Mr P.F. Culverhouse, Dr P.D. Pearce and Professor J.ST.B.T. Evans. The funding awarded to the present researcher extended over a period from the 1st September 1986 to the 30th December 1989.

(3) It should be noted that the present research project was one of three associated projects that was funded under the aforementioned N.A.B. grant. Whilst, however, the three projects were motivated by the common desire to develop a software system to facilitate engineering design processes, each researcher's work was undertaken as a distinct and separable programme of research. The author of the present thesis, then, was concerned with investigating the nature of cognitive processes in engineering design so as to inform the development of the intended design aid. The other two researchers, however, were concerned with the actual development of respectively (a) the underlying functionality of the design aid and (b) the interface to the system.

(4) A course of advanced study has been completed in partial fulfilment of the requirements for the degree consisting of (a) attendance at selected lecture and seminar series run under the M.Sc. Intelligent Systems course at Polytechnic South West and (b) attendance at a number of relevant professional conferences and workshops.

ABSTRACT

Cognitive Processes in Engineering Design

by Linden John Ball

The central aim of the current research programme was to gain an understanding of the cognitive processes involved in engineering design. Since little previous empirical research has investigated this domain, two major exploratory studies were undertaken here. Study One monitored seven final-year students tackling extended design projects. Diary and interview data were used to construct detailed design behaviour graphs that decomposed activities into structured representations reflecting the goals and subgoals that were pursued. Study Two involved individual observation (using video) of six professional engineers "thinking-aloud" as they tackled a small-scale design problem in a laboratory setting. A taxonomic scheme was developed to classify all verbal protocol units and other observable behaviours.

In interpreting the data extensive use was made of theoretical concepts (e.g. schemas and mental models) deriving from current research on human problem solving and thinking. Evidence indicated that the engineers studied had many similar methods of working which could be described at a high level of abstraction in terms of a common "design schema". A central aspect of this schema was a problem reduction strategy which was used to break down complex design problems into more manageable subproblems. The data additionally revealed certain differences in design strategy between engineers' solution modelling activities and also showed up tendencies toward error and suboptimal performance. In this latter respect a particularly common tendency was for designers to "satisfice", that is to focus exclusively on initial solution concepts rather than comparing alternatives with the aim of optimising choices.

The general implications of the present findings are discussed in relation to both the training of design skills and the development of intelligent computer systems to aid or automate the design process. A final, smaller scale of experimental study is also reported which investigated the possibility of improving design processes via subtle interventions aimed at imposing greater structure on design behaviours.

CONTENTS

Page

VOLUME I

Chapter 1. General Introduction..... 2

Chapter 2. Review of Research Literature..... 15

Chapter 3. Investigating Engineering Design Processes: Some Theoretical and Methodological Considerations..... 69

Chapter 4. Study One: Undergraduate Design Projects..... 88

Chapter 5. Study Two: Professional Engineers..... 192

Chapter 6. Final Discussion..... 290

References..... 342

VOLUME II

Appendix A. Undergraduate Design Study..... 1

Appendix B. Professional Design Study..... 120

Appendix C. Intervention Study..... 258

Chapter 1

General Introduction

Chapter 1. General Introduction

Page

1.1. THE PRESENT PSYCHOLOGICAL RESEARCH IN CONTEXT

1.1.1. Motivation behind the present research..... 4

1.1.2. Aims of the present research..... 8

1.2. SOME PRELIMINARY DEFINITIONS

1.2.1. Engineering design: characterising the domain..... 9

1.2.2. The nature of engineering design tasks..... 11

1.1. THE PRESENT PSYCHOLOGICAL RESEARCH IN CONTEXT

1.1.1. MOTIVATION BEHIND THE PRESENT RESEARCH

The programme of psychological research detailed within the present thesis was undertaken as part of a collaborative project involving contributions from electronic engineers, computer scientists and cognitive psychologists with the overriding aim of developing a prototype software environment for the facilitation of engineering design. Whilst the main focus of this thesis is on the psychological component of the project as undertaken by the present author, it seems sensible to start off with some background details concerning the larger project as a whole. This background information should serve to (1) indicate the motivation behind the psychological contribution to the larger project (2) show how the psychological work can be viewed within its broader applied context and (3) pave the way toward a description of the specific aims of the psychological research.

From the outset of the project it was decided that the intended design aid - which has come to be called the Plymouth Engineer's Design Assistant (PEDA) - should be rather different in form and function to many existing engineering CAD systems. If one examines the nature of contemporary design aids for engineers (see, Culverhouse, 1986, for a review) they appear to be of two main types. Firstly there are the conventional "toolkit" variety which appear as little more than sophisticated draughting aids for assisting with low-level circuit design and simulation. These systems (e.g. Mentor Graphics Idea 1000) possess little in the way of any embedded intelligence and therefore

exist as passive design aids which are blindly obedient to the engineer's wishes. Secondly there are the "expert system" variety of design aids which are aimed at automating intelligent design processes and require only that the user, when requested, provide factual details about the particular task at hand (e.g. the Design Automation Assistant of Kowalski and Thomas, 1983). Currently, design automation tends be the dominant approach to the application of intelligent systems in engineering, though most work still exists at a research level.

The view held by the current research team was that engineering CAD systems falling into either the toolkit or the expert system categories tend to be limited in three major respects. Firstly, many such systems tend only to be useful in extremely narrow subdomains of engineering such as bridge design or VLSI chip design rather than being more broadly applicable to a multiplicity of engineering subdomains. Secondly, most systems tend to be focussed on the lower levels and later stages of the design process such as design validation or design optimisation and thereby fail to support the higher levels and earlier phases of the process. Thirdly, most systems generally fail to fully exploit the differing strengths of human expert and computer and are thereby unlikely to facilitate the rich designs that could result from a cooperative and jointly intelligent interaction. It appears that other researchers have noted these limitations as well for a few systems are currently being developed which are aimed at cooperating intelligently with the designer as well as assisting with multiple phases of design projects. Examples of these systems include the Carnegie-Mellon University Design Assistant (see report by Director, Shen, Siewiorek & Thomas, 1982) and the "Design-to-Product" software being developed at

Edinburgh University (see reports by Smithers, 1985, and Popplestone et al., 1986). However, whilst these latter systems are felt to be a positive step toward bridging the gap between designer's toolkits and automated design systems there is a tendency for these cooperative systems to be limited in a further crucial respect. That is to say, these systems generally do not appear to be based upon any explicit psychological model of the processes that are involved in engineering design. Instead they tend to be founded upon the intuitions and speculations of their creators - who are usually engineers - about how people design and where they are deficient. Basing design systems on intuitive considerations about the nature of design processes is particularly worrying in the light of psychological evidence which suggests that people - experts included - have very poor self-knowledge of their own cognitive processes yet often believe that they know the strategies and procedures that they use (see chapter 3 for a discussion of the issue of self-knowledge in thinking). The point is, how can you truly hope to facilitate engineers' design work without an accurate understanding of the psychological processes involved and the inherent limitations of these processes? CAD systems derived from potentially erroneous, speculation-based models of design processes could conceivably hinder rather than assist designers by, for example enforcing adherence to unnatural or overly rigid sequences of activities.

In stark contrast to existing engineering design aids, then, it was decided that the PEDA should be soundly based upon a *psychological* understanding of the cognitive processes that are involved in engineering design. It was felt that such a psychological basis could

enable the development of a genuinely cooperative system capable of working intelligently with the engineer, facilitating his or her thought processes and helping optimise the design produced. In addition to the psychological foundation of the PEDA it was felt that there were two further respects in which the system could address the limitations of existing engineering design aids. Firstly, it was though desirable that the PEDA should take "electronics design" in general as its broad subdomain of application and that it should furthermore be readily extendable to other engineering subdomains (e.g. mechanical engineering). Secondly, it was felt that the PEDA should be aimed at supporting not only the later phases and lower levels of engineering design but also the *earlier* stages and *higher* levels of the process where major structuring and planning decisions are likely to be made and where relatively abstract design concepts and ideas are likely to be generated and evaluated.

In view of the importance of designing the PEDA system to take account of the psychological characteristics of engineering design, the present programme of psychological research was initiated. Whilst, the research was certainly an integral aspect of the whole attempt to develop the PEDA, it is important to realise that it was undertaken in a way that made it a distinct and separable programme of work to the computing and engineering components of the larger project. Additionally, it should be borne in mind that although the psychological research was clearly motivated in the first instance by the specific applied objective of informing the design of the PEDA software, this clearly does not detract from the much broader contributions to knowledge that it makes. For example, from a more overarching applied perspective, anyone concerned

with the the development of design support environments would be likely to find aspects of this work informative. Likewise, from a purely theoretical stance, anyone interested in the nature of the thought processes involved in "real world" domains of human activity might gain some relevant insights from this research. Indeed it is the more global applied and theoretical aspects of the present research which this thesis deals with and it is only in the final chapter that the discussion turns more toward some of the specific implications that this work has had for the development of the PEDA.

1.1.2. AIMS OF THE PRESENT RESEARCH

The primary aim of the present research was to derive a psychological understanding of the engineering design process, identifying (1) any common strategy or schema for engineering design (2) any individual differences in methods of working and (3) any characteristic biases or cognitive limitations constraining the attainment of optimal designs. The attainment of this theoretical understanding was seen as critically important for advising the design of the PEDA so that the system would be able to allow the user to design in a natural and flexible manner whilst at the same time helping to counteract any design errors arising, for example, from biased judgement or limitations of working memory capacity.

To help counteract errors, inconsistencies and deadlocks arising during the design process it was envisaged that the PEDA should be capable of lending constructive assistance to the designer in the form of reminders, advice, guidance or help with problem structuring. Indeed the possibility of optimising the design process by means of fairly

unobtrusive and subtle modes of intervention seemed an appealing avenue for psychological investigation. Would, for example, positive affects on design processes arise by periodically requiring designer's to be explicit about their current goals or recent decisions? The psychological investigation of the influence of such intervention strategies formed an important secondary aim of this present research programme.

Having outlined the basic aims of the psychological research detailed within this thesis it next seems appropriate to move on to consider what the domain termed engineering design is fundamentally concerned with as well as to determine how engineering design tasks may best be characterised from a psychological perspective.

2. SOME PRELIMINARY DEFINITIONS

1.2.1. ENGINEERING DESIGN: CHARACTERISING THE DOMAIN

A fairly comprehensive characterisation of engineering design as a domain of human activity has been provided by Fielden (1963). He proposed that it is "the use of scientific principles, technical information and imagination in the definition of a mechanical structure, machine or system to perform prespecified functions with the maximum economy and efficiency". This description seems to pave the way toward a realisation of the essential feature that differentiates engineering design from other more artistic forms of design related activity (e.g. craftwork or musical composition). This defining feature can be described as the extent to which *technological* information must be

made use of by its practitioners (cf. Asimow, 1962). It seems true to say, however, that most forms of design-oriented activity involve the application of some degree of technical information, and therefore it is perhaps justifiable to think of design domains as extending along a continuum of technological involvement as depicted in figure 1.1. Engineering design, then, would seem to fall at one end of this continuum. For the engineer, the quantity of the technological information that must be manipulated is often vast even in a single area of application, and is ever increasing as science progresses. Moreover the complexity of these technological factors is generally considerable, as is, too, the nature of their underlying physical phenomena - some understanding of which is usually needed by the engineer.

```
     quantity and complexity of technical information --->
 craft   creative   musical     software  architectural  engineering
 work    writing    composition  design     design         design
 |_______|__________|___________|__________|______________|____________|
```

Figure 1.1: A continuum of design domains.

The plausibility of a characterisation of design domains along a continuum of technological involvement, although speculative, is heartening for the psychologist interested in understanding engineering design processes since it implies that there may be certain commonalities between engineering design and other forms of design-oriented endeavour. Thus architectural design and software design, which both revolve around sophisticated technologies, may have a close psychological resemblance to engineering design. This suggestion will be pursued to a greater extent in chapter 2 where discussion of empirically

based research into engineering design frequently broadens to include research on architectural and software design.

1.2.2. THE NATURE OF ENGINEERING DESIGN TASKS

A common opinion concerning the nature of design tasks is that they are "problems". The legitimacy of such a view - expressed by practitioners and psychologists alike - can easily be made evident if the concept of a problem is briefly examined. In psychological terms, then, a person is confronted with a problem when it is necessary for them to reach some goal which they are not immediately able to achieve (see, for example, Newell & Simon, 1972). Thus, having to solve simultaneous equations, desiring to beat your opponent at chess, wanting to fix your car when it fails to start, being asked to figure out a crossword puzzle, or having to improve the country's economy are all examples of problem situations that may confront a person or a group of individuals. These problems clearly cover a range of complexity, yet they all have certain general features in common. They all have an "initial state", whether it be a set of equations or the present condition of the economy, and they all have a "goal state", such as a car that starts or an completed crossword puzzle. Additionally, in actually solving the problem, the person selects and applies "operators" for example, movement of chess pieces or adjustment of tax levels. Frequently, too, there are restrictions - or what are commonly referred to as "constraints" - on the operators that may be applied. Some constraints may be specific such as the legal movement characteristics of individual chess pieces, whilst others may be more global such as the the time you have set yourself for completing the puzzle.

Clearly, then, the legitimacy of referring to design tasks as problems is readily apparent since such tasks involve goal states which are only rarely immediately attainable. Generally the goal state of a design task concerns the formulation of a description or model (i.e. a "design") of a desired artifact which meets certain specified functional and structural requirements. Quite often, however, the desired goal state is actually more than *just* the design of an artifact in that an actual "implementation" of an working prototype is also sought. This is likely to be so in many engineering domains where it is practicable to produce such prototypes, though in domains such as architecture and civil engineering it is almost always the case that the prototype and the final structure - building, bridge or whatever - are one and the same. Continuing, then, with the characterisation of design tasks as problems, it is clear that in attempting to attain the goal state (unless it is a truly trivial one) the designer must utilise an extensive number of operators - particularly mathematical and graphical ones - defined in terms of the domain itself. In addition, the designer must select operators with due awareness of an abundance of constraints. Such constraints not only place restrictions on the possible options available for meeting functional requirements but also place restrictions on the design process itself, by, for example, limiting the time that the designer has available to actually attain the final design solution.

Since design tasks seem to fall so readily within the category of "problems" it is hardly surprising that most researchers who have engaged in empirical work on the psychological aspects of design have approached the domain with a panoply of theoretical notions derived from

general problem-solving research. Such notions are introduced in the following review chapter which presents (a) a brief though comprehensive overview of general problem-solving research and theory, as well as (b) a more detailed synopsis of psychological research that has been undertaken on design. Before moving on to this chapter, however, it is worth bearing in mind that most of the researchers who have attempted to pursue an understanding of the cognitive processes involved in design rarely fail to comment that design is both a highly complex domain of human skill and one that as yet is very poorly understood.

PAGE

MISSING

IN

ORIGINAL

Chapter 2

Review of Research Literature

Chapter 2. Review of Research Literature

Page

2.1. INTRODUCTION TO THE REVIEW

2.1.1. Design as problem solving..... 17

2.1.2. Focus of the review..... 18

2.2. PROBLEM-SOLVING: RESEARCH AND THEORIES

2.2.1. Well-structured and ill-structured problems..... 22

2.2.2. The information-processing approach in problem-solving research..... 23

2.2.3. Early information-processing theories of problem solving. 27

2.2.4. The role of analogy in problem solving..... 32

2.2.5. Domain problem solving and the nature of expertise..... 35

2.2.6. Interim summary and concluding comments..... 42

2.3. PROBLEM SOLVING IN DESIGN

2.3.1. Early applications of problem-solving theory to design... 45

2.3.2. Recent studies of design..... 55

2.3.3. Mental models in design..... 61

2.4. CONCLUSION TO THE LITERATURE REVIEW

2.4.1. Cognitive processes in design: a summary and synthesis... 64

2.1. INTRODUCTION TO THE REVIEW

2.1.1. DESIGN AS PROBLEM SOLVING

In the previous chapter the commonly held view that design tasks are "problems" was briefly examined and found to be a legitimate one. It comes as little surprise, then, that the approach adopted by most psychologists interested in the nature of design processes has involved the application of both (a) general theoretical notions arising from research on problem-directed thinking and (b) investigative techniques such as protocol analysis which have commonly been used to understand such thinking. There is certainly a great deal of sense in such an approach since it seems likely that theoretical concepts and research methods which have proved their worth in already quite well investigated domains of problem solving (e.g. chess and physics) will also be useful for understanding the processes involved in domains which have not yet become well-established areas of psychological interest (e.g. engineering design).

Whether or not such an approach can indeed prove fruitful in novel areas of investigation is, of course, something which can only be determined through empirical investigation. Certainly the applicability of established methodologies and concepts should not be blindly assumed *a priori* - though if an attempted application fails, it clearly serves to indicate the limitations of existing theoretical perspectives. More often than not, however, it transpires that the relevance of existing theoretical ideas is justified. For example, one dominant theoretical notion to emerge from studies of problem solving (cf. Simon 1978a) is

that the main stream of problem-directed thought proceeds in a serial manner (i.e. only one mental operation is carried out at a time). One might expect, then, that engineers tackling "design problems" would similarly display such seriality of thought, and clearly this would be a useful phenomenon to look out for. As will emerge later in this chapter (i.e when empirical studies of design are reviewed) it does indeed appear that designers - like other problem solvers - are only able to explore one line of problem development at a time.

2.1.2. FOCUS OF THE REVIEW

Since existing problem-solving research appears to have formed an important basis for psychologists investigating design processes, it clearly seems worthwhile to engage in a reasonably brief but fairly comprehensive general survey of this research literature. Such a survey is undertaken in the first half of this chapter, that is, prior to later sections of the chapter in which the limited psychological literature that exists on design is reviewed. Whilst general surveys of problem-solving research and theory abound in the literature (see, for example, Kahney, 1987 and Gilhooly, 1988 for two recent extended reviews) the specific aims of the present, more limited, undertaking are clearly somewhat different and less ambitious to those of existing works. Firstly, then, and at a rather prosaic level, the survey presented in the first half of this chapter is intended to serve as a vehicle for the introduction of numerous theoretical concepts that have arisen in problem-solving research. The point is simply that a knowledge of these concepts is essential for understanding later sections of the chapter (and indeed later chapters of the thesis). Secondly, however, and at a

somewhat deeper level, a major aim of the survey is to draw out the evidence for the existence of general cognitive "principles" that appear to characterise the nature of the cognitive processes involved in problem solving. The rationale behind this latter emphasis clearly revolves around (a) the fact that only an extremely limited quantity of psychological research has actually been carried out on design problem solving and (b) the point highlighted above (section 2.1.1) that established principles concerning problem-directed thinking provide a useful starting point in the attempt to understand novel areas of research interest - in the present case engineering design.

In the second half of this chapter an attempt is made actually to review the existing psychological literature that deals with the nature of cognitive processes in "design". Whilst, of course, it would be desirable to focus exclusively on studies of "engineering design", the sheer paucity of systematic research that has been done in this particular area (literally only a couple of investigations) would enable only an very sketchy picture to be drawn of the cognitive processes involved. Of necessity, then, the review will be a fairly diverse one - broadening to include discussion of psychological research and theories relating to architectural and software design. It is appreciated that such a broad review is by no means an ideal way to confront the issue of how "engineers" think. All the same, however, it should at least provide some tentative insights into the processes that may be involved, particularly in light of one basic factor which - albeit at an abstract level - links all design domains together, that is, they all involve *design* as their central concern. Although this may seem a mundane point to make, its triviality may be more apparent than real. For

example, in treating a wide range of tasks as being broadly categorsisable as "well-defined problems", psychologists are adopting the same kind of approach in their belief that quite similar types of cognitive processes may underly problem solving on different tasks. Likewise, then, the suggestion being made here, is that irrespective of the actual content of design problems or the design domain from which they are derived, similar types and sequences of processes *may* be occurring during the phases of problem representation and solution development. Certainly at a broad level of analysis this does indeed appear to be the case, since many authors (e.g. Jones, 1980; David, 1987; Smyth, 1988) present the view that, in general, design activities are characterisable in terms of broad cyclic phases of "analysis" and "synthesis". Whilst the value of such an abstract conceptualisation of design processes is questionable (cf. Malhotra et al., 1980), the example simply serves to illustrate the point that differing domains *can* be seen to involve similar sequences of processing.

It should be noted that whilst existing psychological analyses of design are few in number, there is an abundant professional literature which is available, written by practitioners of design, which claims to shed light on the nature of the processes involved. Much of this literature is speculative in form and tends to involve (a) retrospective accounts by people of their own design experiences (b) discussions of the results of highly informal observational studies of design and (c) presentation of ideas concerning the strategies and techniques that seem to be important for success. Indeed the emphasis of this literature appears to be mainly on the prescription of design methodologies and formal

procedures which should be to adhered to for the production of enhanced design solutions (see Asimow, 1962; Hubka, 1982; Pahl & Beitz, 1985, for example texts specific to the domain of engineering). Whilst, then, this general literature is certainly interesting, it seems to be of little actual value in the context of the present thesis because of its highly speculative nature. That is to say, it does not seem to provide any information on design processes that we can be truly confident about - particularly in light of the evidence (to be discussed in chapter 3) which suggests that people's knowledge of their own cognitive processes is often highly unreliable. Within the present thesis, then, no attempt will be made to review this professional literature on design.

In concluding this introduction it is worth mentioning that throughout the two review sections of the present chapter - which respectively deal with the psychology of problem solving and the psychology of design - coverage of material will be essentially chronological. This presentation strategy should enable advancements that have occurred in theoretical formulations of design processes to be viewed in the light of theoretical progress made in the more general field of problem-solving research. Whilst space considerations have generally precluded the actual presentation of much detail on specific experimental manipulations, performance measures, data analyses and the like, detailed references to all research are, of course, provided.

2.2. PROBLEM SOLVING: RESEARCH AND THEORIES

2.2.1. WELL-STRUCTURED AND ILL-STRUCTURED PROBLEMS

Psychologists interested in the nature of human problem solving have broached the idea that problems (see section 1.2.2 for a definition of the term problem) can be classified in terms of the extent to which they are "well-defined" (or "well-structured") as opposed to being "ill-defined" (or "ill-structured"). Reitman (1964, 1965) appears to have presented the first extensive discussions on this issue, and the theme was soon taken up by other researchers including Newell (1969) and Simon (1972, 1973). Nowadays the majority of psychologists interested in problem solving tend, at some point, to expound upon the distinction between ill-defined and well-defined problems, though a prominent feature of such expositions is their frequent diversity. Whilst, however, a certain degree of fuzziness surrounds this intuitively appealing distinction, it generally seems to be the case that a problem can be classified as ill-defined if either (a) the information specifying the initial state, the goal state, and the permissible actions is not entirely contained in the problem statement and/or (b) the criterion that determines whether the goal has been reached is either complex or inexact and/or (c) there is no simple "legal move generator" for finding all of the alternative possibilities at each step. The basic idea, then, is that problems exhibiting one or more of these preceding deficits may be described as having less definition or structure than problems that exhibit no such deficits.

Looking first at examples of well-defined problems, the most clear-cut

cases appear to be the "transformation" puzzles which have featured so extensively in laboratory research on problem-directed thinking. Such puzzles include the Tower of Hanoi problem (e.g Luger, 1976), the 8-puzzle (e.g. Ericsson, 1975) and the missionaries-and cannibals task (e.g. Reed, Ernst & Banerji, 1974) and involve the subject making moves that transform one problem state into another in order to attain a solution state. It should be noted, however, that well-defined problems are also quite common in domains of specialist knowledge (for example, theorem proving in geometry) as well as in basic everyday life (for example, having to get to a particular destination when you are in an unfamiliar city). Still, though, it seems true to say that most of the more interesting or frustrating problems that face us in our lives tend to be rather *ill-defined* such as striving to achieve happiness or success or having to pass examinations.

A final point which is worth mentioning is that the majority of problems that confront people outside of the psychological laboratory tend to arrive with very few explicit "givens" and thus should really be categorised as ill-defined according to the classification scheme detailed above. Often however, such real-world problems tend to become very rapidly well-defined since the solver immediately augments the problem givens with knowledge retrieved from memory. When a solver's knowledge is taken into account, then, it perhaps becomes evident why clear criteria for differentiating well- from ill-defined problems have proved somewhat difficult to attain.

2.2.2. THE INFORMATION-PROCESSING APPROACH IN PROBLEM-SOLVING RESEARCH

The fundamental research technique which has been employed in

investigating people's problem-solving processes has involved obtaining "think aloud" protocols whilst subjects work on given tasks. Since the following chapter presents a detailed discussion of the many issues surrounding this process tracing technique, the theme will not be pursued any further here. Instead, it seems sensible to focus briefly upon another research method, namely "computer simulation", which has frequently been made use of by investigators in order to *test* their specific theories about human problem-solving processes.

In the simulation approach the researcher writes a program that embodies a model of the cognitive processes believed to be used by subjects in tackling a particular problem. This simulation program can next be fed with a description of this problem and a subsequent step-by-step output trace of the program's performance can be obtained which depicts all decisions made and operations taken. The goodness of fit between computer trace and human protocol gives an indication of the extent to which the process model is valid. Whilst the computer simulation approach to modelling cognitive processes has many adherents (see Rips, 1989, for example, for a recent application in the context of reasoning research) the approach is not without its difficulties (Evans, personal communication). In general, the value of producing a simulation program seems to lie in the fact that it forces the theorist to be clear and unambiguous about the assumptions actually entering into a particular process model, since without such rigorous specification the program is unlikely to run.

The bulk of early problem-solving research based upon the information-processing perspective made considerable use of both protocol analysis techniques and computer simulation methods. Often, too, the focus of

these early investigations tended to be on the problem solving processes involved in tackling the *well-defined* types of "puzzle" tasks mentioned in section 2.2.1 above. Such puzzle problems are frequently spoken of as being ***semantically-impoverished*** (e.g. Chi, Glaser & Rees, 1982) or ***domain-free*** (e.g. Best, 1986), both terms referring to the fact that these tasks do not require the solver to have any specialist background knowledge in order to tackle them. Most people, for example, would be able to attempt, and eventually attain, a solution for the following problem even without any prior experience of such "river crossing" tasks:

> "Three missionaries and three cannibals are on one side of a river. They all have to get to the other side via a canoe that can hold only two persons. At no point can there be more cannibals than missionaries on either bank of the river, or the cannibals will kill the missionaries they outnumber. How do they all get across in the simplest schedule of crossings? It is assumed that all passengers on the boat unboard before the next trip, and at least one person must be on the boat for each crossing".

In problem-solving research, the major advantage in using well-defined, semantically-impoverished puzzles like the previous example appears to be threefold in nature. Firstly, since all subjects will be equally unfamiliar with such tasks, the investigator does not need to worry about different people bringing to bear different amounts of prior domain-knowledge. Clearly, then, any differences that emerge in solution performance are attributable to differences in basic underlying processes. Secondly, since it is easy to "dress-up" the core aspects of such problems in different guises, it is possible to develop overtly different problems which are in fact identical at a structural level. This can prove useful in investigating the affect that knowledge has on problem-solving (i.e how does experience derived with one problem transfer to performance in tackling another problem?). Thirdly, because

of their small-scale, puzzle tasks tend to involve fairly short solution times (usually less than one hour) and so are more amenable to investigation than many types of real-world problem.

Section 2.2.3 below presents a broad overview of the major theoretical ideas concerning problem-directed thinking that arose from early research which made extensive use of well-defined and semantically-impoverished tasks. As will be seen, much of the early research work appears to have been directed toward the pursuit of "general-purpose" problem-solving processes which could be applied to a wide range of different problems. More recent research, however, has shown a clear shift in emphasis away from this early focus on puzzle tasks toward an interest in the processes involved in tackling *semantically-rich* problems (e.g. those occurring in domains such as chess, physics and mathematics) which require the solver to have extensive background knowledge. In section 2.2.5, then, some coverage of this more recent research on domain problem solving and the nature of expertise is provided. As will emerge in this discussion, this recent research has considerably broadened psychologists' understanding of the kinds of knowledge that are essential for efficient problem solving. Much of this latter work has been particularly influential in revealing the inadequacy of general-purpose procedures as a characterisation of domain problem solving and has led to the view that "domain-specific" procedures more readily characterise problem solving in semantically-rich areas of goal-directed thinking.

One noticeable deficiency in both early and more recent research on problem-solving, is the limited amount of any work that has been undertaken with *ill-defined* problems. Investigations that have

involved ill-defined tasks have tended to focus on the role that "analogy" can play in problem solving, and in this regard, puzzle-like problems - particularly Duncker's (1945) classic radiation problem and variants of it - have again featured extensively. Findings arising from the research on analogy in problem solving have considerable relevance to recent theories of problem solving in semantically-rich domains and such findings are therefore discussed in section 2.2.4 prior to the discussion of research on domain problem solving.

2.2.3. EARLY INFORMATION-PROCESSING THEORIES OF PROBLEM SOLVING

Newell and Simon (1972) proposed a theory of human problem solving based predominantly upon research with well-defined puzzle tasks, and many basic tenets of this theory have remained constant throughout its subsequent developments right up, in fact, to its most recent manifestations (see, for example, Simon, 1978). Within this theoretical framework the first thing that a person attempts to do when confronted with a task environment (i.e. a problem and its context) is to form some mental representation of its ***relevant*** features. This internal representation is termed a "problem space", and if its construction has been reasonably successful it will consist of an initial state, a goal state and information about operators and their restrictions. An important point, however, is that features of a problem which are "relevant" for the subject - and which thus form part of his or her problem space - may not, in fact, be the features that are critical for attaining a solution. Indeed, if an inappropriate problem space is constructed, the problem will probably prove to be unsolvable (see Hayes, 1978, for an illustration of this phenomenon).

The actual process of solving a problem which has been mentally represented is viewed by Newell and Simon as involving a "search" through the mental problem space from the initial state, towards the goal state, via a set of intermediate states. Under this proposal, transitions from one state to another are achieved by the application of permissible operators (i.e. rules for action), which, in the case of puzzle tasks, are defined in the problem statement itself and comprise the legal "moves" that may be taken. Since the size of the problem space for many tasks, including transformation problems, is potentially quite vast, Newell and Simon have argued that random or exhaustive search does not occur. Instead it is proposed that people use a variety of general strategies to narrow down considerably the search space. It is argued that these general strategies - or so called "weak methods" - comprise *domain-independent* procedural knowledge which can be applied to process information that has been derived from *any* task environment. A number of these domain-independent strategies have been succinctly described by Newell (1980). Included below is a synopsis of some of the weak methods noted by Newell as well as some others which have been suggested to underly human problem solving:

Generate and Test: Generate in any way possible (e.g. systematically or haphazardly) a sequence of candidate states, testing each for whether it is the desired state.

Heuristic Search: Apply heuristics to reject possible operators from the current state and to reject newly produced states; remember the states with untried operators in some systematic way. Different heuristic schemes yield different searches such as "depth-first", "breadth-first", "progressive deepening" and "best-first".

Hill Climbing: Generate and apply operators from the current state; select the operator that produces a state with an improved evaluation and move to that state. Repeat hill-climbing until no state with an improved evaluation is attainable.

Means-Ends analysis: Compare the current state with the desired state to detect any difference. Use the difference to select an operator that can eliminate it. If the operator can immediately be applied, then do so and continue from the newly attained state using means-ends analysis. If the operator cannot be applied immediately, then use means-ends analysis recursively until the blocking conditions are eliminated. Note that when an operator cannot be directly applied, the recursive use of means-ends analysis will often require building up a mental plan of a sequence of operators that can then be applied directly in the physical world (see the example below of getting to a distant destination).

Subgoaling: This can be used in conjunction with means-ends analysis and usually involves picking out a desired intermediate state on the solution path and attempting to reach this from the initial state as a temporary goal. There are several heuristics for finding subgoals. One method is first to work backwards a little way from the main goal until an intermediate state is found that can then be worked forward to from the start state. Another possibility with some problems is actually to "decompose" a main goal into a few independent subgoals whose solutions can later be combined as an overall solution to the overriding problem. Numerous other subgoaling strategies have been suggested (see Chi and Glaser, 1985, for some further examples).

Newell and Simon (1972) have argued that means-ends analysis is a

particularly useful weak method, applicable not only to puzzle tasks but also many real-world problems. Making a travel plan provides a concrete example of how means-ends analysis might be applied in a real-world situation. Imagine, for example, that you have just received a telephone call in your office that requires you to make an immediate trip to London. Comparing your current state with the goal state reveals a difference that is one of location. Some operators that come to mind such as "walk" or "cycle" can be rejected as unfeasible because they would be too slow. The "going by train" operator, however, seems to be a particularly good possibility. Unfortunately, this operator cannot be immediately applied since trains do not stop in your office. The new desired state is therefore to be at a train station, and again the difference between current and desired state is one of location, so travel operators are again generated, and "taxi" is selected. However the taxi driver doesn't know that he or she is needed; the difference is one of communication. What enables communication? A telephone.. and so on.. When this analysis is complete a mental plan is produced consisting of a sequence of operators that can be applied (in the reverse order to which they were generated) to enable the goal to be attained.

The usefulness of the means-ends approach appears to reside in the fact that only operators that are relevant for getting closer to the goal state are selected. Clearly, by rejecting operators which hold little promise, the solver avoids the backing-up that would be necessary if such operators led on to unviable avenues of exploration. Newell, Shaw and Simon (1958) actually embodied the means-ends method within their famous computer simulation program called the General Problem Solver (GPS). This program was able to tackle a range of tasks

with a fair degree of success including transformation puzzles as well as problems in symbolic logic and integral calculus. Moreover, comparison of the processing steps taken by the GPS program and the steps evident in the think-aloud traces produced by human subjects tackling identical tasks revealed a good degree of match, clearly providing some corroboration for the means-ends model of problem solving.

More recently, the means-ends scheme (and variations of it) has been claimed to mediate subjects solving cryptarithmetic problems (e.g. Newell & Simon, 1972), transformation problems (e.g. Reed & Simon, 1976; Atwood & Polson, 1976) and elementary physics problems (e.g. Bhaskar & Simon, 1977; Larkin, McDermott, Simon & Simon, 1980). Some of the more sophisticated variations of the means-ends strategy that have been proposed to account for problem solving in semantically-rich domains, do, however, seem to go well beyond the assumptions contained in the original formulations of the strategy. What seems to be particularly ironic about studies which have provided evidence for means-ends analysis occurring in domains such as physics is that the subjects looked at have been either (a) novices who have very little experience of tackling domain problems or (b) domain experts given tasks which fall outside their true spheres of knowledge (i.e. problems with which they are unfamiliar or they haven't had to deal with for many years). Studies that have investigated more realistic domain problem solving in which experts work on tasks with which they are relatively familiar, suggest that the strategies used are in fact fairly specific to particular classes of problem situation. The suggestion is that the vast amount of experience that the person has within his or her area of expertise has

led to the acquisition of a wealth of domain knowledge that directly influences problem solving. Some of the major aspects of research on domain problem solving will be dealt with in section 2.2.5, but first some important work that has investigated the role that analogy plays in problem solving will be discussed.

2.2.4. THE ROLE OF ANALOGY IN PROBLEM SOLVING

It has often been proposed that a useful way to solve new problems is to use a method that has proved successful in solving a previous analogous problem. Gick and Holyoak (1980) carried out a series of investigations looking specifically at subjects solving an ill-defined puzzle problem when it was preceded by the presentation of an analogous ill-defined problem and solution. Gick and Holyoak repeatedly found that subjects used the analogy to solve the current problem *only* when they were also provided with a hint that the two problems were related. It generally appeared, then, that whilst people were poor at spontaneously spotting analogies, they were actually quite good at applying them once they had been noticed.

Subsequent studies looking at the role of analogy in problem-solving (e.g. Gick & Holyoak, 1983; Keane, 1988; Holyoak & Koh, in press) appear to have been mainly concerned with determining more precisely the conditions under which prior analogues will prove useful in facilitating solution of a subsequent target problem. What appears to emerge from these studies is that a single analogue can be useful even without a hint - but only when it very closely analogous to the target problem. In this regard an important point to note concerning Gick and Holyoak's (1980) study is that much effort was actually made by them to bury the

similarity between problems under a lot of surface details and dissimilarities. In many real world domains, of course, the similarity that problems bear to each other is likely to be both quite close and thereby more readily apparent. Consider, for example, the real-world case of how students learn from physics texts. Having detailed particular principles which are supplemented with worked out examples that illustrate them, the typical textbook then presents the student with a number of exercise problems to work through. Not only, then, does the mere fact of presenting worked examples prior to exercise problems provides a strong - though admittedly implicit - hint that the former are somehow relevant to the latter, but also the basic similarity between problems is usually quite evident. Indeed, the fundamental purpose behind presenting a set of related exercise problems appears to be to provide the student with an opportunity to abstract ***knowledge*** concerning (a) the common properties possessed by a set of examples and (b) the methods and principles needed to tackle the general class of problems. Within the psychological literature a unit of knowledge which is relevant to understanding or solving a particular class of problems is frequently referred to as a ***schema***.

From a theoretical perspective schemas are interesting since they are knowledge structures which represent both declarative <u>and</u> procedural knowledge (i.e. they include rules for action). Rumelhart (1980) has presented an elegant initial formulation of a schema theory that appears to have much potential for explaining how people can use experience to solve novel problems. Rumelhart has proposed that schemas have six main features:

1. Schemas have variables.

2. Schemas can embed, one within another.

3. Schemas represent knowledge at all levels of abstraction.

4. Schemas represent knowledge rather than definitions.

5. Schemas are active processes.

6. Schemas are recognition devices whose processing is aimed at evaluation of their goodness of fit to the data being processed.

A concrete example (partially derived from Rumelhart, 1980) may suffice to describe the essence of the mechanism which could enable people to tackle a novel problem situation by utilising analogous experiences with similarly structured problems. Consider a schema representing knowledge at a fairly intermediate level of abstraction for the concept "buy". This buy schema, it is proposed, will contain variables such as PURCHASER, SELLER, MERCHANDISE and MONEY and will also contain fairly generalised procedures for action relevant to a buying situation (e.g. check whether several items having a similar desired function are available and at what prices; decide how much you want to spend; investigate the value for money of items within your price range; ask the seller for assistance - and so on). Imagine, then, entering your local hi-fi dealer with the intention of procuring the latest piece of essential equipment to complete your existing system. As aspects of the environment become associated with (i.e. "bound to") variables, so a buy schema will become fully instantiated and procedures for action specific to a "buying" situation will become available. As will be seen in the next section, schemas seem to provide a useful theoretical framework for conceptualising how a person's domain-specific knowledge can play a fundamental role in skilled problem solving with semantically-rich

tasks.

2.2.5. DOMAIN PROBLEM SOLVING AND THE NATURE OF EXPERTISE

Many tasks that we all tackle quite regularly involve goals that are not immediately achievable and so may be defined as "problems" by the definition provided in chapter 1. What is particularly clear about such tasks (e.g. having to find the product of several numbers by long multiplication, preparing meals, or having to drive to the other side of town) is that they exist in domains that are very ***familiar*** to us. In fact our actions appear to have become so automatic that it is hard to realise that we are actually solving problems at all. We are. It's just that with practice we have become proficient in performing the necessary operations in these areas of activity. We have become "expert" at solving these problems. Another conceivable way to look at this (cf. Anderson, 1983) is to say that we have built up from experience a large repertoire of schemas that can be readily applied to solve problems within a familiar domain (as is the case with the purchasing scenario described in section 2.2.4 above).

Whilst the examples mentioned so far in this section represent capabilities of a large proportion of the population, there are many problem-solving tasks at which only a very small proportion of the population ever become expert (e.g. playing chess, composing music, writing research papers and computer programming). The present section focusses on some of the research that has looked at problem solving within these kinds of specialist domains. As will be seen, the majority of studies have been primarily concerned with elucidating whether differences exist in the problem-solving processes and knowledge

structures of novices, intermediate learners and experts within semantically-rich domains.

De Groot (1965, 1966), and later Chase and Simon (1973a, 1973b) attempted to determine what made master chess players different from less-expert ones. De Groot showed tactical chess positions (i.e. real-game configurations in which a number of moves are possible) to former world champions and to Class A tournament players and asked individual subjects to analyse the board to determine what they thought would be the white's best move. The think-aloud protocols that were obtained revealed some intriguing results. Firstly, all players tended to consider roughly the same number of potential moves before chosing one of them. Secondly, all players were seen to use the same "continuation" strategy to evaluate each of these potential moves. This strategy involves pursuing mentally a series of highly probable alternating white and black moves until a clearly identifiable evaluation point is reached. Thirdly, however, was evidence for a striking difference between players in terms of the quality of the move that was finally selected. (Note that move quality was rated by independent chess experts). Interestingly, in this regard, it seemed that upon first seeing the board, the masters were able immediately to apprehend what the truly *best* move should be. This move would be the one that was evaluated first, with other possibilities seemingly only being considered and evaluated as a precautionary measure. It thus appeared that masters have the ability to perceive the problem in a way that rapidly narrows down the search space.

In another of de Groot's experiments, chess masters and less experienced players were presented with real middle-game positions for just five

seconds. It was found that even with such a brief presentation time, the masters could immediately reconstruct positions to a considerable degree of accuracy, often getting sixteen or more pieces right. Novices, on the other hand, only managed to reconstruct correctly five or six pieces. Chase and Simon showed that this result could not be attributed to any visual memory superiority in experts *per se*, because when random configurations were used (i.e. ones unlike real chess configurations) masters did as poorly as novices. Rather, it appears that chess masters have a large repository of stored "patterns", that is, representations in memory of legitimate and meaningful "clusters" of pieces. Simon and Gilmartin (1973) have estimated that, through their vast experience with chess, masters have acquired of the order of fifty-thousand patterns (or "chunks" as they are sometimes termed), and the suggestion is, then, that these stored chunks of domain-related knowledge are what underly superior memory for board positions.

Before considering further research on chess skill, it is worth mentioning that studies have also been undertaken looking at expert-novice differences in memory for game information in other domains of adversary problem solving such as GO (e.g. Reitman, 1976) and bridge (e.g. Engle & Bukstel, 1978; Charness, 1979). Generally the results arising from these studies are very similar to those obtained within the domain of chess, that is, higher levels of expertise are associated with greater recall ability for structured game information (e.g. bridge hands in which the cards were arranged systematically).

Returning, then, to the domain of chess, Chase and Simon (1973a, 1973b) investigated further the nature of the chunks underlying chess memory by requiring that subjects reproduce on one board a position that remained

in view on another board. Video protocols were obtained and later analysed and revealed that subjects were exhibiting a repeated behavioural sequence involving (a) glancing at the original board (i.e to encode a single meaningful pattern) and then (b) placing a few pieces on the reconstruction board. The clusters of pieces that were reconstructed by masters after each glance tended to be highly meaningful relations such as "pawn chains", "castled-king positions" and various attack and defence relations. Interestingly too, it was shown that less skilled players produced similar clusters to those of the masters, although they generally contain fewer pieces. Other studies by Chase and Simon (1973a, 1973b) were important in demonstrating that in addition to knowing many relational patterns between chess pieces, skilled players also appear to know "what to do" in the presence of such patterns. Indeed a common feature of games between high-level players is the frequent occurrence of stereotyped sequences of moves. It would seem, then, that the so called "chunks" could quite legitimately be conceived of as schemas that contain both declarative knowledge (i.e. relational patterns of pieces) as well as procedural knowledge (i.e. relevant actions or responses to be made).

Research within the domain of computer programming has provided further evidence for the view that certain aspects of expertise reflect such sophisticated organisations of knowledge in long term memory. McKeithen, Reitman, Reuter and Hirtle (1981) and Schneiderman (1976) have both performed experiments on memory for computer programs that are very similar to the chess experiments of Chase and Simon. In one study carried out by McKeithen et al. (1981), for example, experts and novices had to attempt to reconstruct programs after brief presentations. With

meaningful programs the experts were seen to display far higher recall accuracy than novices. With scrambled programs, however, both groups performed equally poorly. Soloway and Woolf (1980) has argued that expert programmers have developed a vast number of "templates" for aspects of programs that frequently occur. Moreover, programmers have associated these templates to specific programming goals, so that when they are confronted with these goals, the templates can be immediately generated. Expert and novice programmers also show some interesting differences (and similarities) in the way that they design software (see Anderson, Farrell & Sauers, 1984; Jeffries, Turner, Polson & Atwood, 1981) though discussion of this research will be deferred until later sections of this report.

In research on problem solving in other domains such as physics and geometry, the picture that has emerged is very similar to that which has already been presented. One particularly interesting set of findings from these domains (see, for example, Larkin, McDermott, Simon & Simon, 1980; Larkin, 1981; Anderson 1981) reveals that experts have developed excellent "forward searching" abilities. What this implies is that when confronted with the statement of the problem, experts are able to make very accurate inferences about the information that will be needed to solve the problem. Novices, on the other hand, are inclined to either use a means-ends strategy to tackle the problem, or search "backwards" from the goal for subgoals. Again, then, the suggestion is that experts have learned through their domain experience to associate inferences with particular patterns of features that are present within a problem.

A set of results which relate to this latter idea (again derived in the formal domain of physics) show that experts tend to represent problems

in terms of the underlying principles of physics by which they can be solved, whilst novices tend to be heavily influenced by surface features that are far less likely to lead to an appropriate solution method (see Chi, Feltovich, and Glaser, 1981; Chi, Glaser & Rees, 1982). Chi et al., (1981), for example asked subjects to classify an extensive number of physics problems (accompanied by diagrams) into similar groups, and to give explanations for their categorisations. It was revealed that novices chose the surface characteristics of the problems that were manifest in the diagrams (e.g. inclined planes or rotations) as a basis for classification. With experts, however, problems that were completely different on the surface were seen as similar if they reflected the same underlying principles such as the Conservation of Energy or Newton's second law of motion.

Most of the domain-related problems that have been discussed so far in this section appear to be quite well-defined. An important issue, however, is whether the nature of the expertise involved in tackling such well-defined domain problems also captures the nature of problem solving in the case of ill-defined domain tasks. Indeed this issue is particularly relevant in the context of the present thesis, since many researchers have suggested that design tasks are particularly clear cases of ill-defined problems (see, for example, Eastman, 1969a; Simon, 1973, and also section 2.3.1 below). Generally, it appears that little systematic research has actually been carried out with ill-defined tasks at all and that which has been undertaken has (a) mainly used semantically-impoverished tasks and (b) predominantly been concerned with examining the role that analogy plays in problem solving (see section 2.2.4 for a discussion of this research). Some work carried out

by Voss and his associates (see Voss, Greene, Post & Penner, 1983a; Voss, Tyler & Yengo, 1983b) is, however, noteworthy because of its focus on expert-novice differences in problem solving with ill-defined, *semantically-rich* tasks derived from the domain of political science.

In one such study (Voss et al., 1983a) the task confronting subjects was to suggest ways of increasing crop production within the Soviet Union. Think aloud protocols were obtained from three types of subject, that is, (a) expert social scientists specialising in the Soviet politics (b) students taking a course on Soviet domestic policy and (c) chemistry professors. Not surprisingly, strong effects of expertise were observed - with the best solutions being produced by the experts and the worst by the students and chemists. Of particular interest were results indicating that the Soviet specialists tended to engage in three major phases of problem-solving activity. Initially, then, they spent a considerable time (i.e. about a quarter of their total solution time) actually reviewing and representing the problem at hand. In this "problem representation" phase, constraints would be identified (e.g. "available arable land") and the problem would be decomposed into a few high-level subproblems reflecting possible general *causes* of the overriding problem. In the second phase of problem solving, the experts proceeded to find a solution for the overriding problem (e.g. "improve investment in agricultural technology") which also served to solve a number of subordinate problems. In the third and final phase of their work, the experts then explored the ramifications of their proposed solution and strived to support the solution with detailed argument.

Voss et al's study additionally indicated that not only experts but also

novices made use of the problem reduction (or subgoaling) strategy - although the novice subjects tended to seek specific solutions for many separate "cause-oriented" subproblems rather than the all-encompassing solutions sought for by the experts. Interestingly, subgoaling is a general problem solving method commonly observed in the context of more well-defined problem solving (see section 2.2.3). In the case of political science problems, it appears that the use of a problem reduction strategy helps the the solver to convert the initially ill-defined problem into a set of more tangible and more well-defined subproblems. Voss et al. (1983a, 1983b) also noted that in many ways the solution-oriented activity of their expert political scientists was very much like that of other types of expert. That is to say, experience appears to have led to the development of a large repertoire of schemas onto which problem information can map which fairly readily leads to the generation of solution concepts. However, unlike physics, for example, where it appears important to expose learners to selected, special cases of problems (see Larkin, 1979), the indication appears to be that skill in political science is crucially dependent upon exposure to a large *variety* of problem types.

2.2.6. INTERIM SUMMARY AND CONCLUSIONS

So far, then, the present chapter has reviewed many important studies that have been undertaken on human problem solving within the information-processing framework and has introduced several important notions concerning the nature of the knowledge involved. Consideration of the distinction between well-defined and ill-defined problems provided a convenient entry point for this survey. The distinction,

though itself somewhat ill-defined, appears to be a useful way to conceptualise qualitative differences between problems. Most of the research work that has been undertaken on problem solving has been concerned with investigating problem solving on fairly well-defined tasks. Many of these tasks have required the solver to have only a minimal amount of background knowledge (semantically-impoverished problems), though more recent studies have shifted in emphasis toward problem-solving tasks in real-world, semantically-rich domains. Early research - which predominantly used well-defined puzzle tasks - led to the advent of Newell and Simon's (1972) highly influential theory in which problem solving is characterised as a search through a problem space guided by the application of general strategies (the so called weak methods such as means-ends analysis) that are essentially both content-independent and context-independent.

Whilst this initial theoretical framework seemed adequate for describing behaviour in contrived puzzle situations, contemporary research has shown it to be somewhat too simplistic an approach for understanding domain problem solving. Recent investigations of novices and experts attempting problems within their particular areas of endeavour continually reveal that specific knowledge of the domain is crucially important for the effective solution of problems. In research on problem solving in domains as disparate as chess, computer programming, physics and geometry, a consistent picture has emerged which indicates that experts have learned to perceive recurring "patterns" in the problems that they work on, and have learned to associate their problem solutions to these patterns. These units of conceptual and procedural knowledge have been termed schemas and the suggestion is that an expert will

possess a vast store of these that can be evoked by environmental or internal conditions. Moreover, these schemas will be organised at different levels of abstraction. This is demonstrated by the finding that problem-solving ability can be transferred between tasks that differ in their information content but which possess similar underlying structures. Indeed, expertise implies the ability to develop procedural knowledge that whilst being domain-specific is still generalisable to problems of a similar "type" within the domain. When working on a problem, the, experts appear to be able to rely on the recognition of familiar patterns which leads to the subsequent evocation of pattern-specific problem-solving processes that readily lead to solutions. Novices, contrastingly, are usually seen to have very limited domain-specific recognition and solutions processes (that is, they have a limited store of relevant schemas) and appear to rely - at least initially - on very general, domain-independent procedures. Familiarity with the domain appears to allow (somehow) for the gradual acquisition (i.e. learning) of the schemas that underly expert performance.

It should be noted that (a) investigations of novice and expert differences in real-world domains of skill and (b) research on the role of analogy and experience in problem solving, have both been highly influential in leading contemporary cognitive psychologists toward an interest in the dynamics of human "learning processes" and away from the earlier emphasis on understanding and modelling "performance processes". Such a shift of concern is surely a worthy endeavour and one which is currently producing a multiplicity of sophisticated and intriguing cognitive theories of skill acquisition and ideas as to how schemas may be built up and refined (see, for example, Holland, Holyoak, Nisbett &

Thagard, 1986, and the selection of papers contained in Anderson, 1981). Anderson (1983) has in fact argued that the development of expertise in quite specialised areas such as scientific research is essentially no different to the development of the skill that is exhibited by the majority of individuals in everyday areas of activity.

Most of the problems that have been discussed in this chapter so far - whether semantically-impoverished or semantically rich - have been well-defined in nature. Little systematic research has actually been carried out with ill-defined tasks at all - though some of the work on analogy is problem solving has employed ill-defined puzzles. The investigations undertaken by Voss et al. (1983a, 1983b) are interesting in that they represent some of the few studies to have actually used ill-defined tasks derived from semantically-rich domains to look at of expert-novice differences in problem solving. Aspects of the results of this study appear to parallel closely findings relating to domain problem solving with well-defined tasks. Another area of human endeavour in which the problems appear to arrive in an ill-defined state is the domain of *design* which forms the all-important focus of the next half of this review chapter.

2.3. PROBLEM SOLVING IN DESIGN.

2.3.1. EARLY APPLICATIONS OF PROBLEM-SOLVING THEORY TO DESIGN

Empirically based work attempting to explicate the psychological processes involved in tackling design tasks began to emerge in the late 1960's. Such studies, however, were few in number, and this initial

state of affairs has seemingly remained unchanged right up to the present day. Even the relatively recent and continuing surge in development of a multiplicity of computer-aided design systems has only seen a very limited increase of interest in the psychological characteristics of design. Indeed, as was noted in chapter 1, most of the design tools that are available for designers (whether they be architects, computer programmers or engineers) do not appear to be derived from any explicit psychological model of the design process. It is arguable that certain inherent features of design tasks have been highly influential in stifling the enthusiasm of would-be researchers in this area. Most design problems, for example, involve (a) quite considerable time scales for solutions to be attained (b) the application of complex technical information and (c) the frequent participation of many different people. Such features of design tasks clearly give them an aura of intractability as far as psychological investigation is concerned. Not surprisingly, then, those researchers who have attempted to investigate the nature of design processes have, for practical reasons, attempted to simplify and control the nature of the tasks employed. Thus, most studies of design have employed small-scale tasks being tackled in a laboratory setting by single designer.

The pioneering investigator into the cognitive processes involved in design appears to have been Eastman (1968, 1969a, 1969b) who approached this domain with many of the information-processing notions that were arising from contemporaneous studies of problem solving which employed well-defined tasks (see section 2.2.2 & 2.2.3). It is noteworthy that Eastman seemed impressed by the *ill-defined* nature of design tasks and appeared to be motivated not so much by a genuine interest in

studying the cognitive processes in design *per se*, but rather by the desire to extend the "general purpose" conceptualisation of problem-solving mechanisms into domains where tasks tend to fall well toward the ill-defined end of the problem continuum (refer to section 2.2.1 and also see below). It is perhaps unfortunate, and certainly ironic, that Eastman actually directed his research toward quite trivial design problems and often ones that were not strikingly ill-defined. It would seem judicious, however, not to dismiss this work immediately but to instead briefly examine the nature of the tasks studied and the findings that were obtained.

Eastman's investigations were almost exclusively focussed on the "space-planning tasks" that are ubiquitous in engineering, architecture, and urban design. Space planning involves the selection and arrangement of physical elements (i.e. design units) in a two- or three-dimensional space, subject to a variety of constraints - this latter term being used loosely by Eastman to refer to the desired "relationships" between elements and the desired "attributes" of elements. One small-scale, space-planning task that was prevalent in Eastman's studies may be termed the "bathroom re-design task". The subject would be given an existing design - in plan view - of a bathroom layout and would be required to re-design it so as to make it more spacious and luxurious. A boundary would also be set on the extra finance available for the purchase of alternative or additional fittings and accessories. Such tasks were generaly fairly open-ended in that more information was required to solve the problem than was initially given. In these experiments, then, the subject could seek additional information from the investigator if this was desired.

The subjects studied by Eastman ranged from those totally unfamiliar with design work to those with some design experience, although there is no indication that any subjects were derived from populations of superior or expert designers. Eastman obtained verbal protocols of subjects attempting their re-designs and additionally collected any pen-and-paper work they produced. What was consistently found was that subjects initially devoted considerable time to gaining a clearer specification of the problem goals (i.e. the existing and extra design units for inclusion in the new design) together with constraints relating to these. Most subjects tended to augment the problem givens with both personal knowledge of what a good bathroom should be like and information obtained from the experimenter. When a clearer definition of the problem had been derived (i.e. it had become reasonably well-defined), the subjects were then seen to proceed in searching for a configuration of design units that matched the constraints that had been generated. Often during this search, new constraints were seen to be generated and sometimes original ones either overlooked or rejected. The scheme of the search, however, usually appeared to proceed according to the use of very general problem-solving strategies such as means-ends analysis and generate-and-test, along with some intelligent guesswork.

While Eastman's results have elements that are of interest, it is clearly arguable that his findings concerning the strategies adopted by subjects in tackling these space-planning problems relate only to the tasks used (i.e. trivial re-design problems) and the subjects studied (i.e. inexperienced designers). The generalisability of these results to experts dealing with complex design problems is therefore highly questionable. Kaplan (1968) stresses a similar point when he says that

"Eastman appears to assume that the superior designer proceeds in the same way as the ordinary designer ... Without such an underlying assumption, it is difficult to understand how he hoped to elucidate the nature of intuitive thinking in superior, creative, innovative designers by studying people who were not known for superior design". In the light of recent research on domain problem solving that has revealed the highly domain-specific nature of expert problem-solving processes, it could be argued that the research approach adopted by Eastman was doomed to inadequacy from the start.

While certain design tasks such as those studied by Eastman appear fairly superficial, they certainly have an aura of ill-definedness about them. Indeed, evidence from informal observation and empirical investigation suggests that most real-world design tasks are truly classic cases of ill-defined problems. For most design tasks, then, it appears that:

(1) the initial statement of the design problem shows incompleteness in information content concerning (a) the goals to be achieved (these tend to be very broad, fuzzy and poorly articulated), (b) the initial state (specification of this may range from nothing whatsoever to sometimes fairly detailed material on a pre-existing design to be altered or extended, (c) the operators to be used (a considerable number of operators will be available for transforming any design state, but it is unlikely that any of these will be explicitly mentioned), and (d) the constraints (these often appear to be given in fairly global and loose terms);

(2) many of the criteria that determine whether the goal has been

truly and satisfactorily achieved will be complex and inexact. Aesthetic criteria, for example, will vary considerably from one individual to another;

(3) there is certainly no "simple legal move generator for finding all the alternative possibilities at each step.

Simon (1973) warmed to the theme of the ill-defined initial form of design problems during his contemplation of the processes involved in ***architectural*** design, and he pointed out that for many architectural tasks the problem space cannot be objectively described in any meaningful and "limited" sense as is possible in the case of well-defined tasks. To do this the problem space would have to encompass (1) all the possible structures (e.g. geodesic dome, truss roof, cantilevers, arches, etc.) that an architect might at some point consider, together with (2) all the kinds of materials (e.g. wood, metal, plexiglass, reinforced concrete, camel's hides etc.) that might be contemplated, in addition to (3) the complete stock of operators that could be made use of. (Note that the examples in parentheses are derived from Simon, 1973).

Simon additionally sketched out a theory of the psychological processes involved in design that is worthy of some attention. Simon's theory was based in part upon an empirical study carried out by Reitman (1965) who obtained and analysed a think-aloud protocol of a professional composer writing a fugue (i.e. an ill-defined task). From this protocol it was noticeable to Reitman that over any short interval of time the composer was dealing with perfectly well-defined subproblems that were part of the overall ill-defined problem. It appeared that as the composer continued at his activity, new information was repeatedly being evoked

from memory that sequentially transformed the problem space he was working in so that it related to a series of specific and fairly clearly structured subtasks. Moreover Reitman noted that there was a tendency for these subtasks to be hierarchically arranged such that "Just as 'sentence' transforms to 'subject plus predicate', and 'subject' may transform to 'article plus noun phrase'..., so 'fugue' may be thought of as transforming to 'exposition plus development plus conclusion', 'exposition' to 'thematic material plus countermaterial' and 'thematic material' to 'motive plus development of motive'". Interestingly, then, the cognitive processing that appeared to be involved in the development of this fugue appeared to be driven by the recognition of familiar configurations of features that arose along the solution path. Thus, relational patterns that were recognised in the melodic fragments that had had already created evoked ideas about how to proceed further. These findings clearly compare well with results concerning the nature of expert thinking in relatively well-structured, semantically-rich domains discussed in section 2.2.5.

Simon's (1973) theory relates not to a composer of fugues but to a hypothetical architect designing a house. From his speculative musings it appears that he is proposing a general scheme along which the design process can be imagined to proceed. A possible interpretation - perhaps incorrect - of this suggested scheme is as follows. Firstly, then, the architect will have been given some specification by the client concerning such factors as the family size and needs, the number of rooms required and the available budget. (Additional specification may later be obtained from architect-client dialogues, but still many design goals will remain incompletely specified). Secondly, then, taking these

initial goals and constraints, the architect begins to derive some global specification from them, such as the square- or cubic-footage of the house. Thirdly, however, the task itself - "designing a house" - evokes from long-term memory (a) other attributes of the house that will have to be specified at an early stage of the design, such as its general style, whether it is to be single- or multi-leveled, the type of structural materials to be used, and (b) some overall high-level "executive program" for controlling the design process. Simon suggests that this executive program will not provide a strict and rigid procedure for designing a house, but rather will allow for fluidity within the design sequence. Thus although it carries out high-level coordination and controlling functions, much of the cognitive processing that goes on within the confines of this control will be stimulus-driven, that is, "elements already evoked from memory and the aspects of the design already arrived at up to any given point, would serve as the stimuli to evoke the next set of elements". Requirements that any components of the design should meet can be similarly evoked at appropriate times in the design process, as can design alternatives and appropriate courses of action to pursue.

According to this proposed scheme, then, what is initially a seemingly ill-defined design problem to the architect "begins to acquire structure by being decomposed into various problems of component design, and by evoking, as the design progresses, all kinds of requirements to be applied in testing the design of its components. During any given short period of time, the architect will find himself working on a problem which, perhaps beginning in an ill-structured state, soon converts itself through evocation from memory into a well-structured problem".

Simon (1973) suggests that a "production system" model of the designer's knowledge would allow for such flexibility in the design process. In psychological terms, a production system is a theoretical formalism useful for characterising the procedural knowledge underlying cognitive processing. A production system consists of a set of "productions", which are rules possessing two components, that is, a "condition" component and "action" component. The condition typically consists of a statement of a goal and certain tests to determine if the rule is applicable to that goal. If these tests are met, the rule will apply and the action will be performed. The action may involve changes in working memory or long-term memory (storage or retrieval), or motor acts. It is noteworthy that production system models are a pervasive feature of Newell and Simon's (e.g. 1972) approach to problem-solving in which a person's procedural knowledge is conceived of as being essentially general-purpose in nature. The productions presented below, for example, have been devised by Anderson (1981) to represent part of a problem solver's domain-independent, recursive "means-ends" strategy:

```
P1 ..     IF  the goal is to eliminate a difference "D"
              and "O" is an operator relevant to reducing the difference
        THEN  set as a subgoal
              1. To apply "O"


P2 ..     IF  the goal is to apply an operator "O"
              and "D" is the most important difference between
              the application condition "O" and the current state
        THEN  set as a subgoals
              1. To eliminate the difference "D"
              2. To apply the operator "O"


P3 ..     IF  the goal is to apply operator "O"
              and there are no differences between the application
              condition of "O" and the current state
              and operator "O" calls for action "A"
        THEN  set as a goal to perform "A"
```

The use of productions for modelling information processing, however, does not need to be restricted to domain-general knowledge, and Simon's (1973) theory of design capitalises on the involvement of productions that are specifically related to the domain of architectural design. The productions in Simon's theory act as a recognition and retrieval system which continually modify the problem space and lead to new courses of action. It is noteworthy that a production system theory of the nature of knowledge and its role in information processing appears to have much in common with the schema theories discussed in section 2.2.5. Indeed many formulations of schema theory actually incorporate productions into the schema units (see, for example, Cheng & Holyoak, 1985). A final point concerning Simon's theory is that although it has much in the way of intuitive appeal, it is presented without any empirical back-up and therefore appears as purely speculative. It does, however, seem to allow for the derivation of experimentally testable predictions about the nature of cognitive processes in design.

A point not noted by Simon, but which is pursued by Moran (1968) in his "model of a multilingual designer" is that a designer cannot generally search for solutions to a problem by actually building in the real world the states along the solution path, but must instead make use a rich variety of "languages" - not necessarily verbal - for modelling these states in a problem space. This modelling is carried out by constructing representations, both internal (in the head) and external (e.g. on paper). Representations that may be constructed externally are suggested to include schematics, block diagrams, graphs, flow charts, mathematical models and the like, and Moran hints at the possibility of "mental imagery" as a form of internal representation, though such a notion is

nowadays a highly controversial issue (see respectively Pylyshyn, 1973, 1981 and Kosslyn, 1980, for some contrasting views on the role that imagery plays in cognition). Moran (1968) proposes that whenever a designer translates a portion of his problem from some base representation into a new kind of representation he or she will be working on a subproblem in a subspace of the solution space. The formal notation available for working with one type of representation will often be quite restrictive. The suggestion is, then, that an important aspect of expert design is to be able to deal with a problem using a multiplicity of representational systems each of which can capture aspects of the problem which would be logically precluded by the use of a different form of representation. It is interesting in regard to Moran's ideas, that a contemporary theoretical notion has emerged known as "mental modelling" which refers to cognitive mechanisms which are capable of simulating, for example, the qualitative aspects of a device's functioning (see section 2.3 below).

2.3.2. RECENT STUDIES OF DESIGN

The literature reveals an extreme want of anything in the way of relatively recent research on the psychology of design and suggests that investigation of this area has hardly been pursued at all. Out of the few reported studies that do exist, that carried out Jeffries, Turner, Polson and Atwood (1981) is particularly worthy of attention because of its detailed and systematic nature. These researchers carried out an experimental investigation on software designers by collecting TA protocols from both novice and expert subjects as they worked on a moderately complex programming problem. It was found that both types of

designers developed programs in what is termed a ***top-down*** manner. What this means is that the subjects tended to work from the statement of the problem towards a solution by decomposing the initial task into a series of minimally interacting high-level subproblems which were then worked at separately - often themselves being broken down into more and more subtasks. This can be imagined as an expanding tree-like structure with the solution elements at the end of each branch forming the details (actual code) to be combined to make up the complete solution. As Anderson (1985) notes, this top-down development in software design is basically the same as the "backward working" strategy that is observed in geometry and physics where the solver searches back from the goal for possible subgoals. In these domains this method is usually used by novices, while expertise is associated with a change to working forward. Anderson suggests that working forward is a very good strategy in physics and geometry where problems have a rich set of givens that are more predictive of solutions than the actual goal. Contrastingly, argues Anderson, in software design the goal is usually the most elaborate part of the problem statement and is likely to be highly predictive of a solution method. In design domains, then, working back from the goal statement to subgoals appears to be the method *par excellence.*

Jeffries et al. (1981) found not only similarities between novice and expert designers, but also some essential differences. Although both used top-down strategies, experts tended to expand their designs ***breadth-first,*** whereas novices tend to expand their designs ***depth-first.*** In the former case the expert works at breaking down the total set of subproblems that exist at at one level of detail before

then paying attention to the set of newly produced sub-subproblems, and so on until the lowest level is attained. In the latter case the novice concentrates on pursuing a single high-level subproblem right down to its very lowest level of detail before focusing on another high-level subproblem, and so on until all subproblems have been dealt with. The breadth-first method appears to be particularly advantageous when subgoals within the developing design have some degree of interdependence (i.e. when a certain solution to one subgoal could hinder or prevent the solution of another associated subgoal). Similar findings to those obtained by Jeffries et al regarding strategies in software design have been reported by Anderson (1983) and Anderson, Farrel & Sauers, (1984). Other differences between expert and novice designers that emerged in this study related to (1) the evaluation of alternatives (2) the retrieval of known solutions (3) the ability to access background knowledge (4) the understanding of concepts and (5) the understanding of implications. In all these respects experts displayed a considerable superiority to novices - a superiority, it seems which is again based on familiarity with the domain.

The experimental work conducted by Jeffries et al. was fundamentally aimed at testing their theoretical formulation of the global processes that experts use to control the development of a software design. These "global control processes" may be equated with what Simon (1973) was referring to as the "executive program" in his theory of the architectural design process, and which he failed to explicate in any real detail. Jeffries et al.'s fundamental assumption is that skilled software designers have implicit knowledge concerning the overall structure of a good design and of the process of generating one. This

knowledge is therefore highly abstract, relating to no specific tasks in particular but to software design tasks in general. Additionally, this knowledge is both declarative and procedural in nature and is organised in what they refer to as the ***design schema*** which develops and becomes refined as a result of experience with software design. Whilst originally, then, a designer's approach to the design might tend to involve very general problem-solving strategies such as subgoaling or a rudimentary "divide and conquer" technique, as the individual becomes more and more familiar with the domain these general methods are transformed into a specialised design schema. Jeffries et al. suggest that whenever the designer's specialised schema is inadequate for tackling a problem then the more general strategies will take over. Expert and novices differences in the development of software designs can therefore be related to the fact that the experts will have firstly a more adequate, polished and sophisticated design schema and secondly a greater quantity of domain knowledge at all levels of abstraction which is also more well organised.

Jeffries et al. view the design schema as driving the generation of a software design by breaking up the initial task into a set of subproblems. How this decomposition proceeds will, of course, differ somewhat from expert to expert since their experiences (for example, with the professional literature on formal design methodologies) will be different. Jeffries et al., however, believe that the essential form of the schema will be reasonably similar for most experts and they simplify their discussion by considering a design schema that they believe to be prevalently used that decomposes a problem in terms of input, process and output elements. The basic idea, then, is that the initial

application of the design schema will coordinate the modularisation of the problem into subproblems relating to input, process and output functions. This modularisation is achieved by the recognition and retrieval of specific and concrete knowledge relevant to the problems, presumably - though this is not explicitly stated - by the operation of other schemas at lower levels of abstraction. The design schema will then *coordinate* and *control* the sequence in which subproblems are worked on, the search and retrieval of potential solutions or analogous solutions, the evaluation of the applicability of alternative solutions, and the further expansion of subproblems - in which case the design schema is applied ***recursively***. It will also coordinate the selective storage (in long-term memory) of newly generated or acquired information, which must be integrated with the existing store of factual knowledge of computer science, and procedural knowledge of solution methods. No doubt, too there would be other unmentioned or unthought of functions that the design schema would need to perform. Needless to say, Jeffries et al. provide little in the way of a detailed exposition of certain of these aforementioned mechanisms, and they admit that their theory is at an early stage of development. The major control processes of the design schema are, however, discussed extensively, and are expressed as series of production rules. For example DESIGN SCHEMA RULE 1 shown below functions to provide a coherent way for the production system to determine what problem to tackle next. The rules assume that the list of unsolved problems are stored on an agenda.

DESIGN SCHEMA RULE 1: SELECTION RULE

```
IF      (no current subproblem exists)
    AND (any unsolved subproblems on agenda)
THEN    (select highest priority subproblem or, if multiple subproblems
         at highest priority, select next subproblem in breadth-first
         order at highest priority and make it a new current subproblem)
```

Jeffries et al.'s theory of the cognitive processes underlying software design appears similar in many respects to Simon's (1973) theoretical speculations on the mental processes involved in architectural design. Jeffries et al.'s theory, however, is certainly superior in that it presents a fairly detailed statement regarding how the overall coordination and control of processing is accomplished by means of an expert's high-level knowledge of software design. Numerous empirically testable predictions can be formulated from this theory, and Jeffries et al. themselves present much experimental data which appears to be compatible with their theoretical formulation. Whilst some of the findings of this empirical study have discussed above, for a full rendition of the goodness of fit of the complete set of results to the model, see Jeffries et al. (1981).

Moving on to consider some other recent ideas concerning the psychology of design, it is noteworthy that Anderson (1985) has suggested that within a design situation, the stated "goals" are of essential importance to the designer since they are likely to be highly predictive of a solution method. Certainly a common finding in the studies by Eastman was that subjects spent a considerable time clarifying problem goals when these were deficient in the given problem specification. Presumably, then, goal-oriented information is highly directive in leading the way toward a solution path since conceptual and procedural knowledge relevant to such goals is evoked from memory and readily enables the narrowing down of a potentially enormous search space of design concepts. Malhotra, Thomas, Carroll, & Miller (1980) present some results of an observational/experimental study that are of interest in this regard, in which designers and clients were videotaped while

engaged in problem-oriented dialogues. The situations involved a client consulting with an expert designer who is expected to derive a solution for the problem at hand. The complex dialogues that ensued revealed an essentially cyclical structure involving three basic stages. Firstly goals are stated, developed, examined and understood, next partial solutions are outlined, and finally partial solutions are developed further, carefully evaluated and possibly accepted. New requirements were frequently seen to be uncovered during the process of solution examination and these requirements often initiated a new design cycle. Malhotra et al. termed the initial stage of the design cycle "goal elaboration," and it was frequently observed to be composed of substages involving decomposition of a parent goal into subgoals reflecting functional requirements. The study also revealed that once subgoals had become well developed, solutions to the problem appeared to follow from these quite readily. It should be noted, however, that Malhotra et al. fail to provide any theoretical description of the cognitive mechanisms that underly behaviour in these client-designer dialogue situations.

2.3.3 MENTAL MODELS AND DESIGN

Before concluding the present review of psychological research on design it is important to discuss the theoretical notion of a "mental model" which was briefly introduced in section 2.3.1. This notion is nowadays very popular in cognitive psychology and it has in fact been applied in the context of design problem solving (see below). Unfortunately the concept itself, whilst frequently bandied about in the literature, remains rather poorly defined in many fundamental respects (cf. Rips, 1986). Much of the confusion surrounding the term mental

model appears to derive from the fact that it is often used to refer to quite different, though mutually compatible, aspects of cognition (see for example, Evans, 1988; Gilhooly, 1987).

The first way in which the term is used, and perhaps the simplest to understand, is in the sense of a mental representation of a situation in the real world (or in a possible world) which is formed by a person and which can be manipulated and changed (cf. Craik, 1943). The idea here is that the mental representation or model is only an internal *analogue* of that which it represents. Johnson-Laird appears to use the term mental model in this sense in the theory he proposes of deductive reasoning where the models constitute mental encodings of situations described by verbal propositions (see for example Johnson-Laird, 1983; Johnson-Laird & Bara, 1984). A second and related way in which the term mental model is used is in the related sense of a representation that, once formed and stored in memory, exists as a "knowledge structure" which can itself be later *evoked* and *used* to interpret and represent new information (see, for example, Thorndyke, 1984). Like the first sense of the term, this second sense is also a fairly "static" one in that models essentially provide a *framework* for either (a) recognising and encoding new information into pre-existing categories or (b) building up new models based on the retrieval of existing models.

A third are rather different way in which the term mental model is used by contemporary psychologists, however, is in the much more dynamic sense of a knowledge structure that contains both declarative *and* procedural knowledge (cf. the schema notion introduced earlier). In this third sense, then, the model includes mechanisms not only for

encoding information but also for making ***inferences*** and ***predictions*** that go beyond the encoded information. Generally the important idea here - which renders the mental model different to a schema - is that the model itself can be "run" to enable either (a) forward exploration from a current state of affairs in order to generate predictions and forecasts such as anticipating the probable outcomes of alternative decisions (i.e "what if?" or "forward reasoning") or (b) backward exploration from a current state of affairs to evaluate possible scenarios by which the current state might have come about (i.e "backward reasoning" which might occur, for example, in fault diagnosis). Clearly the underlying idea in these latter uses of the term is that of active modelling or what may more usefully be termed "mental simulation" (see for example, Kahneman & Tversky, 1982; Borgman, 1986).

The mental models notion in its third sense has been applied to the engineering domain to describe the way in which electrical engineers reason about qualitative aspects of a circuit when analysing it (see particularly de Kleer & Brown, 1983; de Kleer, 1984). With respect to a theory of engineering design, then, the implication is that the engineer could engage in mental modelling to simulate the dynamic behaviour of a particular collection of subcomponents being used to synthesise a system. This would allow for the comparison and evaluation of alternative designs to be achieved. The ability to construct and use mental models within a domain is certainly a capability that is developed through experience and familiarity with the area itself. It should finally be noted, however, that no research appears to have been done to investigate directly the mental models' theory within design contexts.

2.4. CONCLUSION TO THE LITERATURE REVIEW

2.4.1. COGNITIVE PROCESSES IN DESIGN: A SUMMARY AND SYNTHESIS

One overriding purpose of this present review chapter has been to present a survey of research on human problem solving since the findings and theories arising from this work appear to have formed an important foundation for the few investigations that have been undertaken on design itself. General characteristics concerning the nature of problem-directed thinking were drawn out and discussed throughout the first half of the chapter and were additionally summarised in section 2.2.6.

The second overriding purpose of this chapter has been to review the limited psychological research that has been undertaken on design. Generally, many of the concepts and principles introduced in the first half of the chapter appear to have been successfully applied within the context of understanding the nature of design problem solving. This is heartening since it seems to indicate the validity of an approach which uses established theoretical frameworks for understanding the processes involved in novel domains of psychological interest such as design (refer to section 2.1.1).

One particularly noticeable feature concerning psychological research on design is the sheer scarcity of systematic studies that have actually been undertaken. Indeed investigations of "engineering design" - the subdomain which the present thesis is predominantly focussed on - appear to be all but nonexistent. Thus the review of design research has, of necessity, involved an examination of studies of software design and

architectural design. Interestingly, it appears that many of the findings that emerge from these separate domains are mutually compatible, suggesting that similar cognitive mechanisms may underly problem solving in disparate domains of design endeavour. The aim of this final section, then, is actually to attempt a tentative synthesis of the current understanding of design processes in a way that is suggestive of a consistent, though somewhat loose, framework to motivate further exploratory research.

(1) In general, design tasks appear to arrive in an ill-defined form since the initial problem statement lacks completeness concerning the goals to be achieved, the operators to be used, and the constraints that restrict the application of operators. Additionally the criteria that determine whether goals have been achieved tend to be fuzzy and subjective. The starting point for the designer, then, is actually to attempt to define the problem. In design situations goal-oriented information tends to be highly predictive of solution methods and is thus particularly important for narrowing down a potentially huge search space of design concepts. The designer will, therefore, strive to gain a clear understanding of the nature of the artifact that must be designed, especially its functional requirements (concerning what it must do) and its performance requirements (relating, for example, to its style, reliability and manufacturability).

(2) The definition of goals, requirements and constraints generally enables the designer to decompose the overall task into a collection of manageable and minimally interacting subproblems which may themselves be factorised further (this is termed top-down expansion). At the highest level, these subproblems seem to relate to separate functionality goals.

Top-down expansion has been demonstrated empirically in software design and there is an abundance of informal literature that suggests that it is a prevalent method of working in engineering design. This top-down development of a design will mean that at any particular moment in time, the designer will actually be working on a fairly well-structured subproblem that exists in a subspace of the overall problem space. The designer, then, may be conceived of as moving between subproblems in a hierarchy of problem spaces. The designer will also shift between problem spaces when attempting to model aspects of the problem using different notational systems such as schematics, graphs, mathematical models and the like.

(3) As a top-down development of a design progresses, the designer will be working on problems at greater levels of detail. In the design of digital electronic systems, for example, the design may be iterated through several levels of abstraction, ranging from, say, an algorithmic level where the basic algorithm to be developed in hardware is specified, to the detailed circuit level where concrete elements such as resistors and transistors are being dealt with. While both expert and novice designers are seen to work on design tasks in a top-down fashion, experts tend to pursue a breadth-first expansion, while novices tend to expand the design depth-first. The former method is generally considered to be better since it enables the designer to deal appropriately with interdependent goals and subgoals.

(4) The abstract knowledge that an experienced designer has concerning how to control and coordinate the generation of a good design may be referred to as a "design schema". This high-level schema is built up through experience and can be applied to a wide range of domain problems

having similar fundamental structures but which differ in their content.

(5) Schemas are essentially recognition devices that can be evoked by environmental or internal conditions, and which contain both conceptual knowledge and procedural knowledge. There will not only be a high level design schema but also a hierarchical organisation of embedded schemas at all levels of abstraction. Thus as a task is tackled, schemas at intermediate or low levels of abstraction will be instantiated by current conditions, thus evoking new information relevant to the problem at hand as well as procedures for action. Depending on the level of abstraction of the instantiated schema, the procedures evoked may be generalisable to a set of problems of similar type within the domain, or may be highly specific to a single problem. Expert engineering designers working in a familiar areas of their domain will possess a great quantity of well-organised domain-specific knowledge (schemas) whereas novices will have less knowledge that will be more poorly organised. Experts will therefore be better able to: retrieve known solutions or relevant algorithms, access background knowledge, understand concepts and evaluate alternatives.

(6) Expertise in enginering design also implies the ability to construct mental models to simulate the dynamic behaviour of aspects of the developing design. Such mental modelling (in addition to external modelling) would be particularly useful for comparing and evaluating the viability of alternative design options. Mental modelling may also be associated with the construction and use of mental images (though this is a highly contentious issue).

(7) An inexperienced designer will tend to rely on quite general,

domain-independent problem-solving strategies such as generate-and-test and hill climbing, or a rudimentary "divide and conquer" technique. With experience these general methods become transformed into a sophisticated and polished design schema. The experienced designer, however, may at times fall back on general problem-solving strategies, when, for example, he or she is confronted with an unfamiliar design task. Indeed, in some situations the use of general problem-solving methods may be the only sensible way to make any advancements at all.

Chapter 3

Investigating Engineering Design Processes: Some Theoretical and Methodological Considerations

Chapter 3. Investigating Engineering Design Processes: Some Theoretical and Methodological Considerations

Page

3.1. THEORETICAL AND METHODOLOGICAL CONSIDERATIONS

3.1.1. The distinction between declarative and procedural knowledge..... 71

3.1.2. Psychological methods of eliciting procedural knowledge.. 72

3.1.3. Techniques for analysing think-aloud protocols..... 80

3.1.4. Conclusions..... 85

3.1. THEORETICAL AND METHODOLOGICAL CONSIDERATIONS

3.1.1. THE DISTINCTION BETWEEN DECLARATIVE AND PROCEDURAL KNOWLEDGE

In attempting to understand the nature of cognitive processes in engineering design we are specifically focussing on what may be termed ***procedural knowledge,*** that is, the knowledge people have of "how to do things" (e.g. strategies and methods for designing). Procedural knowledge can be viewed as being distinct from ***declarative knowledge*** which is essentially factual knowledge. For example, the knowledge that an engineer possesses of design-oriented objects, concepts and their relationships can be thought of as being declarative knowledge. In computing parlance, procedural knowledge would exist as programs, comprising rules, heuristics or algorithms capable of processing information, whilst declarative knowledge would exist as a passive database of facts which could be manipulated by the programs. Although some researchers (e.g. Claxton, 1978) have argued that the procedural/declarative distinction can at times prove both strained and clumsy, others believe it to be a valuable theoretical device for understanding and modelling cognitive processes. Anderson (1983), for example, has used a variation of the basic distinction to devise an elegant cognitive theory of learning that can readily explain many psychological phenomena relating to the acquisition of mental and physical skills. More recently, Evans (1988) has relied on the distinction in addressing issues that relate to the elicitation of knowledge from experts in the context of training problem-solving and decision-making skills.

It is clear that in design situations the engineer will be bringing to bear both procedural and declarative knowledge in the attainment of design solutions. A fundamental feature of procedural knowledge which has been noted by cognitive psychologists is that it tends to be tacit, implicit or intuitive. What this means is that people can be seen to *use* such knowledge to perform complex mental and physical activities yet they are unable to explicitly ***report*** what knowledge is being brought to bear in carrying out these activities. This seems to be an access limitation; the knowledge is possessed yet it cannot be directly accessed for verbal description. There is considerable anecdotal evidence concerning this phenomenon and a number of psychological experiments have also demonstrated it (for a particularly interesting study see Berry and Broadbent, 1984).

Several major reasons why procedural knowledge should exhibit such inaccessibility have been proposed and are summarised by Cooke and McDonald (1986), though this theme will not be taken up here. It is important, however, to bear in mind that the limited ability that people have to provide introspective reports of procedural knowledge has important implications for the role that self-report methods can play in the study of cognitive processes. These implications are pursued further in the next section in which psychological methods that have customarily been used in the study of human procedural knowledge are discussed with the aim of determining which might be the more suitable for investigating design processes.

3.1.2. PSYCHOLOGICAL METHODS OF ELICITING PROCEDURAL KNOWLEDGE

In this section a discussion is provided of the main methods that have

been used by psychologists in the investigation of human procedural knowledge and an attempt is made to assess which of these might prove the most effective in the present context of understanding engineering design processes. Three broad classes of method will be dealt with, these being (a) the inferring of cognitive processes based on experimentally derived performance measures (b) introspective techniques such as acquiring self-reports of strategies and (c) observational methods such as obtaining and analysing behavioural protocols.

The conventional psychological method for studying procedural knowledge involves setting short, experimentally controlled laboratory tasks and obtaining simple performance measures (such as response latencies and errors) from which underlying processes are inferred theoretically. Whilst this laboratory method has been successfully employed in much cognitive research, it does appear to be of limited practical value for investigating problem solving and decision making processes arising in real-world domains. Evans (1988), for example, stresses that this methodology is not only slow, costly and difficult to implement but is simply not intended for the study of extended thought processes which presumably involve many stages of information processing before a solution or decision is reached. Evans also argues that this experimentally based methodology usually needs to be theory driven, that is, by the deduction from a stated theory of logically necessary predictions which can be tested by means of experimental manipulations. As such, the methodology is not suited to research in a novel area of psychological interest where theoretical models of processes are lacking and where a more inductive type of exploratory data gathering is initially more useful. This latter point seems particularly pertinent in

the context of the present research where the body of theoretical information relating to cognitive processes in engineering design appears to be almost non-existent (see chapter 2 for a review of the limited psychological literature on design).

Introspective techniques such as the acquisition of strategy reports have proved a popular method for investigating human procedural knowledge. Interviewing, for example, is the predominant method employed by contemporary "knowledge engineers" in the development of their expert system computer programs (e.g. the automated design systems discussed in section 1.1.1) which are aimed at replacing aspects of human expertise. Certain authors within the knowledge engineering community are, however, becoming much more critical of the use of self-report methods of knowledge elicitation (see particularly Kidd & Wellbank, 1984; Gammack and Young, 1984). The origin of this criticism appears to be a growing awareness of psychological evidence which indicates that much procedural knowledge is by nature tacit and implicit and therefore not accessible to verbal description (see section 3.1.1 above). Two very important implications concerning the utility of self-report techniques (either in cognitive research or in knowledge engineering) arise from a consideration of this access limitation.

The first implication is simply that self-report techniques may often be of almost zero utility in that people may be unable to give anywhere near a complete account of the strategies and procedures that they use when asked to do so. Support for this suggestion comes from psychological studies (e.g. Berry and Broadbent, 1984) as well as from the comments of knowledge engineers who note that experts working in certain domains have considerable difficulty in expressing the rules and

heuristics that they commonly use (see, for example, Burton & Schadbolt, 1988). The second and more worrying implication concerning the usefulness of self-reports techniques, however, is that they may actually be of negative utility in that subjects may provide inaccurate or misleading accounts of their procedural knowledge. Again there is evidence to support this suggestion - particularly in cases where reports have been requested subsequent to a subject's performance on a task. The indication is that people, when probed, may readily - and with total sincerity - give an account of how they did something which bears little or no resemblance to the processes that actually occurred (inferrable from more objective measures of behaviour). Research suggests that the retrospective descriptions that people provide of cognitive processes may actually be post-hoc rationalisations (e.g. Wason and Evans 1975) or theories (see particularly Nisbett & Wilson, 1977) devised to account for their behaviour. It is worth noting that certain psychologists (e.g. Morris, 1981) have argued that just because reports are sometimes inaccurate does not imply that this is always the case. Indeed, White (1988), in reviewing a number of papers relevant to this latter argument, reports several experiments that show verbal reports to be true strategy reports. However, as Evans (1989) points out, although accurate and complete self-reports of procedural knowledge may *sometimes* be obtained using introspective techniques, if no explicit *a priori* criteria exist as to when this will be so, then the practical value of these techniques remains doubtful.

In line with these foregoing considerations it clearly seems that introspective techniques are of little value to the researcher interested in investigating cognitive processes in design. This is not

to say, however, that such techniques should be totally discredited as a potential source of psychological data. The point is that although interview and self-report methods are fraught with pitfalls when it comes to investigating procedural knowledge, they do appear to be of value when one's objective is to gain an understanding of people's declarative knowledge. Evidently people have much greater access to factual and conceptual knowledge than to procedural knowledge, and are thereby more able able to verbally express details concerning, for example, the characteristic concepts within a domain of expertise, the nature of problems that are commonly dealt with and the criteria that determine the adequacy of solutions. Schadbolt and Burton (1989) have argued that structured interviewing techniques - where the knowledge engineer guides the inquiry in a formally planned manner - can be particularly useful in eliciting factual knowledge in an organised, rather than a haphazard, manner.

Although traditional experimental and introspective techniques appear to be of little value for the investigation of extended cognitive processes, the elicitation of such procedural knowledge seems to be more readily achieved by use of a third class of techniques. These techniques are characterised by the fact that they entail some form of observation of behaviour whilst subjects' are actually working on tasks peculiar to their domain of expertise. For example subjects could be monitored informally in their work environment whilst engaged in their professional activity. Alternatively a more formal observational study could be undertaken in a laboratory setting with the subject tackling problems analogous to those normally dealt with. In such a controlled laboratory situation it could be quite easy obtain a blow-by-blow record

(or protocol) of behaviour by videoing the subject's ongoing activities. This data could subsequently be analysed to identify the organisation and sequencing of activities, thereby enabling the nature of underlying thought processes to be inferred. If - as is often the case - one desires as complete an understanding of a person's procedural knowledge as possible, then it is usual to also ask a subjects to "think aloud" continuously whilst working on a particular task. The think-aloud (TA) trace or "verbal protocol" that is extracted can later be analysed in a similar way to other behavioral data to provide an insight into the procedural knowledge that the person used in tackling the task. This TA technique is arguably the most favourable method available for deriving an understanding of people's procedural knowledge in domains involving extended stages of thought (cf. Ericsson and Simon, 1980, 1984; Evans, 1988) though it is often most useful when supplemented with other behavioural traces such as video recordings of the subject's activities as described above.

The most powerful arguments concerning the value of TA protocols in cognitive research have been provided by Ericsson and Simon (1980, 1984). As a basis for their arguments these authors have formulated a comprehensive information processing model of cognition which deals with the nature of the mechanisms involved in producing verbal reports in response to a variety of differing verbalisation requirements (e.g. instructions to think aloud, requests to reply to concurrent probes or instructions to produce retrospective reports). In confronting the nature of the cognitive mechanisms underlying verbalisation, this model incorporates detailed explications of both (a) the processes which produce verbal reports as well as (b) the memory stores from which

verbalised information is derived. For example, these authors propose that concurrent verbal reports provide details of the information attended to or heeded by a subject at any particular point in time in that they directly reflect the current contents of short-term memory (STM) - a concept analogous to the notion of working memory. Under this proposal it is important in analysing TA verbalisations to take account of the fact that one is dealing with the "products" of cognitive processes rather than with self reported "descriptions" of cognitive processes. This means that underlying processes can only be *inferred* on the basis of careful and systematic interpretation of protocol content. Discussion of techniques for analyising TA protocols will be deferred until the next section as it is first important to outline some further aspects of Ericsson and Simon's theory.

In considering the nature of the processes involved in generating TA reports, Ericsson and Simon (1980, 1984) specifically differentiate three levels of verbalisation. Level 1 verbalisation involves direct articulation of information that when heeded exists in a verbal code. Level 2 verbalisation occurs when heeded information is encoded in a non-verbal form (e.g. images) which must be recoded into a verbal description. Level 3 verbalisation arises in situations where the task instructions require that the subject concurrently verbalise only selected types of heeded information or information that would not normally be attended to during performance on a task. It is clearly evident that each of these three levels of verbalisation is respectively associated with the occurrence of an increasing numbers of mediating processes between the heeding of an item of information and its externalisation. This "three-level" characterisation of verbalisation is

significant since it provides a foundation for formulating specific predictions regarding which TA procedures are most likely to be reactive (i.e. to change the actual cognitive processes being studied). The basic suggestion is that TA requirements may have problematic effects when they require a degree of processing additional to that involved in the direct articulation of verbally heeded information as in level 1 situations. Indeed after reviewing a large number of studies Ericsson and Simon conclude that thinking aloud does not affect the structure and course of processing for level 1 verbalisation while for level 2 verbalisation it may slightly decrease speed of performance. However, if subjects are instructed to concurrently verbalise only specific types of heeded information or information that would not normally be available (e.g. details of where they are looking), then normal processes appear to be more readily subject to distortion.

Another issue that Ericsson and Simon (1980, 1984) have addressed concerns the completeness of verbal reports - i.e. what information do subjects generally fail to report. With respect to concurrent verbalisation, Ericsson and Simon believe that there are two main types of omission. The first omissions reflect various intermediate products of recognition or retrieval processes. The intermediate products of such processes generally do not register in STM although the final products are usually reportable. For example, recognising of a familiar faces generally occurs directly without any apparent intermediate storage in STM of the specific facial features extracted from the stimuli and used for discrimination. The second type of omission concerns intermediate products of "automatic processes". Such processes appear to have become so well practiced as to render unnecessary any attention to intermediate

products which are therefore no longer deposited in STM. Automation of processes is generally regarded as being an intrinsic aspect of expertise, both in situations where skilled performance involves essentially perceptual-motor processes (as is the case in sports) and where it involves predominantly higher-level reasoning and problem solving processes (as is the case in domains such as medical diagnosis or business decision making).

3.1.3. TECHNIQUES FOR ANALYSING THINK-ALOUD PROTOCOLS

In light of the value that TA methods have for providing information concerning underlying procedural knowledge, it is clearly sensible to tackle the methodological issue of what sorts of techniques can best be employed for the actual analysis of verbal data. It should be noted that this present discussion focusses primarily on the general question of what makes an analysis technique good or bad and only a little consideration is given to the detailed question of what specific techniques might be suited to the analysis of designers' protocols. It is felt that this narrower issue might be dealt with more adequately after some design protocols have actually been obtained (i.e. in the context of the design studies that were undertaken within the present research progamme).

In considering techniques of protocol analysis, it is immediately worth stressing that certain methods appear to be of a very dubious nature and are perhaps best steered clear from if spurious interpretations of verbal data are to be avoided. For example asking subjects to help interpret their own protocols by giving retrospective comments on what they were doing (e.g. Rasmussen & Jensen, 1974) clearly renders the

analysis prone to the distorting effects of rationalisation and self-theorising discussed in section 3.1.1 above. For similar reasons the practice of getting experts within a professional domain to assist in the interpretation of other subjects' protocols is also highly suspect. The general recommendation, then, is that the researcher undertake the analysis of protocols without requesting subjects' assistance in the clarification or embellishment of the existing verbal trace. This said, however, it is obvious that even investigator-based interpretation of protocols could involve recourse to methods of doubtful validity. For example, analyses which simply take the form of informal discussions by the investigator of personally selected protocols can be criticised on the grounds that the interpretation of data may be highly subjective and the choice of protocols biased.

Whilst the sorts of technique outlined previously are obviously problematic, it is important to realise that even strictly formalised interpretation of protocol data can never be a totally objective or theoretically neutral endeavour. That is to say that protocol analysis will inevitably be influenced by (a) the actual theory of thinking - including the theory of verbalisation - which forms the overarching framework for the analysis (b) the knowledge of the actual problem domain possessed by the investigator and (c) the purpose for which the analysis is being undertaken (e.g. pure or applied objectives). These kinds of influences, however, appear as an integral aspect of all scientific research - and just because they are a little more so in the case of verbal protocol analysis does very little to discredit such methods. Newell and Simon's classic analyses of human problem solving, for example, clearly reflect their theoretical view of problem solving

as involving a search through a problem space from an initial state to a goal state via a set of intermediate states (see chapter 2 for detailed discussion of Newell and Simon's work). Under this theory, movement between states is achieved by the application of specific operators or "rules" defined by the problem domain itself. The final outcome of their type of analysis is usually a "Problem Behaviour Graph" (PBG) which depicts a trace of the state transitions made by the subject during the search of the problem space, including the erroneous paths taken and the backing up that is a characteristic feature of problem solving on these tasks. Clearly the extent to which Newell and Simon's theory was incomplete or inaccurate would have been revealed in the difficulty they experienced in encoding and structuring the protocol as a PBG. Weaknesses of the theory would also be evident to an independent observer in terms of the objective goodness of fit between the PBG and the subject's original verbal data.

One recommendation (see Ericsson & Simon, 1984) for attaining greater objectivity in the interpretation of protocols is for the researcher to define a finite set of categories for encoding information expressed within the verbalisations prior to any detailed analysis. Newell and Simon (1972), for example, frequently undertook preliminary analyses of both tasks and sample protocols in order to extract a vocabulary of objects, operators, constraints and goals which could then be used for the formal encoding of verbal data. Some attempts to further improve the objectivity of protocol analysis - as well as to facilitate the procedure - have resulted in the development of computer programs which automate or semi-automate the process (see, for example, Waterman & Newell, 1971, 1973; Bhasker & Simon, 1977). The major advantages of

automated protocol analysis appear to be threefold. Firstly, it makes clear exactly what assumptions have entered into the encoding scheme, since, of necessity, the encoding vocabulary - as well as the underlying theory on which it rests and the rules for its application - must be explicitly defined prior to the development of the analysis program. Secondly, the encoding scheme can be applied with total consistency such that reliability of encoding is perfect (i.e. given the same protocol the program will always reach the same interpretation). Thirdly, automating the process of protocol analysis clearly makes life considerably easier for the investigator since anyone who has undertaken the process will appreciate how tedious and time-consuming an endeavour it can be.

Encoding schemes for categorising protocol data can clearly be derived at a variety of levels of abstraction. Sometimes it might be appropriate to undertake statement-by-statement encodings of protocols which retain the detailed semantic content of the verbalised information. This technique seems to be most useful when the protocols are short and relate to problem solving on fairly well structured tasks. At other times, however, it is often more practicable to pursue some form of global encoding of aggregated sets of statements. For example, it is frequently possible to categorise sets of consecutive protocol segments as reflecting the occurrence of major processing steps (e.g Newell & Simon, 1972) or the execution of high-level heuristics (e.g. Pitt, 1983). Often in protocol based research, investigators strive for fairly intermediate levels of analysis (i.e not too detailed and not too abstract) though sometimes multiple levels of analysis are undertaken. This latter approach is perhaps typified by Rowe (1985), who, in an in-

depth study of over 1,000 problem solving protocols, encoded verbalisations using a three-tiered hierarchical classification scheme. Thus, for example, a low-level classification of a statement as reflecting "assessment of adequacy of progress" would also fall under the broader classification of "critical evaluation/judgement" (N.B. a 3-digit code was devised by Rowe to represent each classification). Generally, when low or intermediate levels of protocol analysis are undertaken, verbalisations are segmented on the basis of pause information (e.g, Newell & Simon, 1972; Byrne, 1977) or on the basis of a pre-defined time units such as three-second time intervals (e.g. Rowe, 1985).

Newell and Simon's PBG method appear to be most suited to the analysis of TA verbalisations derived from subjects attempting transformation problems, that is, puzzle-like tasks such as river crossing problems that involve making "moves" to transform one problem state into another. As will be discussed in the next chapter, PBG techniques have also been applied to the analysis of verbal protocols derived from subjects attempting space-planning tasks (see, for example, a study by Eastman, 1968, of people designing bathroom layouts). Space planning arises in many design domains including architecture and engineering and requires the location of a set of objects within a finite space such that specified constraints are met. Byrne (1977, 1981), however, has argued that many planning and design tasks do not generally produce such extensive backing up as is seen in transformation problems and are therefore more appropriately described by means of an "Object Transition Graph" (OTG). Time in the OTG flows downward and to the right, and each terminal branch is defined by an item that makes up a part of the final

solution. Byrne has successfully produced OTGs for verbalisations on several types of planning tasks, including the devising of a menu for a three course meal, and the designing of the decor for an apartment. In the example OTG that Byrne (1981) provides of a subject's attempt to design the interior of an apartment, it can be seen that rooms (i.e. abstract or high-level design objects) are treated by the subject one at a time and expanded hierarchically into their constituent parts (e.g. carpets, curtains, furnisher) in a highly recurrent and patterned manner. The emphasis of Byrne's OTG method is clearly on depicting the structure of the problem solving process, and indeed he has argued (Byrne, 1981) that the main purpose of protocol analysis should be to identify patterns and sequences rather than to simply look at the content of verbalisations. Whilst it is certain that the OTG method is of relevance to the study of problem solving processes in design, it is possible that the level of analysis may be too fine-grained for studying the sheer density of data that would be expected from any study of non-trivial design tasks (i.e most real-world design problems involve time scales many orders of magnitude greater than the tasks which Byrne studied). Further consideration of techniques of protocol analysis will be deferred until chapters 3 and 4 where detailed coverage is given of two studies of engineering design that were undertaken as part of the present research programme.

3.1.4. CONCLUSIONS

This chapter has primarily addressed the methodological issue of how best to derive an understanding of the cognitive processes involved in a real-world problem-solving domain such as engineering design. In

approaching this issue particular weight has been given to the theoretical distinction between declarative and procedural knowledge. Declarative knowledge relates to the factual understanding that people have of domain objects and concepts whilst procedural knowledge concerns the knowledge that people possess of methods and procedures. One important feature of procedural knowledge that has emerged from research is that it is frequently tacit and thereby of limited access for explicit verbal description. This characteristic clearly limits the role that self-report methodologies such as interviews can play in the study of cognitive processes (note that cognitive processes essentially equate with the notion of procedural knowledge). Indeed evidence suggests that people may either be unable to provide explicit descriptions of their strategies and methods or may alternatively hand provide accounts which are actually inaccurate or misleading. The traditional method of studying procedural knowledge, i.e. inferring underlying cognitive processes from experimentally derived measures of performance, also appears to be of limited practical value as far as the domain of design is concerned. This is primarily because design problems generally require many extended phases of processing before solutions are attained.

Arguably, the best methods for discovering people's procedural knowledge involve studying their behaviour whilst they are actually engaged in tasks that exercise their cognitive skills. Amongst such methods the derivation of a continuous "think-aloud" trace (or verbal protocol) whilst a subject works on a domain task has proved to be especially useful laboratory technique in the field of problem solving research. Supplementing the think-aloud protocol with a video recording of a

subject's activities is also a useful way to increase the density of behavioural observations. One particularly important issue that surrounds the use of the "think-aloud" technique relates to what methods should be used to analyses protocols that are obtained. The main object of any protocol analysis undertaken is to identify the organisation and sequencing of ongoing activities such that the nature of underlying cognitive processes may be inferred. Most protocol analyses, however, tend to be interpretive and heavily influenced by the particular theories of thinking and problem solving that appear applicable. What, though, seems to be of paramount importance in attaining as objective an interpretation of protocol data as possibleis that the researcher defines a finite set of categories for encoding the information content of verbalisations *prior* to any detailed analysis.

Chapter 4

Study One: Undergraduate Design Projects

Chapter 4. Study One: Undergraduate Design Projects

Page

4.1. INTRODUCTION TO THE PRESENT STUDY

4.1.1. Rationale behind the study..... 91

4.1.2. Aims of the study..... 93

4.2. METHOD

4.2.1. Subjects..... 97

4.2.2. Design..... 98

4.2.3. Tasks and Procedure..... 98

4.3. CODING AND STRUCTURING OF DESIGN PROTOCOLS

4.3.1. Introduction to the coding and structuring scheme..... 101

4.3.2. Goals..... 105

4.3.3. Constraints..... 117

4.3.4. Planned Methods and Methods Used..... 119

4.3.5. Decisions and Rationales..... 120

4.3.6. Difficulties..... 121

4.4. DISCUSSIONS OF INDIVIDUAL DESIGN BEHAVIOUR GRAPHS

4.4.1. Introduction to discussions..... 122

4.4.2. NB's Design Behaviour Graph..... 124

4.4.3. TS's Design Behaviour Graph..... 137

4.5. GENERAL DISCUSSION

4.5.1. Outline of the general discussion..... 152

4.5.2. The definition of constraints during the initial phase of the design process..... 153

4.5.3. Problem reduction strategies in engineering design..... 156

4.5.4. Strategies for managing difficulties and design failures.. 158

4.5.5. Problem models and solution models in engineering design.. 159

4.5.6. Technical models and social models in design..... 162

4.5.7. Decision-making processes in engineering design..... 165

4.5.8. Control of the design process: the design schema..... 169

4.6. CONCLUDING COMMENTS

4.6.1. Concluding summary of findings..... 184

4.6.2. Anticipating some criticisms of the study..... 187

4.1. INTRODUCTION TO THE PRESENT STUDY

4.1.1. RATIONALE BEHIND THE STUDY

The extreme scarcity of psychological studies undertaken to investigate the nature of cognitive processes in engineering design is a point that was stressed in the literature review presented in chapter 2. Indeed, the few reported studies relating to this domain which are known to the author (e.g. Eastman, 1969a) tend to be narrowly focussed on fairly trivial (and possibly unrepresentative) design tasks. It is, of course, likely that other psychological studies of engineering design have been reported but have not proved accessible to orthodox methods of literature searching because they reside in rather obscure publications specific to the engineering community (e.g. esoteric workshop and conference proceedings).

Reported studies relating to design domains other than engineering (i.e. architectural and software design) are somewhat more plentiful, and some of this research appears impressive because of its use of realistic design problems and systematic research methods such as the "think-aloud" technique discussed in section 3.1.2 (see particularly Jeffries et al., 1981; Adelson & Soloway, 1986). However, the true quality and scientific rigour of this latter work is often difficult to assess because in reporting such studies space considerations generally mean that the authors either (a) give very little indication of the techniques used for encoding and interpreting protocols and/or (b) present only a highly limited number of excerpts of illustrative protocol data and/or (c) provide discussions of findings which revolve

around only a single subject's protocol selected as a prototypical example of the full set of protocols.

All in all, then, the general picture is one of a whole domain of human endeavour which, from a psychological point of view, is under-researched, and where the limited work that has been done exists in an inadequately reported state. Still, though, from the studies that have been reported, it appears possible to formulate a basic psychological understanding of design, and a major aspect of the review chapter was the attempt to synthesise a theoretical framework (see section 2.4.1) for characterising both the nature of design tasks and the processes involved in tackling them. The strengths of this framework appear to be that (1) it falls quite naturally out of the mixed-bag of psychological literature that exists on design (2) it is largely compatible with theoretical notions and principles established in the wider psychological literature on thinking and problem-solving and (3) it provides a useful synthesis of existing knowledge on design processes that can motivate further exploratory research. One weaknesses of this framework, however, which stems from its basis on such a limited literature on the psychology of design, is that it is loosely specified - though this does not seem really to detract from its capacity to guide the search for more complete models of design processes. A second - and perhaps more serious - weakness of this framework, which arises from the fact that it reflects the findings of only a couple of studies on *engineering* design, is that it may fail to capture the processing that occurs within this latter domain.

Clearly, then, for the present author faced with the task of assisting with the development of the PEDA system, the overriding indication was

that some form of exploratory research needed first to be undertaken in order to more adequately specify the nature of engineering design processes. Only after this had been done was it felt that any attempt could be made to investigate ways of actually facilitating or improving design processes.

4.1.2 AIMS OF THE STUDY

To reiterate the point made in the previous section, the present study was planned primarily as an ***exploratory*** investigation of cognitive processes in engineering design. As such it was clearly intended to be something of an inductive type of data gathering exercise involving qualitative and descriptive methods for analysing subjects' design behaviour. No attempt was thus made to derive predictions from existing theoretical conceptualisations of design with the aim of testing such theories by conventional experimentation. Whilst this latter type of approach would, of course, be desirable, it is clearly difficult to implement in a domain where the existing body of theoretical knowledge is so impoverished - as is certainly the case for both design in general and engineering design in particular. Indeed it is questionable whether a conventional experimental approach involving manipulations of independent variables and the assessment of quantifiable affects on dependent variables using statistical methods is of much use in under-researched domains like the present one because such an approach is likely to be slow and costly in producing much in the way of new knowledge (see also section 3.1.2).

In keeping with the overriding aims of this research programme, the main objectives of the present study were to determine the nature of (1) any

common general strategy for engineering design (2) any individual differences in methods of working and (3) any characteristic biases or cognitive limitations constraining the attainment of optimal design solutions. Whilst - as already mentioned - the study was not aimed at testing a specific model of cognitive processes in design, this is not to say that theoretical notions introduced in the previous chapter were viewed as irrelevant to the investigation. The point is, that although existing theoretical ideas about design (summarised in section 2.4.1) were not felt to provide an adequate basis for the derivation of particularly worthwhile experimental hypotheses, many concepts were seen to be of potential use for the actual theoretical *interpretation* of the design data that would be derived in the present study.

One important theoretical proposal that emerged in the literature review was that experienced designers may possess generalisable abstract knowledge concerning how to produce good designs in the form of a "design schema" (see Jeffries et al, 1981 and section 2.3.2). This design schema - which would be built up through experience - could be applied to a wide range of domain problems having similar fundamental structures but which differed in terms of their content. When applied in a design situation, then, the design schema would control and coordinate (a) the decomposition of the design problem into subproblems (b) the order in which these subproblems were worked on and (c) the search, retrieval and evaluation of solutions. In interpreting design data, then, it would, seem important to assess whether subjects' behaviour can be viewed as reflecting some form of general-purpose schema or strategy for design. The related issue, of course, is whether the strategy can be seen to be similar across different engineers or whether individual

differences in behaviour preclude this possibility.

A second theoretical idea of interest is that designers may possess schemas at many different levels of abstraction (cf. Rumelhart's, 1980, view of the organisation of human knowledge) such that some schemas might be specific to the solution of quite "concrete" technical problems. In the previous chapter, for example, it certainly appeared to emerge that skill in problem solving within semantically rich domains reflects the development of a large number of domain-specific schemas at multiple levels of abstraction (see section 2.2.5). A final notion concerning enginering design processes is that skill within the domain seems to imply the ability to construct and manipulate "mental models" to reason about the developing design as well as to simulate the dynamic behaviour of aspects of the solution (see section 2.3.3). Such mental modelling - in addition to external modelling - would be particularly useful for comparing and evaluating the viability of alternative design options. Again, then, this contemporary theoretical notion of mental modelling appears relevant to the interpretation of design behaviour.

Before moving on to a detailed consideration of the present investigation it first seems important to make a few comments concerning the subjects that were studied (i.e. undergraduate engineers) as well as the research methodology that was adopted in the investigation. The factors leading to the choice of undergraduate designers were simply that they were available, accessible and recruitable at a point in the research programme when the time consuming process of forging links with locally based design companies - whose designers might be candidates for psychological study - had yet to be pursued. A population of final year students were soon pinpointed as being particularly suitable for the

students were soon pinpointed as being particularly suitable for the present study for three main reasons. Firstly, these students were generally just ***beginning*** to embark on lengthy design and development projects - which was pleasing from a psychological perspective since it was desirable to follow the evolution of a design solution from the foremost stages of its conceptualisation. Secondly, since success on the projects was necessary to obtain the honours degree (i.e. the task was important to the subjects) the study would be focussing on generally highly motivated subjects. Thirdly, the existence of certain constraints, such as time (the projects had to be completed by the Easter of the final year) and cost (a budget of seventy pounds was set for buying components and tools), suggested that these undergraduate design projects were similar in certain essential respects to real-world design tasks in professional settings.

Turning, now, toward the choice of a research methodology for use in this study it is was thought essential to acquire information from the engineers *concurrent* to any episodes of design activity that they were engaged in. This emphasis on concurrent reports is clearly in line with the recommendations detailed in section 3.1.2 concerning techniques for eliciting procedural knowledge. Whilst the possibility of deriving conventional "think-aloud" protocols was obviously out of the question in the present study because of the long-term nature of the design projects being undertaken, some similar form of behavioural tracing was viewed as desirable. The primary method of knowledge elicitation that was finally adopted, then, required subjects to keep a "project diary" concurrent to any design work that was being undertaken. It was realised that this cognitive-diary method, whilst having the potential to

minimise some of the problems associated with after-the-event reports, could still be partially susceptible to (a) the intrusion of rationalisations and (b) omissions of information resulting from the time delay between heeding such information and externalising it on paper (this problem will be returned to in section 4.6.2). Still, however, the use of this diary method seemed to be the most practicable solution to the problem of obtaining information from subjects over an extended period of time. In addition to using this diary technique, subjects were also required periodically to attend interview sessions. The main purpose of these interviews was to obtain (1) any needed clarification of diary entries that were difficult to understand and (2) additional information concerning the subjects current goals and plans. Further details of both of these methodologies are provided below.

4.2. METHOD

4.2.1. SUBJECTS

Eight final-year undergraduates (seven male and one female) who were undertaking BSc honours degrees in electrical and electronic engineering at Plymouth Polytechnic served as subjects in this investigation. These eight undergraduates were approached for possible recruitment in the study on the basis of random selection from the larger group of potential subjects. They were given a briefing as to the intended format of the study and were told that twenty five pounds would be awarded for their time an effort. All eight students expressed a willingness to participate. It is worth noting that these selected students all had prior experience of design work which they had obtained as a result of

coursework assignments and a period of placement in a professional/industrial setting.

It should be noted that in the following discussions all subjects who participated in the present study are referred to with fictitious initials so as to ensure their anonymity. (A similar strategy has also been adopted in referring to all company names, supervisor names and the like).

4.2.2. DESIGN

As has already been mentioned, the investigation was felt to be something of an exploratory enterprise aimed at gaining some general insights into the nature of the engineering design process. Techniques of knowledge elicitation that were used included project diaries and interview methods and it was presupposed that essentially qualitative and descriptive methods of data analysis would be used in examining the resultant verbal material. No attempt was therefore made to impose any conventional experimental manipulations upon the selected sample of engineers.

4.2.3. TASKS AND PROCEDURE

Subjects were undertaking individual long-term projects which required the design (and often the development) of artifacts involving electronics hardware - and in a few cases combined hardware and software systems. Of the eight subjects involved in the study, five (i.e. NB, JC, SC, AD, NO) were attempting projects of their own derivation, two (i.e. TS and SM) were tackling a design task that had been provided by an

supervisor. SM was actually dropped from the study soon after its commencement since it emerged that he was engaged in a totally software-oriented project which was clearly of limited interest for present research which was looking at engineering design processes.

Subsequent to their recruitment, subjects were each given a pre-formated diary, together with written instructions (see appendix A) which emphasised the importance of making diary entries concurrent to any project work that was being undertaken. Additionally the instructions presented a detailed indication of the kinds of design activity that the investigator wanted the subject to report on. In this latter respect, then, weekly diary entries were requested under five preset headings which were aimed at monitoring different types of design behaviour. These five activity headings are listed below. (Note that the questions in parentheses were included in the subjects' instructions to provide an indication of the information that was being sought).

Activities.

(What are you working on and spending time on?)

Aims/Objectives.

(What are you trying to achieve both in the long term and at present?)

Problems/Constraints.

(What problems are you having with your design? What factors are constraining what you are attempting to do? What is holding you back from making progress?)

Decisions.

(What decisions are you making? Are you having to make choices as to which design option to pursue? Are you ever deciding to return to previously considered design options?)

Methods.

(What knowledge - for example, of methodologies or techniques - are you bringing to bear when you are faced with problems or have to make decisions?)

The diaries, also included a ***General Comments*** heading to allow subjects to recount any information that was not subsumed under the main activity headings. Throughout the study diary entries were collected from subjects on a weekly basis at the end of every Tuesday (this being an allocated "project day" during which subjects carried out most of their design and development work). Picking up the diaries at the end of each project Tuesday clearly enforced subjects to record their project activities either during or soon after the actual occurrence of the activities.

Occasional interviews were also undertaken with subjects in order to supplement the diary data and were both fairly informal and of short duration (usually lasting about twenty minutes). The author was assisted in these interview sessions by an experienced electronic engineer (referred to as DS in the interview transcripts).

4.3. CODING AND STRUCTURING OF DESIGN PROTOCOLS

4.3.1. INTRODUCTION TO THE CODING AND STRUCTURING SCHEME

During the course of this study, considerable amounts of verbal data were collected from each subject, all of which were transcribed verbatim into a type-written format (see appendix A for a sample data set). Subsequent to this data collection phase of the study it was clearly necessary to look into possible ways of producing a structured encoding of each subject's data (see section 3.1.3 for a review of several techniques of protocol analysis). In was thought essential that any analysis scheme that was adopted in the present context should enable (1) the reduction of large quantity of verbal material (some of which was repetitive or irrelevant to design) into a more manageable data set (2) the consistent categorisation of this reduced set of verbalisations and (3) the structuring of categorised verbalisations into some readable and coherent manner that made manifest pertinent features of the ongoing process of design.

One technique which seemed to be relevant to the analysis of design protocols was the Object Transition Graph (OTG) method developed by Byrne (1977, 1983) which has already been outlined in section 3.1.3. It may be recalled that this method was devised by Byrne in order to encode and structure think-aloud protocols derived from subjects attempting planning and design tasks (e.g. designing the complete interior decor for an apartment). Within an OTG, verbalisations are structured as discrete but interconnected information units which are depicted with respect to the time course of their productions by the subject (note

that time in the OTG flows downwards and to the right across the page). In a prototypical OTG presented by Byrne (1983), for example, the subject's initial high-level goal (i.e relating to the design of a particular room) was shown to be expanded into its constituent subgoals (e.g. relating to the selection of curtains, carpets, furnisher etc.) for which low-level solutions (e.g. specific styles of carpets) were then attained in a sequential, depth-first manner.

To date, the OTG technique has only been applied in the analysis of verbal protocols derived from subjects attempting tasks of a relatively trivial and short-term nature which require no more than a few minutes of *mental* problem-solving. It was therefore felt necessary to develop a somewhat different analysis and structuring technique for the present application. Many real-world design problems (including the ones being tackled by subjects in the present study) appear to differ from the tasks set by Byrne in that (1) they are much less well-defined (2) they require the processing of larger quantities and varieties of technical information (3) they involve processes which extend over considerably longer periods of time and which lead to many external activities (e.g. the modelling of design concepts on paper) rather than just mental manipulations of information. The OTG approach, then, appears to be of limited use when it comes to representing the rich and detailed varieties of information that appear to result from the pursuit of long-term design goals. Additionally, the technique does not seem to be suitable for capturing the way in which difficulties arising during the pursual of design solutions might actually feed back to the planning level, resulting, for example, in old goals being revised or abandoned, or new ones being generated. In this latter respect a cursory analysis

of the present data set revealed that subjects were indeed periodically undertaking such reformulations at the planning level of their design work. From a more positive perspective, however, the OTG technique generally appears useful for depicting the hierarchical goal structure which emerges as a common feature of planning and design processes as well as for representing the steady accretion of problem solutions as they occur over time. Indeed it was these latter features of the technique that were particularly attractive in the present application. Generally, then, an attempt was made to retain the basic feel of the OTG in structuring the present design data but with many differences to the existing technique being introduced.

In brief outline the approach that was eventually developed for the analysis of the present design protocols may be viewed as a three stage process. The first stage involved the derivation of (a) a categorisation scheme to encode each subject's design-relevant verbalisations using a finite set of labels and (b) a structuring scheme for representing these encoded data in a graphical manner somewhat akin to an OTG. The set of categories that was finally devised for encoding the information content of verbalisations was as follows:

(1) Goal

(2) Constraint

(3) Planned Method

(4) Method Used

(5) Decision

(6) Rationale

(7) Difficulty

(8) Idea

(9) Comment

(10) General Comment

It should be noted that the categorisation of the types of goals that were motivating subjects' behaviour was seen to be particularly important in present analysis, and five separate subcategories of goals were therefore isolated (see section 4.3.2 below).

The second stage of the analysis involved the actual selection, categorisation and structuring of subjects' design data into what have come to be termed Design Behaviour Graphs (DBGs). The diary and interview data of *four* of the original seven subjects were structured into DBGs (presented in appendix A). For these four subjects (NB, MM, AD and TS) data were selected for encoding primarily from the diary protocols, with only minimal recourse being made to the interview transcripts which tended to be of a much more retrospective nature. As can be seen in the four DBGs presented in appendix A, coded verbalisations were structured with respect to time in a manner that clearly revealed both the reduction of goals into subgoals and the ongoing progress that was being made in pursuing goals and subgoals. For example, progress in pursuing a design goal may have involved (1) the formulation of a planned method to attain the goal (2) the occurrence of a difficulty during the execution of the method and (3) the setting of subgoals for resolving the problem at hand before (4) the method could be finally executed to completion and (5) the design goal attained. The two formative stages of the present protocol analysis are described in detail below (see sections 4.3.2 to 4.3.6). In these subsections precise details are given of both the categorisation system that was devised for classifying subjects' verbalisations and also the manner in which these

classified verbalisations were represented graphically. The third and final stage of analysis involved undertaking written discussions of individual DBGs in order to draw out further the nature of the engineering design processes (see sections 4.4.2 to 4.4.3 for discussions of two selected DBGs).

A final point that is worth commenting on in relation to the present analysis scheme is the limited number of DBGs (i.e only four) that were actually produced in the study. This reflects, firstly the time consuming nature of their production, and secondly the fact that certain of the projects were primarily concerned with ***implementation*** work rather than with ***design*** work - which was clearly the desired focus of the investigation.

4.3.2. GOALS

Within the context of the present analysis scheme, then, the term ***goal*** has been used in its usual psychological sense to refer to any desired end or future state, the attainment of which is motivating or directing a subject's behaviour. Under this characterisation, goals are represented mentally - though not necessarily in a form that is consciously accessible. As was explained in section 1.2.2, the overriding goal motivating someone engaged in an engineering design project would usually be to attain a ***description*** or a ***model*** of a desired "artifact" at some specified level of design detail.

Looking objectively at this overriding goal state of a design project, it is useful to view the goal as existing at a point in a multidimensional search space, the dimensions of which are defined in

terms of ***requirements*** that the desired artifact must satisfy. Indeed informal evidence suggests that design specifications tend to be fairly rich in goal-oriented requirement information in that details are given of (1) the artifact's functional requirements (i.e. what it must do) (2) the artifact's performance, resource usage, structural and aesthetic requirements (e.g. its efficiency, reliability, testability, maintainability, simplicity, manufacturability, style, space, power consumption, speed and the like) and (3) requirements relating to the "process" of design itself (e.g. the artifact description might have to be realised within a certain time, under an allocated budget and using stipulated design tools). Additional requirements - particularly implicit ones not provided in the problem specification - are likely to emerge as the engineer defines the problem and works toward a solution. Requirements clearly serve to restrict the possible search space for a desired artifact description, and many authors therefore refer to requirements as "constraints" (see section 4.3.3 below).

Normally, then, a designer's ***initial,*** overriding design goal - represented within his or her mental problem space - will be based upon information about requirements and constraints actually contained within the given design specification. As has already been stated above, the overriding, long-term goal in a design situation usually relates to the attainment of an artifact description meeting a multiplicity of requirements. Clearly, though, the highly multi-faceted nature of such a long-term goal invariably means that the engineer will have to formulate many more readily attainable subgoals on route to the attainment of the overriding design goal. Many of the subjects' verbalisations that have been encoded as goals within the DBGs appeared to be in the form of

statements of intention involving constructions such as "I aim ..." or "I'm trying ...". It appears reasonable, however, to use such intentional phrases as being indicative of subject's underlying goals (cf. Newell & Simon, 1973).

CATEGORIES OF GOALS

A variety of goals were identified when analysing protocol material derived from the eight subjects. The goals appeared to be clearly divisible into the five distinct categories listed in figure 4.1. This figure also shows the abbreviations that were adopted in the DBGs to indicate the membership of a goal to one of the five categories.

Goal categories	Symbol and abbreviation indicating category membership
Design/Implementation Goals	numbered circle
Knowledge Acquisition Goals	KAG
Option Evaluation Goals	OEG
Project Documentation Goals	PDG
Time Schedule Goals	TSG

Figure 4.1: Categories into which goals were divisible, and abbreviations used in the DBGs to indicate the membership of a goal to a category.

To determine the membership of a goal to a category, strict *a priori* criteria were adopted which are described below:

(1) A goal was categorised as a design/implementation goal if the statement of aim or purpose related to either (a) the attainment of a description or model (i.e. a design) of the desired artifact or a module of the desired artifact or (b) the attainment of a verification, optimisation or modification of an artifact description or (c) the attainment of an implementation (as hardware or software) of this artifact description or (d) the attainment of a test, fault-diagnosis or modification of an implementation. An example of a design/implementation goal is:

> "[I aim] to fill out the 'I/O port' and 'Addcount' blocks with gate-array logic components, and to make some initial designs of the control logic +++."
> (GOAL 1.2; TS's DBG)

(2) A goal was a knowledge acquisition goal if the statement of aim or purpose related to the acquisition of knowledge not already possessed that could be of use to progress further on the design task. An example of a knowledge acquisition goal is:

> "Read Apollo CAD system handbook for wiring the block diagrams together, to overcome wiring busses problem which I have encountered."
> (KAG-4; TS's DBG)

(3) A goal was an option evaluation goal if the statement of aim or purpose related to the evaluation of alternative design or implementation options. An example of an option evaluation goal is:

> "Evaluate counter designs used on commercial components to determine "best design" for use on the DMA unit as a 32- bit synchronous counter."
> (OEG-2; TS's DBG)

(4) A goal was a project documentation goal if the statement of aim or purpose related to the making of a documentary record of the design project or some aspect of it. An example of a project documentation goal

is:

"Intend to write report on DMA unit with more technical detail and state the aims of the project"
(PDG-1.2; TS's DBG)

(5) A goal was a time schedule goal if the statement of aim or purpose related to the completion of a piece of work by a specific deadline. An example of a time schedule goal is:

"Aim to get counter designed by next week - 24th Feb"
(TSG-2; TS's DBG)

STATED AND INFERRED GOALS

Many of the goals in the DBGs were verbatim repetitions of statements made by subjects during their design work and were generally written in the Aims/Objectives section of their diaries. Certain design/implementation goals, knowledge acquisition goals and option evaluation goals, however, were inferred by the experimenter after very careful analysis of protocol material, and these may be recognised in the DBGs by their lack of bounding quotation marks. It was felt to be necessary to infer that certain goals were motivating a subject's design behaviour in order to lend coherence to the design sequence as depicted in the DBG. Goals were inferred only within the carefully analysed *context* of subject's ongoing activities. Thus, for example, if the subject had stated in his Activities section that he had been engaged in drawing a block diagram to depict the high-level modules of the device being designed, it would seem reasonable to infer that at some point the subject had set - not necessarily consciously - a goal "to produce a modular block diagram for device" and that this goal was now being pursued. The legitimacy of inferring the existence of unstated goals is clearly based on the assumption that problem-solving activities are

aimed toward achieving some purpose (i.e. intelligent behaviours are goal directed).

NUMBERING OF DESIGN/IMPLEMENTATION GOALS

It was felt to be essential to number design/implementation goals within the DBG such that they could be referred to in discussion. Initially it was thought that it might be sufficient to number these goals in a simple incremental manner that reflected the time of their being set by the subject. After a more careful analysis of each design protocol, however, it was thought that such a simple numbering scheme would not do justice to the complex interwoven arrangement of nested and interrelated design/implementation goals that was arising during the design process. What was particularly evident in the protocols was that subjects were using a ***subgoaling*** strategy (i.e. a problem reduction approach) to break down large design goals into more manageable sub-goals. Often, then, a goal of the form "design functional module M1" would be decomposed into a number of minimally interacting subgoals relating to the design of sub-modules of this higher level module. The attainment of the high-level goal would then be a direct consequence of the attainment of the set of subgoals. Often, too, it was evident in the protocols that blocking conditions would sometimes prevent the subject from pursuing a goal directly. In such cases the subjects were usually seen to set a subgoal which was concerned with removal of these blocking conditions. When the subgoal was attained, activity aimed at the achievement of the original goal would be resumed. Further discussion of these observed design strategies will be provided later in this chapter.

Having observed that subgoaling was prevalent in the engineer's design

behaviour it was felt that a hierarchical numbering system would be appropriate as a way of depicting the nested nature of design/implementation goals as well as satisfying he need to reference such goals for discussion purposes.

It is important to repeat that within the goal categorisation scheme that has been outlined, goals that related to the validation, optimisation or modification of designs or to the testing, fault diagnosis and redesign of implementations were subsumed under the generic label of design/implementation goals. Such goals have been numbered in the DBGs as being offspring goals of "true" higher-level design/implementation goals of the form "design module MI" or "implement module MI".

NUMBERING OF KNOWLEDGE ACQUISITION GOALS, OPTION EVALUATION GOALS AND PROJECT DOCUMENTATION GOALS

On a few occasions subjects were seen to use a subgoaling strategy for managing knowledge acquisition, option evaluation and project documentation goals. Again, then, a hierarchical numbering system was adopted for referencing goals of these types.

CATEGORIES OF DESIGN/IMPLEMENTATION GOALS

Design/implementation goals stated by subjects were clearly categorisable on the basis of the time course of their pursuit and attainment. Thus design/implementation goals were either short-term goals, intermediate-term goals or long-term goals. Within the DBGs any goal belonging to one of the latter two categories was appropriately labelled as an INTERMEDIATE-TERM GOAL or a LONG-TERM GOAL, and by

default, then, an unlabeled goal was a short-term goal.

Some examples of intermediate-term and long-term goals from subject's protocols were NB's goal "[Try] to get this block diagram into something that can be implemented using electronic components" (this was a long-term goal) and MM's goal "+++ design the "black box" to send the current into the eye and also demodulate the resulting signal +++" (this was an intermediate-term goal). As can be discerned from these examples, stated goals of these types tended to be descriptions of large-scale design and development objectives, particularly the design or implementation of major hardware or software modules of the device. As such, then, long- and intermediate-term goals often - though not always - existed as the ancestral goals of whole trees of relatively short-term subgoals, sub-subgoals and so on. The overriding goal of a design project would clearly be expected to be a long-term goal of the form "design device d" and indeed some subjects were seen to provide such an all encompassing goal statement. It is important to be aware, however, that many of the long-term and intermediate-term goals presented in the DBGs were inferred by the experimenter. While subjects seemed readily to articulate short-term goals, goals of a more broad and encompassing nature were generally left unstated. It seems plausible that such goals were left unverbalised by subjects simply because when they came to writing their diary entries they were preoccupied by the pursuit of short-term subgoals at the lower levels of the goal tree.

On occasions during the design process a goal would be stated by a subject at a time when he or she clearly had no intention of immediately pursuing it. A goal of this type - set in the present but whose pursual was intended at a future date - was denoted with the label {FUTURE GOAL:

IMMEDIATE PURSUAL NOT INTENDED}.

IMMEDIATE PURSUAL OF DESIGN/IMPLEMENTATION GOALS

Any long-term or intermediate-term design/implementation goal that was placed in the DBG and which was observed to have been immediately pursued by the subject was followed by the label {ONGOING PURSUAL}. As has been mentioned earlier, subjects were generally seen to reduce long- or intermediate-term goals into several short-term subgoals. In then pursuing these subgoals - whether this was done in parallel or serially - the subject would still clearly be engaged in the ongoing pursual of the parent goal in the goal tree.

Analysis of protocol material revealed that it was quite often the case that a parent goal's full compliment of subgoal siblings (i.e. offspring subgoals existing at the same level in the numbering hierarchy) would not be articulated by the subject in one go. Instead these subgoals would be mentioned at various times over the ensuing weeks, or, as was more likely, could only be inferred from the subject's account of the activities that were being carried out during this time. It sometimes also occurred that a subgoal would arise solely as a consequence of prior work that had been directed toward some pursued (though not necessarily attained or attainable) sibling subgoal. For example a subgoal of the form "design submodule SM1" might be followed by the sibling subgoal "redesign submodule SM1" if the initial design solution was considered unsatisfactory.

When it was clear that a subject had directly pursued a short-term design/implementation goal, the label {GOAL PURSUED} was placed following the statement of the goal in the DBG. When a short-term goal

"N" was eventually attained the label {GOAL "N" ATTAINED} was used to indicate this. The attainment of long- and intermediate-term goals was not indicated within the DBG.

When the actual pursual of a short-term design/implementation goal had been initiated and this pursual of the goal was observed to be stretched over two or more weeks of the project, then the the label {PROGRESS ON GOAL "N"} preceded any statements relating to progress that was being made toward the attainment of this goal. In addition arrowed lines were drawn on the DBG to depict the connectivity between the statement of the short-term goal and statements relating to progress.

Connectivity was never shown between child and ancestor design/implementation goals of the goal tree since it was presumed that the hierarchical numbering system that had been adopted was sufficient to reveal the relationships of goals to one another. Connectivity was also not shown between sibling goals.

DEFFERED PURSUAL OF DESIGN/IMPLEMEATATION GOALS

When the pursual of a design/implementation goal was seen to be temporarily deferred, although its immediate pursual had been intended, the label {GOAL NOT IMMEDIATELY PURSUED} denoted this fact. The decision to defer the pursuit of a goal, if not explicitly remarked on by the subject, was generally inferrable from the omission of any verbalisations relating to its perusal and/or the fact that at a later date pursual of it was decisively initiated (often subsequent to a restatement of the original goal).

If a deferred design/implementation goal was restated by the subject,

then the label {GOAL "N" RESTATED} preceded this restatement in the DBG, and an arrowed line indicated the connection between the two goals. If, however, a restatement of the original deferred goal was not made but pursual of it had been clearly initiated, then the label {PROGRESS ON GOAL "N"} preceded any statements relating to progress that was being made toward the attainment of the goal and the connectivity between these progress statements and the goal statement was depicted in the DBG.

Arrowed lines were also used to connect a "future" goal with its restated version (or the inferred version of this goal) when its pursual was eventually undertaken.

CATEGORIES OF KNOWLEDGE ACQUISITION GOALS, OPTION EVALUATION GOALS AND PROJECT DOCUMENTATION GOALS

Knowledge acquisition goals and option evaluation goals were always seen to be short-term goals, that is, the time course of their pursual and attainment was fairly minimal relative to the time dedicated toward the attainment on most design/implementation goals. Knowledge acquisition goals and option evaluation goals were also always seen to exist as subgoals of design/implementation goals. The relatedness of knowledge acquisition and option evaluation goals to their parent design/implementation goals was depicted in the DBGs either by the goals appearing in the graphs directly under their parent design/implementation goal or by the use of arrowed lines to connect the two areas of activity.

The project documentation goals that emerged in the protocols were seen to range fully between the short-term and the long-term. These were

depicted in the DBGs as goals unrelated to any of the other goal types (i.e. project documentation was viewed as an end in itself rather than as a means of attaining a design).

IMMEDIATE AND DEFERRED PURSUAL OF KNOWLEDGE ACQUISITION GOALS, OPTION EVALUATION GOALS AND PROJECT DOCUMENTATION GOALS

Generally when a knowledge acquisition goal or an option evaluation goal was mentioned by the subject, its pursuit was then immediate. Project documentation goals, however, were frequently liable to postponement after having been initially set by the subject.

The labels that were adopted in the DBGs to denote the pursual, postponement and attainment of goals of these types were the same as those used for design/implementation goals, except that the labels were written in lower case letters for the former as opposed to upper case for the latter.

TIME SCHEDULE GOALS

It would seem predictable that a subject engaged in a complex and multifaceted endeavour such as design and development would be concerned with allocating time scales to the various subgoals of the project. It would also seem likely that externally imposed deadlines would be set by supervisors and collaborating companies that would be influential in determining the design/implementation goals that the subject pursued. It was not surprising, then, to discover within the design protocols comments that related to such time scheduling and deadline setting. The term ***time schedule goal*** was used to classify statements of aim or purpose that concerned the completion of pieces of work by a particular

deadlines.

While some time schedule goals were clearly rigid such as the date that the project had to be completed by, the majority of time schedule goals - being self-imposed by the subject - were flexible and extendible. Strictly speaking a time schedule goal was set by the subject as a constraint on the pursual of one of the other four goal types. As such, a time schedule goal was not really pursued but was either attained (i.e. successful completion of the task on or before the deadline), or not attained (i.e. failure to complete the task by the deadline).

Failures to attain time schedule goals were seen to arise for a variety of resons including, for example, the misjudgement of the time required to achieve a task or the forgetting that a deadline had ever been imposed. In the former cases the time schedule goal that was originally set might be revised (i.e. a new one set) or a time limit might be considered ill-conceived and the task left to be completed "as soon as possible". It should be realised that a subject might abandon or revise a time schedule goal without making any explicit mention of this fact. Similarly if a time schedule goal was not mentioned again it was impossible to infer that it had been forgotten since it was possible that the subject had instead merely failed to verbalise about this particular goal.

4.3.3. CONSTRAINTS

In the context of design, the term ***constraints*** is commonly used to refer to the "requirements" that the desired artifact must satisfy (see, for example, Sussman & Steele, 1980; Steinberg, 1987). As was noted in

section 4.3.2 design may be viewed as a process of searching for an artifact description that resides at a point in a multidimensional problem space whose dimensions are comprised of different types of constraints (i.e. relating to functionality, performance, resources usage, aesthetics, cost and so on). It is clear, then, that in the present chapter the terms "constraints" and "requirements" are being treated as synonymous and interchangeable - though frequently functional constraints will be referred to by the term "functional requirements" to distinguish them from the many other types of constraints.

In practice, it appears that many constraints are explicit in the problem specification given to the designer and many more become manifest during ensuing client-designer dialogues or emerge as a result of work on developing solutions to the problem (see, for example, Malhotra et al, 1980). Informal evidence (see, for example, Simon, 1973; Mostow, 1985) also suggests that constraints may be used as "criteria" for choosing between design alternatives and for evaluating design solutions

The verbal data derived in the present study abounded with the mention of constraints pertinent to the ongoing design or its realisation. Often specific constraints would be outlined soon after the setting of a Design/Implementation goal or would be mentioned in association with a decision and/or rationale. The label CONSTRAINT(S) was used in the DBG to indicate any statements a subject made that related to the existence of some constraint(s) or functional requirement(s). The statement below is an example of a constraint-oriented verbalisation taken from NB's design diary:

CONSTRAINT
"The RESET pin must be held low for at least 20ms on power up, and should also return low in case of momentary supply voltage change, to prevent data corruption or spurious operation." (Diary 5/11 to 11/11)

4.3.4. PLANNED METHODS AND METHODS USED

Methods are essentially a set of actions that are directed toward the attainment of a desired goal state. In the design protocols the method that a subject was actually seen to use in pursuing a goal was sometimes seen to be quite different to that which had been originally planned. Within the DBGs, then, the label PLANNED METHOD was used to indicate any statements that related to a set of actions that the engineer was "intending" to use to attain a particular goal. On the other hand the subject's retrospective account of the method that had been adopted "in actuality" was preceded by the label METHOD USED when this account was placed in the DBG. It should be noted that statements of planned methods were frequently seen to be partially comprised of several subgoals that needed to be pursue, as well as the ordering in which the pursual was to be undertaken. Examples of statements provided by NB which reflect respectively planned methods and methods used are provided below:

PLANNED METHOD
"+++ define the keyboard functions required for the device and understand the relationships between them. +++ represent each function as a series of sub-functions, in order to gain a minimal sub-function set capable of performing functions in combination. +++ use sub-function set as the basis for key functions, and design a keyboard to incorporate all the key functions." (Diary 29/10 to 4/11)

METHOD USED
"Catalogues were consulted to find the capacities and prices of ROM, and the one which had sufficient capacity at the lowest price was chosen. The requirement for RAM was assessed,

and it was found that the microprocessor has enough built in RAM to fulfill this requirement." (Diary 12/11 to 18/11)

4.3.5. DECISIONS AND RATIONALES

The term "decision making" is customarily used to refer to the the process of choosing one particular option from a set of alternatives that have been generated. Within the context of engineering design, options could, for example, be particular hardware components, software algorithms, tools or design methods. The product of decision making, then, is clearly the decision or choice itself (e.g. "use object O", or "employ method M") which is then usually - though not necessarily - followed up by taking the action dictated by the decision.

Within the DBGs the label DECISION preceded any statements provided by a subject that (a) related to the "process" of selecting an option from a set of alternatives (such statements were quite rare) and/or (b) described some option that had been chosen - not necessarily from an overtly defined set of alternatives (such statements were much more common). With respect to the latter type of statements, it was difficult to assess whether in fact options had actually been selected from competing alternatives at all (see section 4.5.7 for further discussion of this point). Still, however, it was decided to use the label DECISION to categorise such verbalisations. Additionally, when subjects stated an option that they had chosen they frequently provided a brief rationale or justification as to why that option had been selected. Such justificatory statements were preceded in the DBG by the label RATIONALE. Often, the explication of both a decision and a rationale would be provided by the subject in an interconnected set of statements.

When this was the case it seemed simpler to present the complete set of comments within the DBG and to head these with the label DECISION & RATIONALE, rather than to dissect the statements into the separate parts. An example of such a set of statements is provided below from AD's diary:

DECISION & RATIONALE
"The problem of the slow access time on the EPROM was overcome by the decision to slow the processor clock down, due to the fact that there is no real need for the processor to run at its full 4MHz, and so the slower clock rate will not reduce the system performance at all." (Diary 22/10 to 28/10)

4.3.6. DIFFICULTIES

Throughout their project work the subjects were continually reaching points where their plans were not working out or actions were being thwarted so that the continuing pursual of some aspect of the design or its realisation was being hindered or prevented. Any statements placed in the DBG that related to the existence of a difficulty, some blocking condition or a failed plan were preceded by the label DIFFICULTY. Difficulties that the subjects mentioned were seen to arise because of many reasons, including, for example, the lack of information, the inadequacy of available resources, or the non-familiarity with a particular skill. An example of a difficulty that arose for MM is presented below:

DIFFICULTY
"The main problem came up when I could not find a constant-current oscillator (I can find an oscillator and also a constant current generator but not something that does both together." (Diary 29/10 to 4/11)

4.4. DISCUSSIONS OF INDIVIDUAL DESIGN BEHAVIOUR GRAPHS

4.4.1. INTRODUCTION TO DISCUSSIONS

When reporting inductive research it is normal and desirable to deal with "specific" findings first (in the present study the detailed design behaviours of individual subjects) before then moving on to draw out the commonalities in subjects behaviours. However, the foregoing sections of this chapter, whilst being primarily concerned with introducing the encoding and structuring scheme that was devised for analysing the present verbal data, have - of necessity - required mention to be made of some general features of subjects' design behaviour. For example, the pervasive use by these undergraduate engineers of a subgoaling (or problem reduction) strategy was a feature of design behaviour that has been noted repeatedly. To some extent, then, the previous sections have run ahead of the desired flow from specific findings to general conclusions. This state of affairs, however, seems to arise as an inevitable consequence of protocol-based research within a new domain of study since to define a worthwhile encoding scheme for categorising verbalisations you clearly have to gain some initial idea of the actual behaviours that are "commonly" occurring and which need to be encoded - which is something that can only be achieved through a preliminary examination of protocol material. Sections 4.4.2 and 4.4.3 below, therefore, stand as an attempt to resume the flow of reporting from specific findings to general conclusions in that the *detailed* design behaviour of two individual subjects (whose verbal data was encoded and structured as Design Behaviour Graphs) is examined. More

general aspects of the engineering design process emerging from these individual analyses will be commented on in the general discussion that concludes this chapter.

In the next two sections, then, attention will be focussed exclusively on the DBGs constructed for NB and TS. During their production it was intended that all DBGs should be highly meaningful when read from beginning to end - assuming, of course, that the reader had gained an understanding of the classification and structuring formalism underlying their construction - and it is believed that this objective has generally been attained. In particular, it is felt that the graphs clearly depict (1) the development of each engineer's actual technical design solution (i.e. the model of the desired artifact) as well as (2) the structure of each engineer's design activities (i.e. the organisation and sequencing of behaviours). Still, however, it would seems judicious to supplement a subset of these graphs with a detailed, qualitative description of the design activities that occurred.

Certain features of each engineer's design activities which were viewed as being particularly relevant to the following discussions were:

(1) the main design goals that were arising during a project, with particular reference to (i) the success or failure of their pursuit (ii) their reduction to subgoals and sub-subgoals etc. (iii) their modification with the passage of time and (iv) their abandonment by decision or by forgetting;

(2) the major difficulties that were occurring during a project, with particular reference to (i) their causes and (ii) the strategies that were being employed to cope with them; and

(3) the decisions that were being made during a project, with particular reference to (i) the goals to which these decisions related and (ii) the procedures that were involved in making them.

It should be noted that the following discussions do not particularly aim at any "theoretical" interpretation of the nature and organisation of design behaviours in terms of underlying knowledge structures and cognitive mechanisms. Rather, the theoretical analysis of design behaviours in terms of existing theories of problem solving and design is felt to be best left to the general discussion section that concludes the present chapter.

4.4.2. NB'S DESIGN BEHAVIOUR GRAPH

THE INITIAL DESIGN PHASES OF NB'S PROJECT

At the beginning of this study, NB (referred to as S3 in the DBG) produced a copy of a "project specification" which he had written prior to the summer vacation. From this document it was clear that he had engaged in some crucial project planning and decision making before the commencement of the investigation. Because of the obvious importance of this pre-diary design work, a tentative attempt was made within the DBG to reconstruct the high-level goals that were motivating NB's behaviour during this formative phase of his project. NB's overriding goal, then, as inferred from his project proposal was TO DESIGN AND IMPLEMENT AN AUTOMATIC TAPE-POSITION CONTROLLER FOR A REEL-TO-REEL TAPE MACHINE (GOAL 1). In pursuing this overriding project goal, it appears that NB began by detailing the desired functionality of the intended tape-position controller. He stated that it should be able (1) to display tape position in real time (2) to store particular tape-position values in

memory (3) to locate particular tape positions automatically (4) to allow the programming of sequences of simpler functions such as "search" for a tape position and "record" upon finding it (several more such functions are listed by NB) and (5) to look at status of inputs to check that the equipment is connected properly. In defining this exact functionality of the tape-position controller, NB seems to have been motivated by a high-level goal of the form: DEFINE FUNCTIONAL REQUIREMENTS OF DEVICE (GOAL 1.1).

NB next appears to have proceeded to determine the set of abstract hardware modules which could meet the functional requirements of the desired tape-position controller. He specified that this artifact would need to be made up of seven distinct modules, namely (1) a tape-motion sensor (2) a microprocessor and memory devices (3) a keyboard for entering information and commands (4) a display (5) interfacing for outputs and inputs (6) an auxiliary switching interface and (7) a power supply unit. During this phase of design work NB also produced a diagrammatic characterisation of this abstract design solution in the form of a set of interconnected and labelled functional blocks (see figure 4.2 below). This phase of work indicates that two more implicit design goals had been attained which may be specified as DEFINE HIGH-LEVEL MODULES OF DEVICE (GOAL 1.2) and DEVISE REPRESENTATION OF MODULES AND INTERCONNECTIONS IN BLOCK DIAGRAM FORM (GOAL 1.3). By this point, then, NB's overriding design problem - as captured by GOAL 1 - was clearly reaching a much more well-defined state. It is noteworthy that the decision to include a microprocessor was certainly a significant one which was apparently made at a very early stage of the design. The rationale given for this decision was that the microprocessor would

enable the provision of some of the more sophisticated functions of the device and it would additionally eliminate the need for some complex and costly circuitry. (The fact that NB had decided that the tape-position controller would be driven by a microprocessor clearly indicates that there was to be a substantial software component to the project).

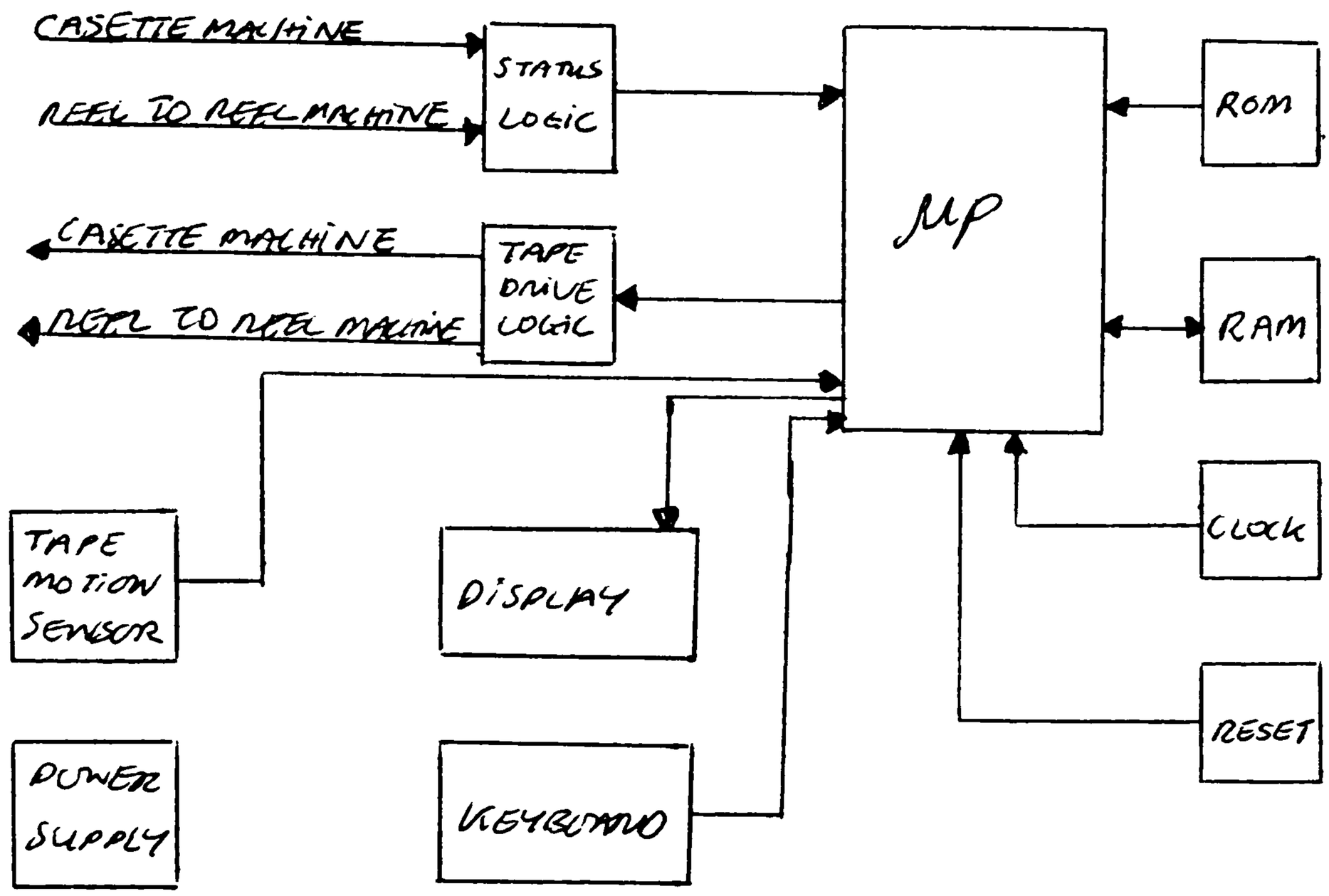


Figure 4.2: NBs initial abstract solution for his tape-position controller design problem.

In attaining GOALS 1.2 and 1.3 NB had arrived at an abstract solution for his design problem. The remaining - and clearly non-trivial - task that now confronted him was to convert these abstract solution concepts into some form of concrete and implementable design. The first hardware module that NB focussed on designing - as evidenced by some

retrospective comments contained within his diary - was the actual "tape-motion sensor", i.e. the device that would detect tape movement. Again, then, NB was inferred to be following a goal of the form DESIGN TAPE MOTION SENSOR (GOAL 1.4) and his comments relating to the pursual of this goal indicate that he must have immediately set a subgoal TO INVESTIGATE WAYS OF MEASURING TAPE MOVEMENT (KAG-1) before actually trying to develop the sensor. It is not clear from NB's diary report whether he was investigating ways of measuring tape movement (a) simply because he had no idea about how this might be done or (b) because he was striving for an optimal way in which such measurement might be achieved. In interview, however, NB claimed that the latter was the case - though without any observable evidence to corroborate this claim its accuracy remains questionable.

Returning, then, to NB's pursual of GOAL 1.4, it emerges that after carrying out some in-depth research on how to go about measuring tape movement he decided to opt for a "mechanical contraption" such as an idler-wheel, which would be designed so as to produce a signal that could be input into the circuit to indicate tape movement. Having arrived at this basic concept, NB next pursued a detailed design of the mechanical sensor (GOAL 1.4.1: DESIGN MECHANICAL ASPECTS OF TAPE MOTION SENSOR). Interestingly, NB said that it was necessary to carry out the design of the sensor at this particular point in the project since the actual construction of the mechanism would be left in the hands of the departmental technicians - with the ensuing possibility (due to heavy workloads) of lengthy delays. This rationale, then, clearly indicates that NB was undertaking the design of functional modules within a realistic model of external constraints that might impinge on the design

and development project and possibly thwart its success.

NB eventually attained a detailed (though unspecified) mechanical design for the tape-motion sensor which he then passed on to the technicians for construction - in line with GOAL 1.5 (IMPLEMENT MECHANICAL ASPECTS OF TAPE MOTION SENSOR). The sensor mechanism was actually constructed by the technicians prior to the Summer vacation. However, NB reports that when he attempted to fit it to the tape machine it proved to be operationally useless. Interestingly this fact suggests that the design concept for the sensor that NB decided upon (see above) was perhaps less than optimal. NB indicates that he returned the sensor mechanism to the technicians for re-building, presumably at the start of the new autumn term.

At the start of this new term (i.e. at the point where the actual diary study commenced) it appears that NB was beginning to direct his efforts toward the detailed design of the complete electronic circuit (GOAL 1.6: "TO GET THIS BLOCK DIAGRAM INTO SOMETHING THAT CAN BE IMPLEMENTED USING ELECTRONIC COMPONENTS"). This high-level, long-term design goal was seen to be pursued continuously by NB over many weeks of the project and the activities that surround its pursual and eventual attainment are clearly of direct relevance to many of the issues which this exploratory study was attempting to address. The following section, therefore, provides a detailed discussion of NB's activities during the continuing hardware design phases of his project. It should be noted that NB's concurrent diary comments provided the main source of verbal data for the construction of the DBG for this phase of his project, and only limited recourse was made to his interview material.

THE CONTINUING DESIGN PHASES OF NB'S PROJECT

Having set the high-level design goal of turning his abstract solution (see block diagram in figure 4.2) into detailed circuitry, NB was now clearly faced with a procedural choice of what to do first to attain this goal. His initial subgoal, then, inferred from what he was seen to be working on, was to DESIGN THE ELECTRONIC ASPECTS OF THE TAPE-MOTION SENSOR (GOAL 1.6.1 or GOAL 1.4.2). Objectively, this seems to have been a sensible place for NB to start the actual circuit design since this work clearly followed on conceptually from the pre-Summer activities relating to the design of the "mechanical" aspects of the sensor. NB appears to have rapidly attained an abstract design solution for the electronic part of the sensor for within the diary he provided a conceptual outline of a "light-gate assembly" that could detect the movement of the idler-wheel and then output the signals to the rest of the circuit. NB in fact commented that he had already decided on this basic design concept when pursuing the design of the mechanical part of the device, though it does seem that this abstract design concept was concretised at the present point in time since a detailed design solution for this functional sub-module was seen to be pursued depth-first to completion.

NB next appears to have set a goal TO SELECT A DISPLAY (GOAL 1.6.2) and immediately to have set as a subgoal to "EVALUATE DIFFERENT DISPLAYS, ALL OF THE INTELLIGENT LCD DOT MATRIX TYPE, OFFERING SIMILAR PERFORMANCE" (OEG-1). The pursual of this option evaluation goal, however, was thwarted by a lack of data sheets - and no further comments relating to its pursual are contained within subsequently derived protocol material. (Clearly though, a suitable display was found by NB

since some months later he was seen to be engaged in wiring and testing it). The omission of any comments regarding the choice of a display would appear to be a case of the subject simply forgetting to write sought-after information in the diary. Clearly this is a problem with the use of cognitive diaries as a method of knowledge elicitation (see section 4.6.2 for further discussion of this limitation of the technique).

Returning to NB's design work, he appears to have next moved on to the design of the keyboard module of the artifact and his activities were, therefore, presumably directed toward the attainment of the goal TO DESIGN THE KEYBOARD (GOAL 1.6.3). The problem of designing the keyboard hardware appears to have been initially decomposed into the problem of designing the actual "layout" that the keyboard should take (GOAL 1.6.3.1: DESIGN KEYBOARD LAYOUT). The major trade-off that NB mentions with regard to a selection of an appropriate layout is that the the number of keys should be kept low whilst the ergonomic quality - based on the idea of "one-key-one-function" - should be kept high. NB's initial attempt at a solution, which involved designing a layout first and then evaluating its suitability, ended in failure. NB therefore decided first (a) to attempt to define the keyboard functions so that the various keystroke sequences that were required for the application could be investigated before (b) settling on a particular layout design. This method appears to have enabled NB to attain a basic concept for the layout - although not without some considerable difficulty arising during the phase of defining the relationships between possible keyboard functions and sub-functions (see the DBG for details of NB's planned method and the method that was actually used as well as decisions that

were made). It should also be noted that NB returned to the design of the keyboard layout several weeks later and spent a fair amount of time iterating and elaborating the basic layout concept before reaching an exact solution.

NB's next set of design activities were inferred to be directed toward attaining GOAL 1.6.4 (TO CHOOSE THE MICROPROCESSOR AND DESIGN ITS SUPPORT DEVICES). This basic goal was depicted in the DBG as three interdependent but conceptually separate subgoals that appear to have been pursued sequentially by NB. These subgoals were: GOAL 1.6.4.1 (CHOOSE THE MICROPROCESSOR), GOAL 1.6.4.2 (SELECT CLOCK CRYSTAL) and GOAL 1.6.4.3 ("DESIGN A CIRCUIT TO HOLD THE MICROPROCESSOR RESET PIN LOW FOR A NOMINAL PERIOD OF POWER UP"). All three goals were attained without any apparent difficulties (see DBG). NB then proceeded to engage in activities relating to the design of the keyboard "circuitry". The goal TO DESIGN KEYBOARD CIRCUITRY (GOAL 1.6.3.2) may be viewed as another subgoal of the higher-level goal TO DESIGN THE KEYBOARD inferred to be set a few weeks earlier. In pursuing the design of the circuit, NB decided on the concept of a keyboard encoder, although the need was seen for some keys to be able to interrupt the microprocessor when it was not polling the keyboard. NB also selected capacitors for the keyboard, thus achieving a lower-level subgoal (SELECT CAPACITORS FOR KEYBOARD CIRCUIT: GOAL 1.6.3.2.1).

NB's activities next appeared to be directed toward the attainment of an intermediate-term design goal which could be stated as DESIGN THE TAPE DRIVE INTERFACE CIRCUIT FOR THE REEL-TO-REEL AND CASSETTE MACHINES (GOAL 1.6.5). The first subgoal of this that was tackled was TO DESIGN THE INTERFACE CIRCUIT FOR THE REEL-TO-REEL MACHINE (GOAL 1.6.5.1) and NB

derived a solution for this design problem by gradual iteration of an initial solution concept (see the series of circuit diagrams presented within his diary for week starting 5/11). The next week NB explicitly set a goal to revise this detailed solution with the intention of producing a simpler design that would be easier to implement (GOAL 1.6.5.2: "TO SIMPLIFY THE DESIGN, AND HENCE THE CONSTRUCTION +++ OF [REEL-TO-REEL] TAPE-DRIVE LOGIC"). This detailed circuit design for the reel-to-reel tape-drive logic was, however, only a small section of the whole tape drive interface that needed to be designed. It can be discerned from NB's comments that he still had (a) to design the remote control switching circuit for a cassette machine and (b) to design the related circuitry to drive a set of LED indicators (since he had previously made the decision that LEDs should be provided to indicate which tape drive functions were currently instantiated by the user). A major constraint that was mentioned by NB with respect to this interface design work was that all the signals which controlled the tape machines were derived from a single 8 bit I/O port - the point being, then, that a clear need existed (c) to design extra circuitry to ensure the appropriate "routing" of control signals. It is clear from NB's diary comments that he conceptualised these three subgoals (note that they have not been represented within the DBG) as being inextricably interconnected, which indeed they were by virtue of this aforementioned constraint. NB, therefore, summarised his immediate goal at this point as being "TO DESIGN A CIRCUIT WHICH WOULD HANDLE THE REMOTE CONTROL SIGNALS FOR REEL-TO-REEL AND CASSETTE MACHINES, AND KEYBOARD SWITCH INDICATORS (LEDs) TO INDICATE CERTAIN FUNCTIONS, ALL FROM A SINGLE I/O PORT". This goal was represented within the DBG as a repetition and elaboration of GOAL 1.6.5 (bearing in mind that it had been partially

fulfilled by the derivation of the switching circuit for the reel to reel machine). NB's account of his design work in pursuing this elaborated GOAL 1.6.5 was in the form of a fairly retrospective summary of design decisions and the reasoning behind them (see DBG). Suffice it to say here that an intricate circuit design was eventually developed in an apparently iterative manner, with careful consideration being given to the constraint that all signals needed to be derived from a single, limited-capacity I/O port.

The design of the interface circuit seems to have been the most difficult aspect of the overall design that NB had to carry out up to this point. Having attained this goal NB now explicitly set as his goal "TO SELECT ROM AND RAM FOR THE MICROPROCESSOR" (GOAL 1.6.4.4). RAM was decided to be superfluous (the microprocessor having enough built in) and ROM was chosen which had sufficient capacity at the lowest cost.

The final design goal that was set by NB was inferred to be TO DESIGN A STATUS SIGNAL INTERFACE (GOAL 1.6.5). NB provided a functional description of what the circuit must do before detailing a solution to the design problem. The remaining hardware design work that NB was seen to engage in related to the continuing pursuit of a satisfactory keyboard layout (i.e. GOAL 1.6.3.1). Having derived a fairly abstract solution to the layout problem earlier, NB now specified exactly what keys would be used and how they would be grouped together (see DBG). The complete and integrated solution that NB produced for the hardware section of his overriding design problem is shown in figure 4.3.

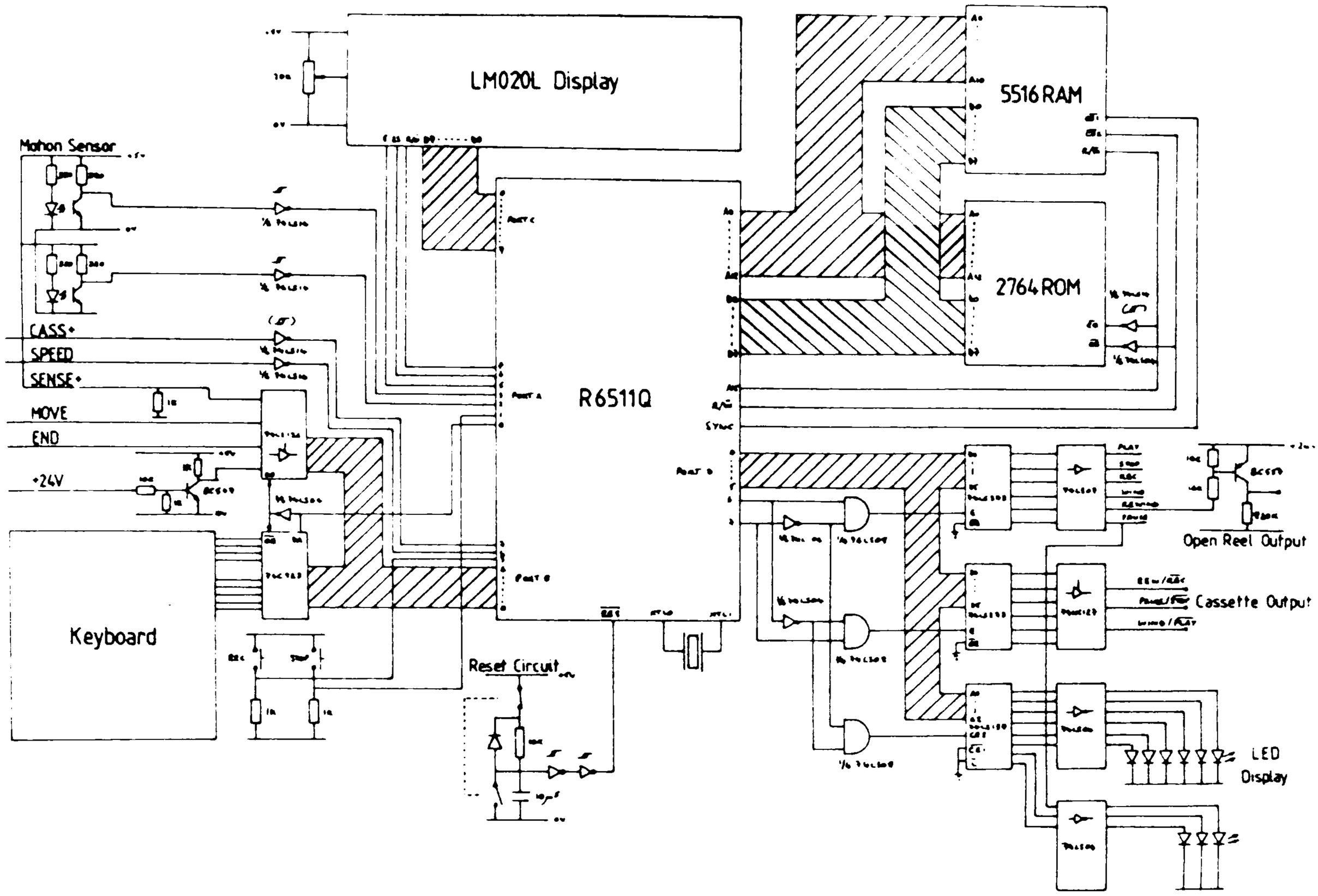


Figure 4.3: NB's complete low-level solution for the hardware aspects of his overriding design problem.

HARDWARE IMPLEMENTATION AND SOFTWARE DESIGN PHASES OF NB'S PROJECT

An extensive part of NB's project was concerned with the actual implementation in hardware of the circuits that had been designed and the attendant testing and redesign that was necessitated. Additionally software was eventually written to enable the full functionality of the intended device to be achieved. Details of hardware implementation and software design activities have not been provided in the DBG, nor will discussion of these activities be provided here. The reason for these omissions is that, strictly speaking, such activities lie outside the

true focus of the present research on the processes involved in *design* as opposed to the processes involved in *implementing* a design solution.

THE GLOBAL STRUCTURE OF NB'S DESIGN PROCESS

From the previous account of NB's design activities, it is clear that he was making use of a definite subgoaling strategy during his design work. In operation, this strategy led to the splitting up of his overriding design goal (i.e. to attain a design for the automatic tape-position controller) into various subgoals which related to the design of modules or submodules of this desired artifact. In terms of a problem-oriented interpretation of NB's subgoaling strategy it could be said that the method involved the decomposition of a top-level design problem into several subproblems each of which revolved around separate sets of functional requirement. This subgoaling or problem reduction method, then, is identical to the informally termed "divide and conquer" technique which is commonly advocated in the design literature as a sensible way of tackling complex design problem. One characteristic feature of the problem reduction method as applied in design situations is that it involves a top-down, depth-first pursual of design subproblems and the consequent top-down, depth-first development of design solutions. NB's design activity displayed these latter features, though it must be admitted that the number of design levels that were traversed rarely tended to be more than three (i.e. detailed solutions were generally seen to be attained quite readily).

It would seem worthwhile here to recount briefly the basic phases of the design process that were seen to arise from NB's application of a

subgoaling strategy. First of all, then, NB defined the fundamental functions that the device would have to perform. Subsequent to this major problem definition phase, NB next derived a set of interconnected hardware modules that could provide the basis of the intended functionality of the device. This set of seven hardware modules may be described as an abstract "solution model" for the overriding design problem. Over the ensuing project weeks NB focussed on the design of these hardware modules, tackling them essentially one at a time. Each module, whilst in one sense existing as part of an abstract solution, in another sense existed as a subproblem of hardware design (i.e. the problem was how to convert the abstract module into a detailed circuit). When NB selected a module to work on, he often first appeared to search for a solution concept for that module. Examples of such solution concepts - which again tended to be fairly abstract in nature - were the use of a "mechanical contraption" for detecting tape movement (the module being dealt with in this case was the tape-motion sensor) and the use of a "keyboard encoder" (the module being tackled was the keyboard circuit). When such an abstract solution concept had been found for a module, NB was then seen either (1) to develop this abstract concept directly to completion by detailing a circuit diagram in a single pass, or (2) to develop the abstract concept gradually to completion through an iterative process of detailing a concrete solution and then correcting small inadequacies in this concrete solution, or (3) to reject the abstract concept as ill-conceived upon attempting to develop it further. In this latter case a new solution concept was searched for and then the process of developing it re-instigated.

One important issue that arises from this general characterisation of

NB's design procedure relates to why the modules were designed in the particular order that was observed. The reason for NB first tackling the design of the tape motion sensor, for example, seems clear; he realises that the sensor device may need to be constructed by someone else and may be held up in being built. It is therefore vital to get a design sorted out as soon as possible to prevent any subsequent delays from having too disastrous effect on his ongoing project. The issue of ordering and control within design will be taken up in greater detail in the general discussion section of this chapter.

4.4.3. TS'S DESIGN BEHAVIOUR GRAPH

THE INITIAL DESIGN PHASES OF TS'S PROJECT

At the commencement of the diary study, TS (referred to as S8 in the DBG) seemed to be quite heavily involved in initial project planning, goal setting and decision making - though some design-oriented activities had certainly been carried out prior to the start of the investigation. The diary and interview reports that TS provided were, therefore, viewed as being fairly concurrent accounts of his ongoing project activities.

TS's overriding goal was inferred to be TO REDESIGN AN EXISTING APPLICATION SPECIFIC DIRECT MEMORY ACCESSING UNIT TO OVERCOME CURRENT PERFORMANCE DEFICIENCIES (GOAL 1). A noteworthy aspect of TS's project, then, was that it involved design and redesign work only rather than implementation work as well. The existing Direct Memory Accessing (DMA) unit, which had been implemented in discrete logic as board level components, was an integral part of a CAD graphics workstation being developed as a commercial product by company C-----. The original DMA

unit that the company were working with, however, whilst seemingly usable, was far from optimal from a software management point of view. TS had, therefore, been asked to attempt to redesign the current device so as to improve its performance, i.e. to make it quicker, easier to use, and, if possible, cheaper. A mutual decision was made by TS and the company to produce a redesign (or really a completely new design) of the DMA unit as a gate array device, the rationale behind this being that a gate array implementation would allow for improved speed and versatility as well as reduced component size, cost and power requirements. TS started off, then, with a fairly clear understanding of the major constraints that were to be reflected in the device that he was to design.

TS's design work throughout his project appears to have been very highly structured, planned and deliberated and these general characteristics of his behaviour emerged particularly clearly in the first few weeks of the project. Early on, then, TS stated his plan to break the project into two distinct phases. The first phase was to be aimed toward attaining the required redesign of the DMA unit as specified by the company. Important constraints that were included in the company specification were - as noted above - that the chip should be cheap, simple, small and fast. Also of apparently crucial importance, however, was the constraint that the chip should be designed so that it was testable. The second project phase that was planned by TS was to be more research-oriented in that it would involve investigation of the control aspects of the DMA unit. One possibility here related to the incorporation of some additional control logic into the design of the DMA chip. The suggestion to investigate control aspects of the DMA unit and the idea

of incorporating additional control logic on the chip both appear to have come from TS's supervisor. TS clearly realised that this second project phase was highly ambitious relative to the time constraints of the project. Moreover he believed that the company would not really be interested in this side of the project anyway - presumably because the constraint of keeping the chip inexpensive would be violated by the need to place extra pins on the chip as a result of incorporating additional control logic (i.e. manufacturing expense of chips tends to increase as a function of pin number).

It is noteworthy that early on in his project, TS decided to use the Polytechnic's Apollo CAD facilities for his planned design work. The main rationales behind this decision appeared to be that the system was readily accessible and that TS had some existing experience with its use. It should be stressed, however, that some kind of CAD system was essential for the gate-array design of the DMA unit since to design at the gate level manually would be an inconceivably arduous task. Also during this early stages of TS's project, a readily discernible feature of his design work was the extent to which his efforts were directed towards gaining a clearer understanding of the problem at hand in terms of the functional requirements of the intended device. It appears, then, that the problem was ill-defined at this point in that TS possessed little understanding of the actual functional requirements of a DMA unit even though he had a clear notion of other constraints that had to be met (e.g. that the DMA unit had to operate at a high speed).

At this design stage, then, TS exhibited a great deal of information seeking behaviour which was inferred to be directed toward the attainment of the knowledge acquisition goal GAIN KNOWLEDGE OF DMA UNIT

FUNCTIONALITY (KAG-1). This goal was immediately decomposed into the subgoals INVESTIGATE COMMERCIALLY PRODUCED DEVICES IN GENERAL (KAG-1.1) and LOOK AT DIAGRAMS FOR PREVIOUS APPLICATION SPECIFIC DEVICE (KAG-1.2). Both of these inferred short-term subgoals appeared to be extensively pursued and presumably TS's knowledge of the functional requirements of DMA units in general started to become fairly expansive as a result of the research being carried out. He clearly felt, however, that he had an inadequate understanding of the functionality of the existing application-specific unit as he was seen to set yet another knowledge acquisition goal "TO EXPAND MY UNDERSTANDING OF THE DMA UNIT OPERATION AND TO UNDERSTAND HOW THE SYSTEM CONTROLS THE DMA UNIT" (KAG-2). To attain this knowledge acquisition goal TS set subgoals "TO WRITE TO C-------- FOR FURTHER INFORMATION" and "TO WRITE DOWN HOW I THINK THE SYSTEM WORKS" (KAG-2.1 and KAG-2.2). Both of these subgoals were pursued by TS with, presumably, the consequence that the DMA unit design problem became increasingly well-defined in terms of its required functionality.

TS's next significant design goal was inferred from the context of his continuing activities to be to DRAW A HIGH LEVEL BLOCK DIAGRAM WITH CONNECTIONS THAT MAY BE NECESSARY (GOAL 1.1). This was divided into two related subgoals, i.e. GOAL 1.1.1 "TO GET DETAILED FUNCTIONAL DIAGRAMS ON PAPER" and GOAL 1.1.2 "TO DO SIMPLE FUNCTIONAL DIAGRAMS ON CAD SYSTEM". The reason for TS designing on paper was that he was as yet unfamiliar with the commands to drive the CAD system. This difficulty was again tackled by the instantiation of information-seeking behaviour which was aimed toward gaining some basic familiarity with the CAD system. On paper, however, TS soon appeared to reach an abstract design solution for the DMA unit which consisted of three high-level functional

modules, i.e. (1) an I/O PORT for transferring data and coping with the speed differences between input and output devices, (2) an ADDRESS POINTER which would act as a counter to select address locations for data and (3) a CONTROL module for controlling the operation of the DMA unit. TS represented these basic blocks and their fundamental connectivity on paper in the form shown in figure 4.4.

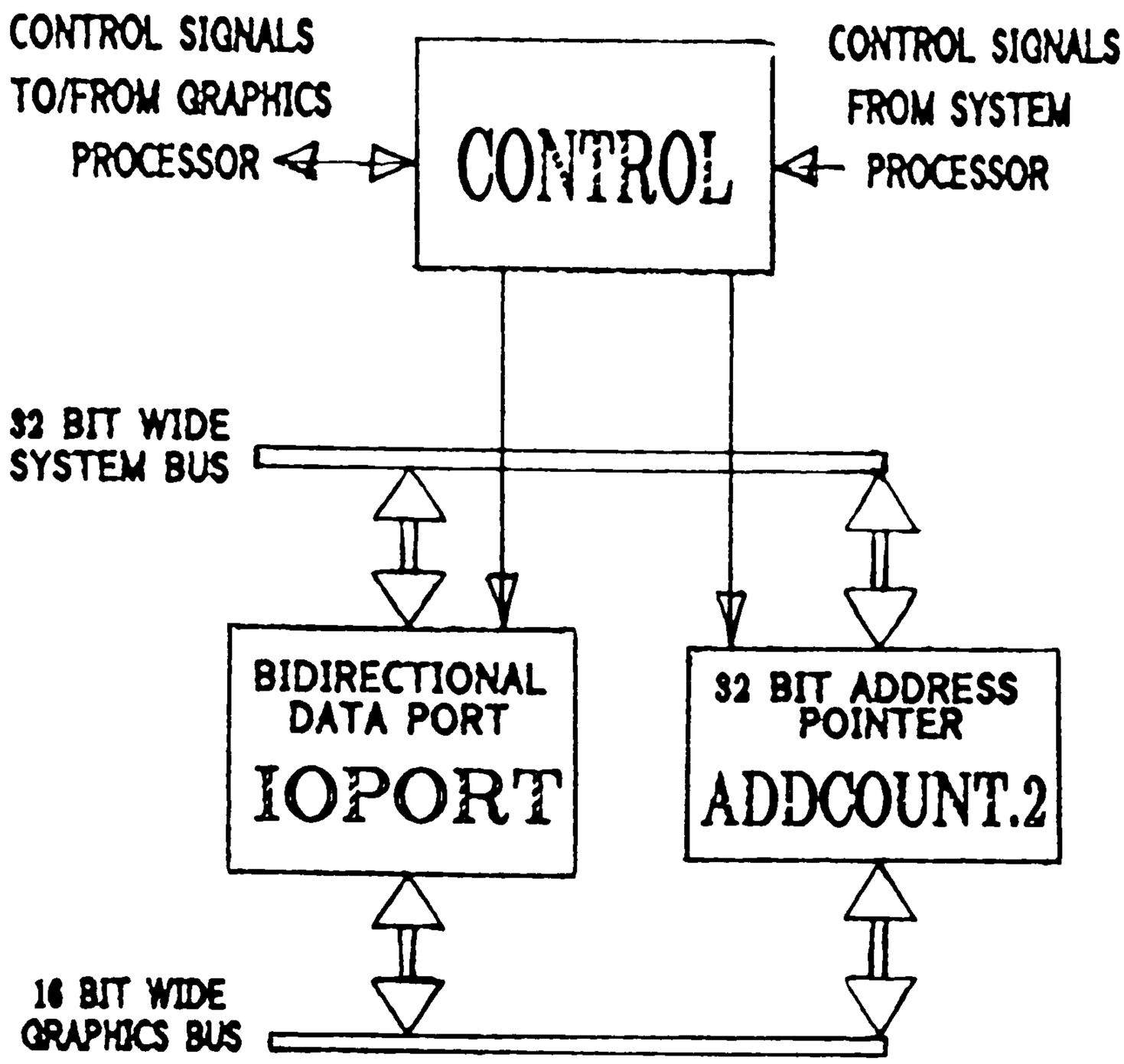


Figure 4.4: TS's initial abstract solution for the DMA controller design problem.

THE CONTINUING DESIGN PHASES OF TS'S PROJECT

During the continuing phases of TS's design project he was noted to repeat a project documentation goal that he had set the previous week which related to the writing of the introduction to his project report (PDG-1.1: "AIM TO GET A FULL INTRODUCTION TO MY PROJECT"). TS clearly viewed the attainment of this goal as being of major importance for

fulfilling the mutual objectives of (1) project documentation and (2) clarification of his understanding of the DMA unit's operation (cf. KAG-2). This project documentation goal was pursued exclusively over the following project week and was soon attained. TS then set a new project documentation goal "TO WRITE REPORT ON DMA UNIT WITH MORE TECHNICAL DETAIL AND STATE AIMS OF PROJECT" (PDG-1.2). This goal, however, was not seen to be pursued immediately.

TS was next seen to have completed the drawing of a single high-level module on the CAD system and he subsequently set a deadline to complete the whole top-level diagram on the system during the forthcoming week (TSG-2). Difficulties arose, however, during TS's efforts to wire the blocks together on the system and thus his attainment of this TSG was thwarted with the consequence that it was revised - the new aim being for the diagram to be completed by Christmas. At this point a knowledge acquisition goal was also set to gain information relevant to overcoming the wiring problem (KAG-4) and this goal was pursued and presumably attained. TS additionally mentioned a very general difficulty that would hinder his continuing design work, i.e. that the new component library which was needed for the detailed gate array design was still awaiting delivery. TS decided that rather than pursuing a detailed design with the existing library - which would only have to be re-done when the new library arrived - he would instead merely "think about" the internal components for the blocks.

TS was next heard to repeat PDG-1.2 in the following form: "I HAVE DECIDED THAT MY PROJECT REPORT WILL REQUIRE A DETAILED ANALYSIS OF HOW THE GRAPHICS SYSTEM WORKS AND CONTROLS THE DMA". This time, however, PDG-1.2 was pursued and attained. During this project week TS also

managed to link the high-level blocks together on the CAD system and thus attained an abstract model of the design solution on the Apollo rather than just on paper. Figure 4.5 shows this abstract solution to the DMA design problem. It can be seen that the connectivity between the three basic modules of the device is rather more elaborate than that depicted in the initial solution that TS externalised on paper (see figure 4.4 above). This increased detail perhaps reflects TS's improved understanding of the fundamental control aspects of the DMA unit which he had spent some time researching into.

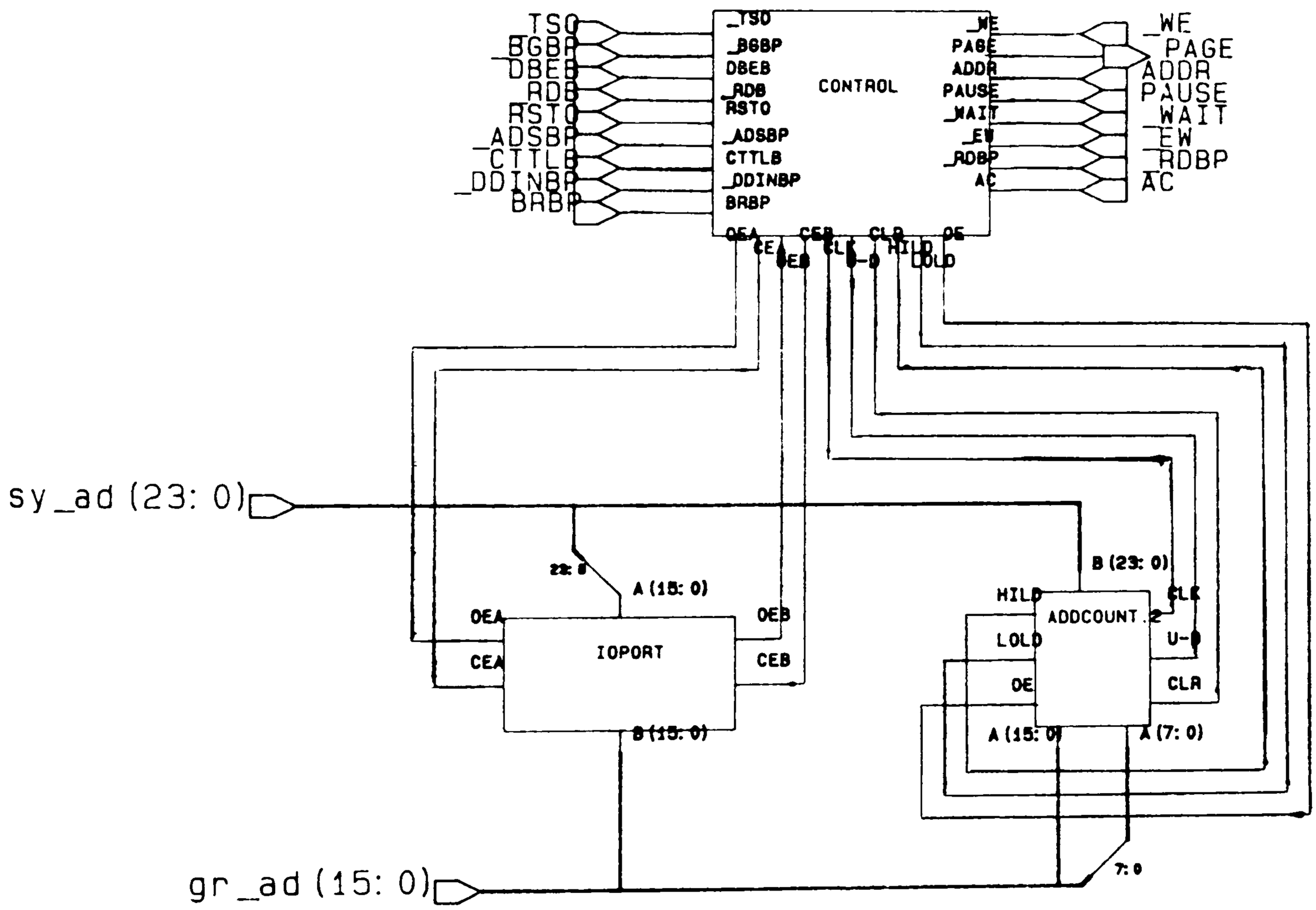


Figure 4.5: TS's abstract solution for the DMA controller design problem as modelled on the Apollo CAD system.

One noteworthy feature of TS's project work at this point was that he was confronted with various design options which he was unsure of evaluating since he lacked information that had been requested from company C-----. In tackling this difficulty he decided to keep his options open and to talk with the company about the design alternatives that had arisen (KAG-5: "WILL GO AND TALK TO C----- [ABOUT DESIGN ALTERNATIVES] DURING THE CHRISTMAS VACATION"). TS also repeated this knowledge acquisition goal the next project week ("I AM GOING TO HAVE TO DISCUSS WITH THEM EXACTLY WHAT THEY NEED, TO DECIDE WHICH DESIGN OPTIONS TO TAKE FOR MY FINAL BLOCK DIAGRAM)". TS provided some detail of the nature of the available design alternatives, which related specifically to the "addressing" section of the DMA unit. Two options existed for loading the address counter, i.e. it could either be loaded directly from the graphics processor (an option that would result in improved speed of functioning) or it could be loaded via the I/O port. TS finally decided to produce two high-level block diagrams representing these alternatives so that he could take them to the company at a later data in order to ascertain which solution they preferred.

TS next appeared to engage in some idea generation and general planning activities relating to the actual design of the DMA unit. Firstly, he was seen to give some consideration to the "phase two" part of his project which concerned the inclusion of more control logic on the DMA unit. He mentioned that to do the latter would require the incorporation of control status registers within the device. After giving some thought to this aspect of the project, however, TS more or less claimed that he would not be pursuing the second phase since he had discovered that he should not expand his ideas too much but should rather restrict his

technical design so that it furthered the the aims of the company. This emerges as a clear example of social constraint impinging on the derivation of a technical design solution and is an issue that will be dealt with further in the general discussion section of this chapter (see particularly section 4.5.6). A second idea that TS mentioned at this point related to the expansion of busses, i.e instead of taking a 16 bit bus he suggested that it might be useful to take a 32 bit bus. This design concept, however, would require a different and more costly processor than was being currently employed and TS indicated that he would be unlikely to pursue this idea further. A third idea that TS had at this time related to the inclusion of a memory management system on the chip which he felt would make things simpler for the software people at the company. Again, however, it appeared that this idea was not pursued further, perhaps because of the inherent complexity involved in instantiating such a system on board the chip.

After engaging in this latter phase of planning and idea generation, TS was next seen to direct his efforts toward the detailed elaboration of the functional blocks of his abstract design solution using components from the newly installed gate-array library. TS explicitly set as his ongoing intermediate-term goal "TO FILL OUT THE 'I/O PORT' AND 'ADDCOUNT' BLOCKS WITH GATE ARRAY LOGIC COMPONENTS, AND TO MAKE SOME INITIAL DESIGNS OF THE CONTROL LOGIC" (GOAL 1.2). This goal can itself be be viewed as encompassing three conceptually distinct subgoals, each of which TS appeared to pursue in an essentially sequential manner. These three subgoals may be stated as DESIGN I/O PORT CIRCUIT (GOAL 1.2.1), DESIGN ADDRESS COUNTER CIRCUIT (GOAL 1.2.2) and MAKE INITIAL DESIGNS FOR THE CONTROL LOGIC (GOAL 1.2.3). TS was immediately seen to

direct his attention whole-heartedly toward the problem of designing the I/O port and gave only limited initial thought to the design problems expressed by GOALS 1.2.2 and 1.2.3.

In pursuing GOAL 1.2.1, TS was initially seen to set a knowledge acquisition goal of the form LOOK AT DATA BOOKS FOR DESIGNS OF COMMERCIALLY AVAILABLE I/O PORTS (KAG-6). As a result of the pursual of this goal, TS decided that he would design the I/O port block using latches and tristate buffers. Unfortunately, at this point in his diary, TS provided rather limited reference to the design work on this module. It was possible, however, to discern (albeit tentatively) the basic stages involved in the design of the I/O port on the basis of circuit diagrams that were produced as well as from retrospective accounts provided within TS's project write-up. The I/O port was a 16 bit bidirectional data port and TS, therefore, saw it necessary to design a 16 bit bidirectional register (GOAL 1.2.1.1: TO DESIGN 16 BIT BIDIRECTIONAL REGISTER). To this end, TS pursued the design of a single bit bidirectional register - or what he termed a "cell" - thus attaining GOAL 1.2.1.1.1: DESIGN CELL). During the design of the cell, relevant control signals were also defined (so attaining GOAL 1.2.1.1.2: DEFINE CONTROL SIGNALS FOR CELL). TS then connected sixteen CELLS in parallel - thus attaining GOAL 1.2.1.1.3: CONNECT 16 CELLS IN PARALLEL- to produce the complete register. The design of cell and thence register appears to have progressed swiftly and smoothly, with circuit diagrams being first devised on paper and then being modelled on the CAD system using the CLA-5000 gate array library. Later, to complete this module, TS incorporated driving components into the design solution so attaining GOAL 1.2.1.1.4 (ADD DRIVING COMPONENTS TO CELLS).

With regard to the work that TS directed toward the pursual of GOAL 1.2.2 (i.e. the design of the address pointer circuit) he was first seen to set a knowledge acquisition goal of the form INVESTIGATE DESIGNS OF COMMERCIALLY AVAILABLE BINARY COUNTERS (KAG-7). Although this goal was certainly pursued by TS, he provided no information of what he been learned from his research or of how he planned to pursue the design of the address pointer (or what he termed the "addcount" block). Similarly no visible manifestations of his research were evident (e.g. diagrams or sketches) until many weeks later. It should also be noted that the activities that TS directed toward the pursual of a design for the control module (GOAL 1.2.3) were likewise very limited at this point in his project. Shortly after this period of work TS stated that he has been in communication with company C----- and had decided to include a second address counter in the design of the addcount block. The rationale that he gave for this decision was simply that it would allow two address locations to be stored - though why this should be advantageous to the functioning of the device was not mentioned.

TS was next observed to direct his efforts toward the redesign of his cell circuits (GOAL 1.2.1.1.5: UPDATE DESIGN OF CELL - AND THEREBY REGISTER ALSO). He stated that the update was necessary since in the previous design solution he had incorrectly included CMOS components which would not have been compatible with the TTL logic levels on the input. TS was then seen to move on to the actual verification of his I/O port, his immediate goal being explicitly stated as "VERIFY (PROVE THROUGH SIMULATION) THE I/O PORT" (GOAL 1.2.1.2). To attain this goal TS was inferred to set a subgoal TO VERIFY THE OPERATION OF A SINGLE CELL (GOAL 1.2.1.2.1). The simulation run of the cell was undertaken and

proved to be successful in that appropriate propagation delays were observed for the circuit operation. TS mentioned that he had some difficulties with the verification of the cell arising from his unfamiliarity with the CAD system. To get around these difficulties he said that he had continually referred to the various reference manuals for the system and the component library.

At around this time TS stated his definitive decision to pursue only "phase one" of the project and to abandon the more ambitious second phase because of time limitations. He additionally stated his aim "TO SPEND LESS THAN TWO WEEKS DESIGNING THE COUNTER - AND TO PROVE IT" (TSG-4). Generally, then, it seems that TS was beginning to feel that time was running out for completing the project. It also emerged at this point that TS made some further progress with regard to a solution concept for the address counter module, i.e. he had decided that it should be a tristate device. The rationale for this decision appears to be that it would enable him to incorporate several independent counters on the DMA unit. With respect to the control block, TS explicitly set as his aim "TO HAVE BROKEN THE CONTROL BLOCK INTO SUBSECTIONS AND HAVE DEFINITE PLANS FOR IMPLEMENTATION SHORTLY HAVING COMPLETED THE COUNTER (BEGINNING OF MARCH)" (TSG-5).

The verification of the complete I/O port was attempted soon after the verification of the CELL operation, but TS was confronted with major difficulties that arose from the fact that he had organised his files on the system in an incorrect manner. He overcame these difficulties by re-organising his files and then successfully completed a verification run of the I/O port. Again the propagation delays were seen to be appropriate for this section of the device under various conditions of

simulated operation. Other work that was undertaken during this period of the project related to the design of the address counter. In particular, TS was seen to set a goal TO EVALUATE COMMERCIAL COUNTER DESIGNS (OEG-2), which was pursued, and he also explicitly revised TSG-4 to give himself an extra week on the design of this module. This TSG, however, was not attained because the laboratory was shut down on the day when TS planned to have attained a concrete design for the address counter.

At this point, with only one module of the DMA unit having been successfully designed and tested, TS ceased to maintain his diary entries. This was presumably because the project deadline was looming close and he was having to devote his efforts toward the completion of his design work. Enough design work had been undertaken by TS up to this point, however, to enable a fairly clear notion to be gained of his basic design strategies and of the global structure of his design processes.

THE GLOBAL STRUCTURE OF TS'S DESIGN PROCESS

It is clear from the foregoing account of TS's design activities that his behaviour was very highly planned and structured throughout the time course of his project. Similar to NB, TS appeared to be adopting a problem reduction strategy to attain a solution to his overriding problem, which in this case related to the redesign (or really a completely new design) of an existing application specific DMA unit. The development of a solution to this design problem was very clearly seen to occur as a top-down, depth-first expansion of the three high-level modules that had initially been detailed by TS in the form of an

abstract solution model.

It would again seem useful here to summarise the fundamental phases of the design process that arose during TS's design project. TS's initial design activities related to the gaining of a clear understanding of the problem at hand in terms of major constraints that had to be met in attaining a design for the DMA unit. The knowledge that he acquired of pertinent constraints seemed to lead him to divide the project itself into two main phases. The first phase, which satisfied the constraints imposed by the company (particularly the cost to actually manufacture the DMA unit) was felt by TS to be fairly attainable. The second planned phase of the project, however, which this student's supervisor had suggested he pursue, was considered by TS to be both highly ambitious and also to partially violate certain constraints of importance to the company. Even early on in his design work, then, TS seemed to have no real desire actually to progress to this second project phase.

TS's work was next directed toward the attainment of a clear notion of the actual functional requirements of the DMA unit that had to be designed. The outcome of this understanding of the functionality of the DMA unit (together with his knowledge of the constraints on this functionality) was the detailing of an abstract solution to the problem. This abstract solution depicted three top-level modules and their basic connectivity. The interconnections between these blocks was elaborated by TS as his knowledge of DMA unit functionality became more advanced over the next few weeks. Then, over the ensuing months of the project, these top-level modules were extensively developed to detailed levels in what appeared to be an essentially sequential fashion. When TS selected a top-level module to develop further, then, he was seen to search for a

fairly abstract solution concept (cf. NB's design behaviour) - a process that required extensive information seeking behaviour in the form of reading-up on commercially available devices. Such an abstract solution concept for a top-level module was again seen to be externalised as a fairly high-level block diagram comprising interconnected functional submodules. In the case of the I/O PORT, for example, the submodules of the solution were the REGISTER and DRIVE CIRCUITRY.

Unfortunately, TS failed to continue his diary reporting during the development of the ADDRESS COUNTER and CONTROL parts of his overriding design solution. From an analysis of his final project write-up, however, it can tentatively be concluded that he pursued a similar top-down, depth-first development of solutions for these functional modules. For example, the ADDRESS COUNTER appeared to have been neatly modularised by TS into blocks which he referred to as BIREG (i.e. a dual 32 bit register) and COUNT32 (i.e. a 32 bit counter) and the indication was that detailed designs were pursued for each of these in a depth-first manner.

Again a final issue of interest relates to why TS designed the high-level modules in the order that was observed. The apparent answer to this is that these modules were tackled in easiest-first sequence - particularly in light of the fact that in his project write-up TS mentioned that the ADDRESS COUNTER was a far more complex module than the I/O PORT.

4.5. GENERAL DISCUSSION

4.5.1. OUTLINE OF THE GENERAL DISCUSSION

As has already been stated repeatedly in this chapter, the present study was undertaken as an essentially ***exploratory*** psychological investigation of engineering design processes. In this regard, then, it is believed that the study was largely successful in that the verbal data obtained - a subset of which has been discussed in detail - appear to shed considerable light on the semi-expert (i.e. undergraduate-level) activities that are involved in tackling long-term design projects. One particularly important feature of the results was the distinct ***similarity*** apparent in the design behaviours of the engineers studied. This similarity in the structure and organisation of design activities emerged very clearly in the Design Behaviour Graphs that were produced for four subjects (see appendix A) and was also supported by a more cursory analysis of of the remaining subjects' verbal data. Such commonalities in design behaviour clearly provide support for the theoretical view that designers may possess some form of generic schema or strategy that guides their activity (i.e. high-level procedural knowledge about "how to design").

In the following sections of this chapter, then, an attempt is made to finally draw out the major similarities in subject's design behaviour (e.g. the common use of a problem reduction strategy) which emerged from the data analyses that were undertaken. Furthermore, it seems an opportune time to strive toward some theoretical interpretation of the present findings by the application of theoretical notions that derive

from psychological research on problem-solving and thinking as well as design (see chapter 2). In should be noted that an important aspect of the present general discussion is the attempt to formulate an explicit theoretical model of the cognitive processes which are involved in engineering design, with particular emphasis being placed upon the more global processes that control and coordinate the subject's behaviour in the search for a solution to an overriding design problem.

4.5.2 THE DEFINITION OF REQUIREMENTS DURING THE INITIAL PHASE OF THE DESIGN PROCESS

Earlier in this chapter it was proposed that the major goal of a design project is the production of a description or model of a desired artifact that (1) meets given functional requirements and (2) meets requirements relating to the artifact's performance, resource usage, structural and aesthetic characteristics. It was also mentioned that on route to this goal state it is likely that the designer must (3) meet implicit or explicit requirements that relate to the design process itself (e.g. the design might have to attained within a certain time or under an allocated budget). All of these requirements may be viewed as "constraints" since they restrict the space of possible artifact descriptions that needs to be searched by the designer (see sections 4.3.2 & 4.3.3). In this sense, then, the ***early*** definition of requirements during a design project would seem to be a particularly sensible way for a designer to cut down an almost infinite search space (cf. Anderson, 1985). It was not surprising, then, that most of the engineers in the present study spent much of their initial design time gaining a clearer understanding of the basic requirements of their

desired artifacts. What was particularly noticeable during this initial design phase was the considerable amount of attention that was given to the definition of the *functional* requirements of their intended designs. As was mentioned in section 4.3.3, it generally appears that functional requirements (i.e. requirements which describe what the artifact must "do") constrain the total problem space of potential design objects by the greatest amount, whilst other requirements such as the speed of operation of a device or its cost, if taken independently, tend to reduce the problem space by a less extensive amount. As such, then, it is hardly surprising that the present engineers were seen to (a) initially spends time defining functional requirements and (b) use such functional requirements as predominant factors in the search for solution concepts.

The initial definition and clarification of functional requirements and design constraints that was observed in the present study would seem to be motivated by the need to convert an ill-defined design problem into a more well-defined state before it is possible to search for an actual design solution (cf. Simon, 1973). Not surprisingly, then, this phase of defining the overriding problem appeared to be undertaken by *all* of the present undergraduate subjects, irrespective of whether they were tackling tasks of their own derivation (e.g. NB) or tasks that had been set by a company (e.g. TS) or by a supervisor (e.g. MM and AD). This process of defining a design problem appears to result in the formulation of what may be termed a "basic model" of the problem (i.e. a qualitative constraint-oriented description) which forms the foundation for subsequent solution development processes (see section 4.5.8 below). It is interesting to note here that Larkin (1978) found that subjects

tackling physics problems engaged in a similar phase of problem representation to that observed in the present study of designers. Larkin's subjects were seen to pursue extensive qualitative analyses of a problem prior to generating any principles and equations appropriate to a solution (see section 2.2.5 for further discussion of problem solving in physics).

Processes aimed at problem definition in the form of clarification of requirements have also been observed by researchers who have undertaken empirical investigations of design behaviour. One particular study of interest, which was mentioned in section 2.3.2, is that reported by Malhotra et al. (1980). These researchers present some results from an observational study in which designers and clients were videotaped whilst engaged in problem-oriented dialogues. These complex dialogues revealed an essentially cyclical structure involving sequential stages of (1) analysis, understanding and development of requirements (2) outlining of partial solutions to satisfy requirements and (3) development and evaluation of partial solutions. During this third stage, it frequently appeared that subjects uncovered new requirements during the examination of solutions and these requirements often initiated a new design cycle. Malhotra et al. term the initial stage of the design cycle "goal elaboration" and observed that once goals had become well developed, solutions to the problem appeared to follow from these quite readily. It should be noted, however, that these authors attempt no theoretical interpretation of their results in terms of underlying cognitive mechanisms.

4.5.3. PROBLEM REDUCTION STRATEGIES IN ENGINEERING DESIGN

Subsequent to the initial definition of the functional requirements and constraints of their desired artifact, the present subjects were seen to use this information as a basis for defining a complete abstract solution for the overriding design problem. It is suggested that in searching for this high-level solution, subjects were using constraint information (particularly relating to functionality) to obtain a "modular" description of the solution. Such solution modules, then, would each satisfy one or more aspects of the artifact's functionality and would exist as manageable and minimally interacting subproblems which could then be attended to separately as essentially independent design problems in their own right. Indeed, once selected for further development, each of these subproblems tended to be focussed on sequentially with a detailed solution being developed depth-first to completion.

This process of decomposing an overriding design problem into an integrated set of slightly more concrete modules clearly indicates that a form of "problem reduction" (or subgoaling) strategy was being instantiated by the present subjects. As has been stated previously (see section 4.4.2), the adoption of a problem reduction strategy can be viewed as involving a complex high-level design problem being "divided and conquered". It would seem that this problem reduction strategy is central to the successful derivation of hardware designs as it appears to serve to break down complex and unmanageable design problems into minimally interacting subproblems of component module design. Other researchers have suggested the use if this strategy in architectural design (e.g. Simon, 1973) and in software design (e.g. Jeffries et al,

1981) though this appears to be the first empirical evidence for its use in the design domain of electronic engineering.

Interestingly, if a solution that had been generated to any particular subproblem was itself understood to be complex, then this solution was again seen to be decomposed into sub-subproblems of submodule design (reflected by the subject setting goals of the form "design submodule X1" and "design submodule X2" and so on). Again, then, each of these sub-subproblems would be developed one at a time in a depth-first manner. Clearly in applying this problem reduction strategy, the attainment of any high-level design goal was crucially dependent upon the full set of subgoals and sub-subgoals at lower levels of the goal tree being attained. Sometimes, too, blocking conditions or difficulties would be seen to be prevent subjects from pursuing or attaining design goals at any level of the goal tree, and in these situations subjects would again set subgoals aimed toward the removal of the difficulties.

It is worth briefly pointing out that the ***initial*** search for <u>abstract</u> solutions to high-level design problems (i.e. top-down design) which was evidenced in the present study, may be regarded as a very important heuristic which again serves to narrow down an enormous search space of detailed solutions (cf. Bowen, 1985). The point is, an abstract solution to a problem represents a whole "class" of detailed solutions, so searching a space of abstract solutions is ***equivalent*** to searching the much larger space of detailed solutions, but clearly much less expensive in terms of the time and effort required.

4.5.4. STRATEGIES FOR MANAGING DIFFICULTIES AND DESIGN FAILURES

During their projects the present subjects often reached points where their actions were being thwarted by difficulties such that the continuing pursual of some aspect of the design or its realisation was being hindered or prevented. Difficulties that the subjects mentioned were seen to arise because of a multiplicity of reasons, such as lack of information, inadequacy of resources, and non-familiarity with a particular skill.

Analysis of protocols revealed that most difficulties could be dichotomised in terms of whether they were being caused by "external" or "internal" factors. External causes of difficulties were the occurrence of events in the subject's environment (e.g. equipment failure, or non-availability of componentry, information or people) whilst internal causes of difficulties were seen to arise because of events peculiar to the actual person (e.g. the lack of a particular skill or the application of an inappropriate heuristic). One clear example of an external difficulty that was affecting MM's design progress was the lack of suitable components - as exemplified by the following comments:

> "Most [components] seemed OK for the job, but one in particular is holding me up, the 'constant-current oscillator' - to produce the current to pass into the eye. Apparently there is not one available."
> (Diary 29/10 to 4/11)

Another difficulty that MM was having appeared to be much more internal in origin as can be seen from the following comment:

> "One quite large problem occurred when trying to get a constructional layout on to paper from the various circuit diagrams of the parts of the circuit. They look quite simple in circuit diagram form, but when all connections have to be considered, it is quite a daunting task."
> (Diary 25/11 to 1/12)

Clearly, difficulties could mount one on top of the other to such an extent that the a whole project could reach a temporary standstill until some or all of the difficulties could be eradicated or circumvented. Indeed such situations were seen to occur quite often for some of the subjects studied - particularly MM and to a lesser extent TS. What is interesting in relation to difficulties that were arising during the present subjects' design projects was the nature of the strategies that were used for coping with them. The most frequently observed method of handling a difficulty appeared to involve the subject setting and pursuing a knowledge acquisition goal. Often the knowledge acquisition goal would be of a form that required the reading of technical literature and documentation such as data sheets, textbooks, journal articles, or handbooks. Frequently, too, knowledge acquisition goals were seen to be set that involved seeking help from a supervisor or some other domain expert. Whilst these information seeking strategies were generally seen to be successful in enabling subjects to tackle and overcome difficulties that had arisen, this was not always the case. When however, it occurred that one knowledge acquisition goal had failed in overcomming some difficulty, it was usual for another such goal to be set until eventually the path toward attainment of the design goal was unobstructed.

4.5.5. PROBLEM MODELS AND SOLUTION MODELS IN ENGINEERING DESIGN

Within the present study the models of problems and solutions that were being developed tended to be externalised in a multiplicity of ways including (a) block diagrams (b) flow diagrams (c) circuit diagrams (d) timing diagrams and (e) written notes. The favoured external

representation of the present engineers, though, tended to be the block diagram, where blocks generally depicted the hardware modules (i.e. objects having functionality and which satisfied other desired constraints) of the intended artifact, and lines (often arrowed) symbolised the connectivity between these modules in terms of information flow. Arguably, the many overt representations of problems and solutions that were generated by the present designers may be thought of as the external analogues of "mental models" that were built up and manipulated as internal representations. Interestingly, these mental models themselves may be thought of as being analogues of real states of affairs (i.e. hardware implementations) that "could" exist in the world (see section 2.3.3 for a discussion of the mental models notion). In design situations, then, mental models would presumably precede any external modelling (e.g. on paper) by the designer. The advantage of producing physical models no doubt arises from the fact that they form an "external memory" for the designer, thus freeing limited mental resources for other processing activities including the construction and manipulation of new mental models. It should be noted in this regard that it is generally assumed that mental models are constructed, manipulated and maintained within some form of limited-capacity working memory system (see, for example, Johnson-Laird, 1983). Whilst a mental model could no doubt be converted into a more durable form such as is characteristic of long-term memory, the conversion process would presumably take more processing time and effort than is needed to simply externalise the model as, for example, a block diagram sketch.

Clearly, the block diagram and its mental equivalent appear to be fairly

static types of representations. It is, however, conceivable that the designer could mentally simulate the actual dynamic behaviour of a device represented as a block diagram, by, for example either (a) running hypothetical data (e.g. voltages or data packets) through the interconnected system to determine outputs at various points or (b) engaging in cause-effect reasoning about subcomponent interactions. These kinds of simulation processes have been termed "envisioning" by de Kleer (see, for example, de Kleer & Brown, 1983; de Kleer, 1984). It should be noted that whilst the present diary method failed to pick up the occurrence of such processes, the interview sessions revealed that the designers could readily engage in such envisioning (e.g. when they were explaining the operation of a device). The following segment from TS's second interview may serve to illustrate - albeit at a high level of abstraction - this mental simulation of a device's behaviour:

> "So when it goes off into its subroutine, if it goes off for longer than - I think it's a few microseconds - I'm not sure of the exact time but it is a fairly short time .. - it's about eight cycles of a clock - then the DMA says "Right I've finished, I obviously haven't got anything else to do, I'll give the bus back to the system processor" and so it looses all control and you can't get any more data without stopping the system processor again and then going back to the start location and going through the table." (Interview 9/12)

This latter kind of mental simulation of the operational characteristics of design solutions would seem to be particularly useful for comparing and evaluating the viability of a design option. Moreover, if a difficulty with a design solution emerged from such a simulation it is likely that the designer could alter the model to produce an improved solution. Mental simulation has also been suggested by other researchers in connection to the processes involved in software design (see Adelson & Soloway, 1986, for some evidence for mental simulation of program behaviour).

4.5.6. TECHNICAL MODELS AND SOCIAL MODELS IN DESIGN

Continuing with the theme of mental modelling in design, one interesting aspect of the present engineers' behaviour was that they appeared to be developing two distinct - but interacting - mental models during their projects. The first was an expanding ***technical model*** of the artifact that was being designed, the elements and constraints of which - at all its levels of detail - were defined entirely in terms of the domain (digital electronics, analogue electronics or whatever). The second was what may be termed a ***social model,*** in which the designer viewed his or her design behaviour in a context constrained by people. It was clear from TS's diary and interview comments, for example, that he was seeking to fulfill the demands of both his supervisor as well as those of the company with which he was collaborating (see verbalisations below). That is to say, he was aware of real world constraints such as the needs of others, including - as the example below indicates - the people who would eventually have to program the chip he was designing:

> "if you had memory management, you could say 'Right, that's at that location in the big storage which is slow to access, and right, we can dump it over here but call it the same memory location and then everything can refer to the same memory locations.' That makes things a lot more simple for the software people." (Interview 9/12)

The interaction between social and technical models was manifest at many times during the present projects and generally took the form of design constraints of a social origin restricting technical design possibilities. Indeed in this regard it is clear that it is only the fact that an artifact is designed both in and for a social context which leads to the imposition of any constraints whatsoever on a functional design. The point is, without such constraints - which, in professional

settings usually derive from client specifications - "anything" would possible within the technical limitations of the domain (i.e. the device could be as expensive, large, ineloquent, or unmarketable as the engineer desired). In reality, of course, if externally imposed constraints are not present then the engineer would impose his own "implicit" social constraints of what a "good" design should be, based on his or her own knowledge of the intended context in which the artifact will function and the end users who would be subjected to it.

Returning to the present subjects' design work, then, a few further examples of comments from TS may suffice to indicate the manner in which constraints of a social origin tended to impinge upon a subject's technical design ideas. The following comment, for example, shows that TS was deriving important design constraints from the company which related to technical design solutions that he would have to attain:

> "I am aiming [at least] for an exact pin for pin copy [of the DMA] +++ I'm [even] hoping to cut down on the number of pins that they're using +++ thus making it a bit of a cheaper device. +++ I'm designing and talking to them on the basis that when I've finished they can just plonk it on to the system that they've got and it will work first time or without extensive software modifications, because they are actually writing a big microcode for this system +++" (Interview 9/12)

Yet another comment made by TS revealed that he had generated interesting technical ideas which could not be pursued because they didn't further the goals of the the company he was collaborating with:

> "I haven't started this second phase - which was the idea of more control - and that's because I need the information from C----- [i.e the company]. Having said that, I've thought about it, because we [i.e. TS and his supervisor] came up with the idea that you need control status registers +++ That's really going to be phase two [BUT] I've sort of discovered that I have got to really keep to what they want and not expand my ideas too much as far as they are concerned ... I don't think they would be interested.." (Interview 9/12)

The influence of factors and constraints arising in the present

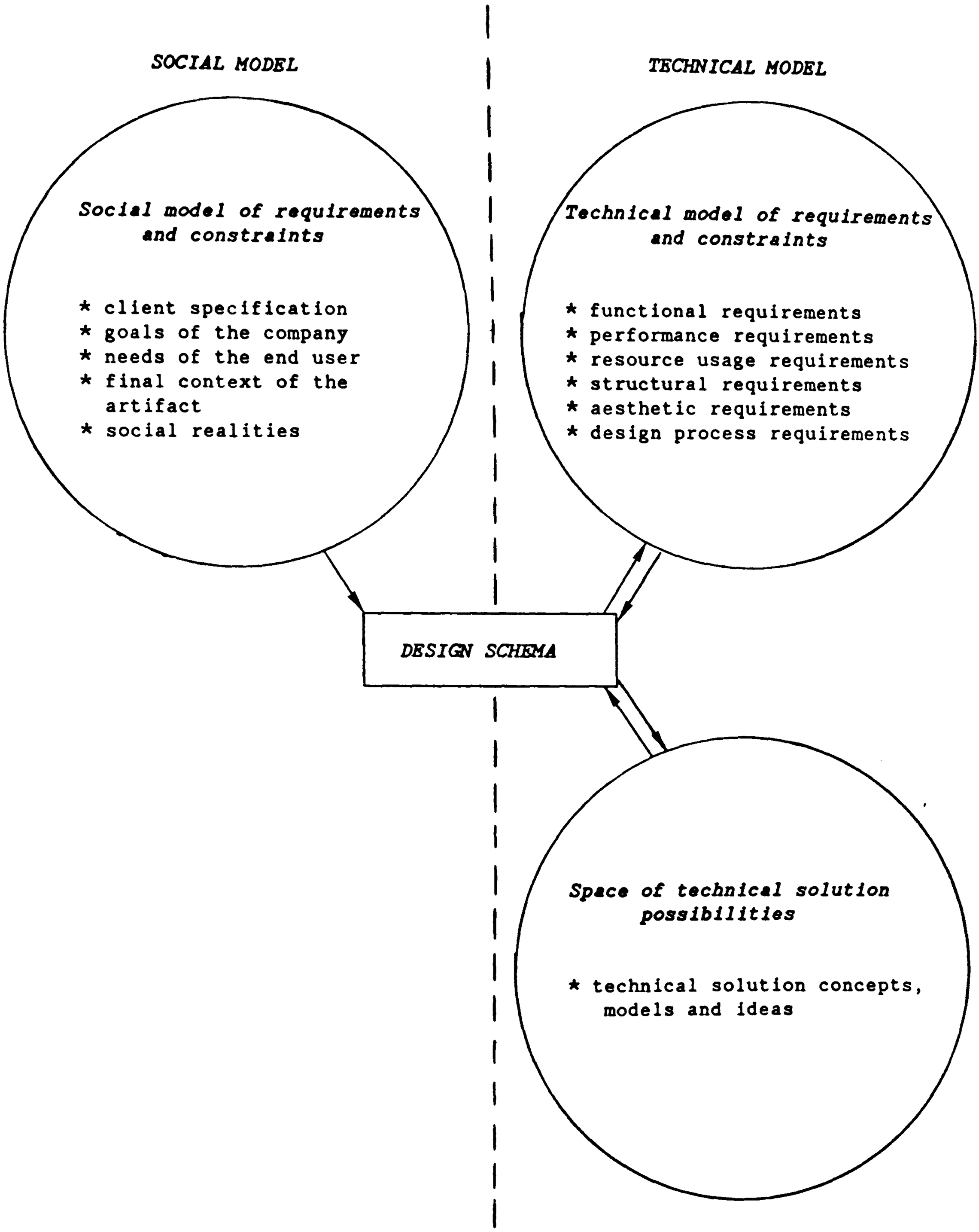


Figure 4.6: Mapping between the social model and the technical model via the design schema. (Externally imposed requirements and constraints are mapped onto the technical model of the problem by the design schema which in turn controls the search for technical solution concepts).

designers' social environments was so pervasive that examples of social-technical mappings could be found on almost every page of their verbal reports. In theoretical terms, then, whilst technical and social models are conceptually distinct, it is clear that (a) some high-level schema must exist which permits mapping between the two to occur and (b) that it is absolutely essential for the attainment of successful design solutions that this mapping is made. In this regard it is suggested that a high-level "design schema" exists which provides the essential mechanism for attaining the connection between social and the technical models (see sections 4.5.8 for a detailed discussion of the design-schema concept). Figure 4.6 below depicts - in schematic form - the essence of the present proposal. The figure shows how information residing in two conceptually separate mental models may be translated form one to the other via the design schema.

4.5.7. DECISION-MAKING PROCESSES IN ENGINEERING DESIGN

Decision making is commonly conceptualised as involving a number of interacting processes aimed at the goal of selecting one option or alternative from a range of options that have been generated (see, for example, Wright, 1984). One important process in decision making is that of ***evaluation,*** which involves analysing information relevant to the competing options (e.g. concerning properties or attributes that they possess) so that a choice may be made amongst them. Many evaluation strategies have been proposed to mediate people's decision making in situations where both (a) a set of identifiable alternatives exist and (b) information relevant to the attributes of the alternatives is available (see Montgomery & Svenson, 1976; Jacob et al., 1986, for some

discussion of strategies in human decision making).

Of particular relevance to the present research were the decisions that the present subjects made concerning solution concepts (e.g. hardware devices, electronic circuits, components and the like). Examples of decisions which related to technical solution concepts - made by respectively NB and TS - are illustrated below:

> "A simple RC network providing the input to a buffer, with T = 20ms, would be sufficient." (Diary 5/11 to 11/11)

> "Looking at data books for I/O ports and their implementation has led to my design of I/O ports using latches and tri-state buffers." (Diary 16/12 to 13/1)

From comments like the above, however, it was discernible that whilst the designers readily ***stated*** the design concept that they had chosen (i.e. the product of their decision making) they generally provided no detail of the actual ***processes*** by which a choice had been reached (see NB's comment above, for a typical example of this phenomenon). Moreover, it was not even evident that these designers had actually generated alternative and competing design concepts in the first place. That is to say, these subjects (a) rarely commented that more than one single solution concept had been considered for a problem and (2) even when they stated that such alternatives ***had*** in fact been considered they generally failed to provide any examples of the options that they had chosen amongst.

There appear to be two ways of accounting for the limited generation of alternatives that was apparent in the present study. Firstly, it is possible that subjects were applying "explicit" processes for generating and evaluating alternatives but that in producing their retrospective diary accounts they were failing to comment on these explicit procedures

that had been adopted (e.g. through forgetfulness). Alternatively, it is possible that subjects were not engaging in any explicit generation and selection of alternatives but were rather pursuing rapid and intuitive evaluations of competing options. These intuitive selection processes would presumably not be open to verbal report because of their "implicit" or pre-attentive nature (see Evans, 1989, for a discussion of implicit and explicit processes in thinking and reasoning). Obviously, then, it is difficult to determine which of these two distinct possibilities is the correct one because of the frequently retrospective nature of the diary reports that were made (see section 4.6.2 for further comment on this point). Certainly the issue might more easily be resolved by the use of concurrent "think aloud" techniques (see section 3.1.2) which would be likely to reveal the occurrence of any explicit processing in the generation of solution concepts.

Whilst, then, it appears difficult to determine the underlying nature of decision-making processes by means of the present data set, there is one point - which relates to the objective nature of design solutions - that gives grounds for the belief that the present subjects' generation and evaluation of alternative solution concepts and models was occurring at an implicit, and perhaps rather superficial, level. The point is, that a design solution (e.g. an interconnected set of components) will often be both complex and multi-faceted such that the only way to determine its superiority over another solution is by simulating the behavioural characteristics of the two and comparing the results. Now, whilst this simulation process could conceivable be undertaken mentally by the person (either implicitly or explicitly), it is likely - certainly in the case of complex, high-level solution models - that the cognitive

demands of such a process would require the use of ***external*** memory aids (e.g. pen-and-paper) or even require recourse to CAD tools (see also section 4.5.5 for discussion of this point). TS, for example, frequently undertook sophisticated timing analyses of solution ideas on the CAD system that he was using. What was noticeable, however, was that this subject - like his fellow designers - <u>never</u> reported pursuing such behavioural simulations of substantially alternative solution models.

It appears, then, that there is at least some evidence for the suggestion that the present subjects were not expending cognitive effort in the search for and modelling of alternative - and potentially optimal - design solutions, particularly at the higher-level stages of their design work. Indeed it is proposed that these designers' solution-generation behaviour may have been dominated by a principle that is neatly captured by Simon's notion of "satisficing" (see, for example, Simon, 1981 for a discussion of this notion in the context of design). The principle behind satisficing appears to be that the designer will search for and accept as satisfactory, any solution that is "good enough" (e.g. fast enough in operation or cheap enough to implement). The use of this principle is certainly one way to narrow down a large search space, since all that is required is a satisfactory solution - rather than one that is better than satisfactory or even optimal.

Interestingly, there is perhaps an additional and more insidious side to this satisficing principle which is suggested by observations made in the context of the present study. That is to say, when the present designers were seen to generate a solution which soon proved to be less than satisfactory (something which occurred quite frequently) they

appeared to be loath to discard the solution and spend time and effort in the search for an alternative. Instead, the subjects appeared to stick almost religiously with unsatisfactory solutions and subsequently to develop them gradually to completion by the production of a variety of slightly improved versions. Generally, then, it appears that the gradual iteration and improvement of solution concepts through levels of increasing detail that was exhibited by these designers does not reflect an optimisation tendency but rather indicates a fixation on initial concepts and the attempt simply to make them satisfy the problem requirements.

4.5.8. CONTROL OF THE DESIGN PROCESS: THE DESIGN SCHEMA

Although much of the discussion so far has presented an account of design as being a highly structured sequence of controlled and coordinated behaviour, this is not to say that design behaviour is inflexible and unadaptive. Indeed one interesting aspect of the present engineers' behaviour was that it could readily be adjusted to cope with sudden social demands and unforeseen design failures or difficulties. Behaviours were sometimes seen to change in relation to earlier planning as a result of either (a) the modification of goals (b) the abandonment of goals or (c) the rescheduling of the order of the planned activities. Any theoretical model of cognitive processes in design must therefore be able to capture not only the fundamental structuredness apparent in design behaviour but also the flexible nature of these behaviours.

The guiding control processes that give design activity its form, whilst at the same time preserving an element of flexibility, are arguably best captured by the notion of a "design schema". The design schema (refer to

section 2.3.2) can be though of as a knowledge structure composed of both procedural and conceptual knowledge which is built up as a result of the engineer's experience with design problems. The design schema may, therefore, be viewed as providing a basic "strategy" for design which can be applied to a range of problems which have similar fundamental structures but which differ in terms of their domain content. When applied in a design context, then, the design schema would control the decomposition of the overriding problem into subproblems relating to the design of component modules of the desired artifact and would also coordinate the order in which these subproblems were worked on.

As was noted in section 4.5.3, it was apparent that the present designers were tackling design subproblems in a top-down, depth-first manner. However, this leaves unanswered the question of why one subproblem at a particular level of the problem hierarchy would be given priority over subproblems existing at the same level. Looking at the design data provides some plausible answers to this question. For example, the reason why NB first tackled the design of the tape motion sensor seemed clear, i.e. he realised the sensor might need to be constructed by someone other than himself and might be held up in being built; it therefore was vital to get a design sorted out quickly to prevent any subsequent delays which might result in an incomplete project. Since subjects in the study generally provided little explicit information regarding the rationales for tackling design problems in the particular orders that were manifest, it is only possible to guess at plausible determinants of the observed sequences of activity . One strategy, for example, might be to deal with subproblems in an order

that reflects increasing degrees of their perceived importance. Other strategies might be based on the perceived difficulty of design subproblems, such as "easiest-first" or "hardest-first" orderings. TS, for example, almost certainly pursued an "easiest-first" ordering for the design of problem modules. In general, however, what is most likely to be the case is that the sequence of design work is a function of both a preconceived strategy as well as the current state of the social context in which the design is proceeding. Thus, for example, an easiest-first ordering that a subject has instantiated while engaged in the ongoing design of an artifact may have to be adapted if at some point a supervisor or a collaborating company requests that the design of a difficult module be given immediate consideration.

With regard to the design schema it is finally worth pointing out that many of its functions must clearly relate to the search, retrieval and evaluation of technical solutions as well as the initiation of further subproblem decomposition if this is necessary (e.g. if an abstract solution module is understood to be complex and multi-faceted). With respect to the search and retrieval of technical solutions it is quite likely that content-dependent design schemas exist which are specific to the attainment of concrete solutions to design problems and which are presumably cued by detailed information present within the current model of the subproblem (cf. Soloway & Woolf's 1980, notion of templates in computer programming). In the event of such a technical schema being invoked, it is conceivable that the immediate control of behaviour (i.e. relating to the construction of a design solution to the current subproblem) would be temporarily given to this schema until the solution was attained. At a later point control of activity would be returned to

the overriding, general-purpose design schema. Clearly the mechanism by which this exchange of control might be carried out between a hierarchical organisation of embedded design schemas needs careful consideration. The theme, however, is not taken up further here, for it is difficult to gain much insight into the issue from the verbal data obtained in the present study since most comments tended to relate to high-level plans and activities rather than to the fine-grained aspects of design work.

An attempt has been made (see below) to summarise the design schema for undergraduate engineering projects as a set of abstract production rules in a manner that partly resembles the software design schema presented by Jeffries et al. (1981). A production rule representation of a designer's procedural knowledge seems to be a particularly good way of capturing the flexibility that is apparent in design behaviour as well as its fundamental organisation and structuredness. One salient aspect of the present engineers' design schema that has been encapsulated within the following production rules is the essentially iterative nature of design activity. That is to say, the rules have be written such that they could be applied not only to the overriding design problem but also to any subsequent subproblems, and so on, until a final, detailed design solution is attained. It is worth noting, however, that unlike Jeffries et al.'s schema, which left the control structure of the software design process as an implicit aspect of the production rules, the schema presented below expresses the control structure for engineering design as an explicit set of "steps". Depicting the design schema in such a way clearly has the advantage of making the control strategy for design activity more readily accessible

to the reader than could be afforded by conventional representations of production rules. Subsequent to the presentation of the schema some fairly detailed explanation is provided of its functional characteristics.

DESIGN SCHEMA FOR UNDERGRADUATE ENGINEERING PROJECTS

PHASE 1: PROBLEM SELECTION AND PROBLEM DEFINITION

STEP 1

R1 IF (no current problem exists)
AND (problems on agenda)
THEN (select highest priority problem and remove from agenda)
AND (make highest priority problem current problem p)

STEP 2

R2 IF (p is current problem)
AND (basic model of p does not exist)
THEN (set goal to develop a well-defined basic model of p)
ELSE (move to STEP 8)

STEP 3

R3a IF (goal is to develop a well-defined basic model of p)
THEN (use own technical knowledge to expand understanding of functional requirements and constraints expressed in p at current level of abstraction)

R3b IF (Rule R3a is successfully applied)
THEN (assert that p has a well-defined model)
AND (move to STEP 8)

STEP 4

R4a IF (goal to develop well-defined basic model of p)
THEN (seek technical advice of person who set the problem so as to expand understanding of functional requirements and constraints expressed in p)

R4b IF (Rule R4a is successfully applied)
THEN (assert that p has a well-defined model)
AND (move to STEP 8)

STEP 5

R5a IF (goal to develop well-defined basic model of p)
THEN (seek technical advice of a domain expert so as to expand understanding of functional requirements and constraints expressed in p)

R5b IF (Rule R5a is successfully applied)
THEN (assert that p has a well-defined model)
AND (move to STEP 8)

STEP 6

R6a IF (goal to develop well-defined basic model of p)
THEN (use technical literature to expand understanding of functional requirements and constraints expressed in p)

R6b IF (Rule R6a is successfully applied)
THEN (assert that p has a well-defined model)

STEP 7

R7 IF (p does not have well-defined model)
THEN (fail and exit)
ELSE (set goal to develop solution model for p)

PHASE 2: SOLUTION MODELLING FOR PROBLEM

Phase 2a: Developing and assessing solution model for p

STEP 8

R8a IF (goal to develop solution model for p)
THEN (search for modular solution concepts at next level of design detail directed by information about important functional requirements and constraints in basic model of p AND generate AND integrate AND externalise solution concepts for p)

R8b IF (application of R8a results in solution model for p)
THEN (set goal to evaluate solution model for p)
ELSE (fail and exit)

STEP 9

R9a IF (goal to evaluate solution model for p)
THEN (check that solution model satisfies functional requirements and constraints as expressed in basic model of p AND ensure potential connectivity of solution model to other solution models at same level of design detail)

R9b IF (Rule R9a is successfully applied)
THEN (assert that solution model for p is satisfactory AND move to STEP 13)

STEP 10

R10a IF (solution model for p is unsatisfactory)
THEN (use understanding of solution model for p to assess viability of refining unsatisfactory solution model into satisfactory solution model for p)

R10b IF (Rule R10a is successfully applied)
THEN (assert that solution model for p is viable)

STEP 11

R11 IF (solution model for p is both unsatisfactory and unviable)
THEN (reject solution model)
AND (augment basic model of p with understanding derived from rejected solution model for p)
AND (set goal to develop solution model for p AND return to STEP 8)

STEP 12

R12 IF (solution model for p is unsatisfactory but viable)
THEN (use understanding of functional requirements and constraints expressed in basic model of p to search for new solution concepts that satisfy failed aspects of solution model for p and generate AND integrate AND externalise solution concepts for p)
AND (set goal to evaluate solution model for p AND return to STEP 9)

Phase 2b: Complete processing of p, generating sub-problems as necessary

---

STEP 13

R13 IF (solution model for p is satisfactory)
AND (solution model for p is not at most detailed design level)
AND (solution model for p exists as two or more modularised solution concepts)
THEN (use understanding of problem and solution to decompose solution model into separate subproblems which relate to the design of separate modular concepts AND use understanding to give priority ratings to these subproblems)
AND (add subproblems that are identified to agenda)
AND (assert that no current problem exists)

STEP 14

R14 IF (solution model for p is asserted to be satisfactory)
AND (solution model for p is not at most detailed level)
AND (solution model for p exists as a single modularised solution concept)
THEN (treat solution model for p as expressing a problem)
AND (add new problem to agenda)
AND (assert that no current problem exists)

STEP 15

R15 IF (solution model for p is asserted to be satisfactory)
AND (solution model for p is at most detailed level necessary for the present application)
THEN (assert that no current problem exists)

STEP 16

R16 IF (no problems on agenda)
THEN (succeed and exit)
ELSE (return to STEP 1)

INTERRUPT RULE

```
IF (external social requirements demand that a problem
    other than current p to be given high-priority status)
THEN (halt progress on current p)
AND (delete p as current problem and add p to agenda)
AND (set high-priority status problem as current problem p AND
    return to STEP 2)
```

Now that the rules which were considered to underly the present undergraduates' design approach have been detailed it would seem sensible to explain some of the intended processing characteristics of each of these rules. Firstly, then, for the schema to allow the engineer to get working at the task of developing a solution to a design problem, some rule needs to exist which can select an actual problem to focus on. Normally the initial problem that would be selected in a design situation is the overriding problem that is set either by an external agent (e.g. project supervisor or industrial party) or by the innovative designer him/herself. After the initial decomposition of this overriding design problem, however, it is likely that multiple subproblems will exist that each need to be tackled and solved. It is assumed, therefore, that all such unsolved problems are listed on a problem agenda (cf. Jeffries et al. 1981) in an order determined by a previously set priority rating (see R13). If no problem is currently being tackled, then the problem-selection rule (R1) results in the highest priority problem being removed from the agenda and being marked as the current problem p.

Subsequent to the selection of a problem, R2 is invoked if a basic model for the problem does not exist - which would be the case, for example, if the problem was an un-read design specification. R2 simply sets a

goal to develop a well-defined basic model of the problem. The term "basic model" has been coined here to refer to a well-defined mental representation of the current problem that the engineer *must* possess in order to attempt to develop an actual technical solution to the problem (see also section 4.5.2). To be useful in guiding the search for solution concepts, a basic model needs to exist as a structured description of all constraints that the desired artifact (or module of the artifact) must satisfy. During the development of a basic model, then, particular emphasis would normally be placed on the structured definition of the functional requirements expressed by the problem (see rules in steps 3 to 7). R3a is the first rule to be invoked if a goal has been set to develop a basic model of the problem and this rule leads the designer to use his or her own technical knowledge to understand the functional requirements and constraints expressed within the problem. During this processing step not only is the understanding of existing requirements and constraints refined but also new requirements and constraints are likely to be added to the problem. It should be noted, however, that requirements and constraints are seemingly most useful when defined at the "current" level of design abstraction since this presumably restricts the amount of information that the designer has to hold in mind when searching for solution concepts (something that is clearly important because of working memory limitations). The point is that abstract requirements and constraints subsume many detailed requirements and constraints whilst at the same time being much easier to hold in mind than the multiplicity of detailed requirements and constraints which they subsume.

If a designer's attempt at developing a basic model of the problem by

the application of technical knowledge has failed such that the problem is still in an ill-defined state, then there may be other possible routes for attaining a well-defined basic model. R4, R5 and R6 represent three possible ways in which the designer may obtain a basic model by respectively (1) communicating with the person who set the problem in the first place (i.e. company contact or supervisor) or (2) seeking and obtaining the advice of a domain expert (e.g. an academic contact) or (3) seeking out and examining relevant technical literature. All three of these methods were commonly seen to be used by subjects within the present study in order to attain further information about their design problems and subproblems. Certain rules, however, appeared to be more dominant than others as determinants of different subjects' behaviour, for example, subject NB seemed rather more independent a designer than many of his cohorts who more readily sought advice from other people.

If the problem definition phase for a current problem is complete such that the problem's functional requirements and constraints have been fully represented in a structured manner, then the design schema now takes the engineer into a phase of solution modelling (see R8a). A solution model may be either a single design concept or a set of distinct but interrelated design concepts which describe "objects" (i.e. modules, units or structures) that attain the requirements and constraints expressed by the problem. Additionally, it should be noted that a solution model may be generated at any one of many different levels of design detail ranging from the abstract through to the concrete. However, it is assumed that the designer will always search for solution concepts at only *one* level more concrete than the level at which the design problem exists (i.e at which the functional

requirements and constraints are expressed). To repeat the point made in section 4.5.3, the top-down development of designs may be regarded as a very important way of limiting the time and effort required to search a potentially vast problem space of design solutions. The point is that searching a space of abstract solution concepts each of which represents a whole "class" of detailed solution concepts is (a) logically equivalent to searching the much larger space of detailed solution concepts but is (b) clearly much less demanding on the cognitive system.

When invoked, then, R8a leads to the search for solution concepts cued by the requirements/constraints that exist within the basic model of the current problem. In the present study, certain types of constraint were frequently seen to be used by subjects to guide the search process. Functional requirements appeared to dominate the engineers' search for solution concepts whilst cost requirements and performance requirements (e.g. a device's speed of operation) appeared to be important secondary constraints. It should be noted that for convenience the term "search "has been used in R8a in a way that applies equally to a mental search (i.e. through domain concepts) or to an external search (e.g. component catalogues or design data books). In reality, of course, separate rules would be invoked to instantiate these two different kinds of search process. When solution concepts are retrieved that are relevant to the current problem R8a additionally ensures that they are integrated and also externalised (e.g. represented on paper).

If a solution model for a problem has been successfully generated then the next schema step (step 9) leads to the evaluation of this solution model. R9a of step 9 results in activities aimed at (a) checking whether the solution model satisfies the functional requirements and constraints

that were expressed in the basic model of the problem and (b) ensuring that the solution model can interconnect sensibly with other solution models that have previously been derived and which exist at the same level of design abstraction as the solution model for the current problem. The successful application of R9a results in the solution model for p being deemed as "satisfactory" with a resulting jump to STEP 13 of the schema (this step is discussed below). A failed application of R9a, however, leads to a direct move to STEP 10 of the schema where understanding processes are applied in order to determine whether the "unsatisfactory" solution model for p is a potentially "viable" solution that might yet be reworked into a satisfactory solution. If the unsatisfactory solution is considered at step 10 to be viable (i.e. it is perceived to need only some small amount of alteration) then R12 will be invoked which results in a search being initiated for new and improved solution concepts for the problem. On the other hand, if at step 10 the unsatisfactory solution fails to be seen as "viable", then R11 results in the solution model being abandoned. R11 ensures, however, that any understanding of the current problem that has been derived in the development of the failed solution model is used to augment the basic model of the problem and this rule finally sets a goal to develop a new solution model for the problem (involving a return to step 8 of the schema).

Steps 13 to 15 of the schema encapsulate certain rules which are critically important in completing the processing of any solution model for a problem. Looking at step 13 first, then, if a solution model that has been developed for the current problem (a) is asserted to be satisfactory (b) exists at some high or intermediate level of design

abstraction and (c) exists as two or more modularised solution concepts (e.g. the model might be represented as a functional block diagram) then R13 is instantiated to reduce the solution model into a set of interdependent and minimally interacting subproblems and to add these subproblems to the problem agenda. If the same conditions that exist for R13 are present with the exception that the solution model exists as a single solution concept, then R14 is instantiated which leads (a) to the solution concept being treated as expressing a new problem and (b) to the subsequent addition to this problem to the problem agenda.

R13 clearly lies at the heart of the whole design schema in that it captures the process of "problem reduction" which produces a complete new set of subproblems to be tackled and solved by means of a the iterative application of the schema. As has already been stated, all newly generated subproblems are viewed as being placed on a problem agenda. It is further assumed, however, that such subproblems are placed on the agenda by the designer first giving them priority ratings and then adding them to the "end" of the agenda in reverse order of their priority. Priority ratings might be given to subproblems based on technical considerations or social considerations or a mixture of both. For example technical considerations might lead to "easiest-subproblem-first" or" most-important-subproblem-first" rankings of subproblems whilst social considerations might lead the designer to work first on a subproblem that a supervisor believes to be the most crucial part of the design. It is finally important to note that subproblems are assumed to be removed from the end of the agenda (and made the current problem) in a "last-on-first-off" ordering. This has the consequence of always leading the production system "depth-first" in its problem selection and

solution development - though the interrupt rule (discussed below) does allows for possible deviations from this regime.

R13 and R14 both deal with the complete processing of solution models that exist at levels of design abstraction which are higher than the level required for the current design application (i.e. as stated in or inferrable from the the original design specification). R15, on the other hand, functions to prevent any further processing of solution models that *have* been taken to the requisite level of design detail. R16 simply ensures that the design schema is re-applied if any unsolved problems are remaining on the problem agenda. Finally, the interrupt rule - already mentioned above - represents one of possibly several ways in which some divergence from the general depth-first pursuit of designs may occur. The interrupt rule allows for progress to be halted on a subproblem that is being pursued if social demands require some other subproblem be urgently attended to. This might be the case, for example, if a supervisor expressed an opinion that the designer should pay attention to a crucial but as yet untouched module of the design.

To conclude this section, then, it is worth making a few final comments that relate to the schema for undergraduate engineering design that has been proposed. The main point to emphasise is that the rules which have been devised should in no way be viewed as representing anywhere near all of the types of processes that occur during the course of a design project. Rather, these rules should be viewed as an attempt to capture only certain salient aspects of the high-level processes that control and coordinate (a) the decomposition of the design problems into subproblems (b) the order in which subproblems were worked on and (c) the definition of constraints and requirements expressed by subproblem

(d) the search for and retrieval of solution concepts and (e) the evaluation of solution concepts. Clearly many aspects of the design schema have not been addresses at all, such as the processes which lead to comprehension of design problems, the processes which lead to the generation and comparison of of alternative design concepts or the processes that coordinate the storage of models for later use.

4.6. CONCLUDING COMMENTS

4.6.1. CONCLUDING SUMMARY OF FINDINGS

The study that has been presented in this chapter constitutes one of the few attempts that appear to have been made to derive a systematic understanding of cognitive processes in engineering design. Though exploratory in nature, the investigation has arguably produced a variety of quite interesting and pertinent findings concerning the nature of design processes. Whilst there is certainly little point reiterating the full set of findings in their fine-grained detail here, it nonetheless seems worthwhile briefly summarising the more salient aspects of the results, since this will usefully serve to set the scene for the second investigation that was undertaken within the current research programme (see chapter 5). This summary of findings is undertaken below in point form.

(1) The student designers that were investigated in the present study appeared to spend a significant part of their early design work developing a clearer understanding of their overriding ill-defined design problems. During this problem-definition phase subjects were seen to be especially concerned with developing a structured representation

of the functional requirements of the desired artifact as well as other kinds of constraints that had to be met (e.g. relating to the artifact's speed of performance or cost). The process of problem definition for the overriding design problem (or indeed for any subsequent subproblem) was considered to result in the construction of what was termed a "basic model" of that problem, and this basic model was then seen to form the foundation for the application of subsequent processes of solution development.

(2) Having gained a clearer definition of their overriding design problems, the present subjects were then seen to produce a high-level design solution which could meet the functional requirements and constraints expressed by the problem. This abstract solution model was invariably seen to be externalised as a set of essentially independent though interconnected hardware modules that were depicted in block diagram form. Subsequent to the generation of an abstract solution model for the overriding design problem, the engineers were seen to make consistent use of a ***problem reduction*** (or subgoaling) strategy during their continuing design work. In the first instance this strategy was seen to lead to the division of the initial, abstract solution model into a collection of distinct subproblems relating to independent "modules" of the desired artifact. Once identified, such subproblems tended to be focussed on by subjects in an essentially sequential manner, with solutions being developed ***depth-first*** to completion through levels of increasing design detail. Further problem-definition and subproblem generation, however, was seen to occur if solution models that had been generated were themselves multi-faceted and complex and still at high- or intermediate levels of design abstraction. The

observation that a top-down, depth-first design approach is employed in electronics design supports evidence deriving from studies of software engineers which also implicates the use of a depth-first design method by more novice designers (see particularly Jeffries et al. 1981). It is valuable, then, to have generalised this finding to the electronics domain.

(3) Another, rather less predictable finding of the present study was that subjects' design behaviour appeared to reflect the operation of what might be termed a ***satisficing*** principle (see Simon, 1981). That is to say, subjects were commonly seen to focus selectively on the pursuit of single satisfactory high-level solution concepts to a problem, and were rarely seen to generate and compare alternative concepts with the aim of optimising choices. This high degree of selectivity in the generation and pursuit of solutions is interesting in that it is clearly at odds with the principles of good design practice espoused by many devotees of formal design theory (e.g. Hubka, 1982; Pahl & Beitz, 1985).

(4) An important role for mental modelling in electronics design was also implicated in the present study. In particular the undergraduate designers could be conceived of as constructing, manipulating and evaluating internal "representations" (i.e. models) of their design problems and the solutions to these problems. An important idea that emerged with regard to mental modelling in design situations was that people construct not only technical models of their problems and solutions in terms of domain-oriented information but also ***social models*** that reflect factors arising in or relevant to the broader social context in which the design work is being undertaken. In

the present study, social and technical models appeared to interact to a high degree in that design constraints of a social origin would often place strict limitations on what could be attained at a technical level.

(5) One further outcome of the present study was the formulation of a general theoretical model of the global processes that control and coordinate a designer's mental movement between problems and subproblems and his or her development and evaluation of solution concepts (see sections 3.5.8 and 3.5.9). The model was formulated as a set of highly abstract production rules which, taken together, described a general-purpose ***schema*** for electronics design which neatly captured the common overriding deign strategy exhibited by undergraduate designers. One crucial aspect of the undergraduate's approach to design that this schema catered for was the problem reduction method, though many other aspects were also addressed such as (1) processes which lead to the understanding and definition of ill-defined design problems and subproblems (2) processes that enable the selection of subproblems as the current focus of attention and (3) processes that allow for the search, retrieval, evaluation and integration of solution concepts.

4.6.2. ANTICIPATING SOME CRITICISMS OF THE STUDY

Before concluding this chapter it would seem prudent to anticipate some of the criticisms that could be voiced against the study and to attempt a reasoned response to them. Whilst it is clearly possible to critique the investigation at many levels, the present section addresses three major features of the work that seem to be most open to negative appraisal.

A first criticism, then, relates to the sample of engineers studied. It could be argued that the focus of the investigation on relatively inexperienced, "semi-expert" undergraduate designers, rather than experienced experts, provides a rather limited view of the nature of cognitive processes in engineering design. This criticism is clearly reasonable since it is certain that undergraduates, with their restricted experience of real-world design, are unlikely to produce designs that are as commercially viable or as innovative as professional engineers. Does this latter point mean, however, that the strategies of a semi-expert designer are qualitatively different to those of the expert, or is it rather that a semi-expert's knowledge of technical facts, concepts and solutions is more limited to that of the expert? Evidence from the psychological literature on software design suggests that there are in fact differences in both the procedural ***and*** the conceptual knowledge of experts and semi-experts (see, for example, Jeffries et al. 1981; Anderson, Farrel & Sauers, 1984). Existing studies of software designers (see section 2.3.2) generally indicate that experts tend to produce program designs in a definite top-down, breadth-first manner and that they also have an elaborately organised set of factual domain knowledge containing many known design solutions. Less expert programmers, on the other hand, appear to produce designs in a more depth-first way, and - not surprisingly - appear to possess rather less background knowledge than experts (which also seems to be less well organised). Whether such expert-novice differences between software designers also exist within the engineering domain, remains, of course, an empirical question. Abundant informal and anecdotal evidence, however, suggests that expert engineers - like expert programmers - show more breadth-first styles of working than inexperts. In general, then,

it seems that as long as the limited aspects of the present results with respect to the issue of ***expert*** design processes is borne in mind, then the present investigation of semi-expert (i.e. undergraduate) designers has been a valuable research endeavour in its own right. Further work on the psychology of engineering design could, however, be profitably directed toward the pursuit of a similar type of longitudinal study to the present one, but with a focus on engineering design projects as undertaken by more expert subjects.

A second criticism regarding the present study revolves around the issue of retrospective reporting in the investigation of cognitive processes (see section 3.1.2 for an assessment of retrospective report methods in cognitive research). It appeared that the primary method of knowledge elicitation that was adopted for the present study (i.e. the cognitive diary technique) sometimes failed to prevent subjects providing heavily retrospective reports of their design activities. Such retrospective verbalisations may clearly have fallen foul to the distorting affects of forgetting and rationalisation, perhaps enabling only a rather sketchy - and possibly inaccurate - picture of design processes to be formulated. In regard to this second criticism, however, it is unclear how cognitive processes that are extended over many months of a design project could be studied in any practically viable way other than the method that was devised for the present study. Clearly the basic message that arises from this criticism, then, is that there is a clear need (a) to treat the theoretical conclusions that derive from the present study in a tentative manner and (b) to pursue other psychological studies of design using more powerful psychological methodologies such as the use of concurrent "think-aloud" techniques described in section 3.1.2. Having

voiced the need to treat the present results in a tentative manner, however, it is worth emphasising that a particularly encouraging aspect of the findings is the distinct commonality that was observed in the design styles of the individuals studied, who were designers of varying ability pursuing projects in a range of technological subdomains of engineering.

Related to the above criticism is also the issue of the legitimacy of inferring that information was playing a significant role in the subjects problem solving processes even though it was not explicitly mentioned. That is to say, in the DBGs and subsequent discussions certain design/implementation goals and knowledge acquisition goals were inferred to have been held by subjects and to have directed the focus of their activities. As has been discussed earlier, however, such goals were only inferred when the context of the subject's ongoing work (understood in terms of the comments made of current "Activities"). When a bout of activity was observed which was directed toward a clear - though unstated - objective it therefore seemed legitimate to assume that that subject had set (not necessarily consciously) a goal that had resulted in the observed activity.

As a final and general reply to these criticisms it is felt essential to stress again the pioneering nature of the investigation which has attempted to penetrate into a fairly unchartered area of psychological interest where the cognitive processes are complex and the methods of studying these processes must of necessity be rather unconventional. It seemed clear from the outset of the study, then, that the task of understanding the cognitive processes involved in engineering design would be a difficult one, yet it seems true to say many that useful

findings and theoretical conclusions have emerged from the verbal data derived in this study. It is believed that a potentially valuable aspect of this investigation was the encoding scheme that was that was devised for categorising design verbalisations and for structuring these in a readable and coherent graphical manner. Clearly it would be worthwhile to assess the robustness of this technique by undertaking an analysis of verbal data derived from other longitudinal studies of designers - though this is something that was not taken up within the present research programme.

Chapter 5

Study Two: Professional Engineers

Chapter 5. Study Two: Professional Engineers

Page

5.1. INTRODUCTION TO THE PRESENT STUDY

5.1.1. Rationale and aims of the study..... 195

5.2. METHOD

5.2.1. Subjects..... 197

5.2.2. Design..... 198

5.2.3. Task..... 198

5.3.4. Procedure..... 199

5.3. TRANSCRIPTION AND CODING OF DESIGN PROTOCOLS

5.3.1. Transcription scheme for design protocols..... 200

5.3.2. Derivation of taxonomic scheme for coding protocols..... 201

5.3.3. Explanation of categories in taxonomic scheme..... 203

5.3.4. Application of the taxonomic scheme to protocols..... 221

5.4. DISCUSSION OF DESIGN PROTOCOLS

5.4.1. Introduction to discussion..... 223

5.4.2. Case analysis of JO's design protocol..... 225

5.4.3. Quantitative analyses of design protocols..... 239

5.5. GENERAL DISCUSSION

5.5.1. Outline of general discussion..... 253

5.5.2. Design problem solving and the problem reduction strategy..... 254

5.5.3. Subproblem generation from basic models..... 257

5.5.4. Subproblem generation from solution models..... 260

5.5.5. Solution model development at different levels of design abstraction..... 260

5.5.6. Modelling in electronics design: some further considerations..... 262

5.5.7. Tendencies toward sub-optimal performance in design..... 267

5.5.8. A generic schema for electronics design?..... 274

5.5.9. Concluding comments..... 288

5.1. INTRODUCTION TO THE PRESENT STUDY

5.1.1. RATIONALE AND AIMS OF THE STUDY

The fact that the behavioural analyses undertaken in the last study allowed for the formulation of a coherent theoretical understanding of the underlying processes involved in engineering design would seem to be without dispute. Where, however, there would appear to be some need for caution is in treating the theoretical conclusions as being generalisable, say, to different populations of designers or to different types of design problem. For example, engineers who are highly experienced at design work might be expected to apply different (and presumably superior) strategies when tackling design problems. Similarly, design tasks having characteristics different to the problems that were tackled in the previous study (such as a a strong mathematical basis) might well require more problem-specific kinds of design approach for the attainment of solutions.

Apart from the issue of the generality of the previous study's conclusions, the investigation also appears open to criticism on methodological grounds, i.e. in relation to the use of cognitive diaries as the primary data collection technique. Whilst every attempt was made to ensure that the subjects kept their weekly diaries concurrent with ongoing episodes of design activity, the verbal data still suggest that subjects were sometimes presenting retrospective accounts of their work. What this means is that subjects' reports may at times have fallen foul to the distortive effects of rationalisation and forgetting (cf. Ericsson & Simon, 1984; Evans, 1989) with the consequence that the model

of design processes that was formulated on the basis of these reports may be of limited accuracy and completeness. This latter criticism of the diary study is certainly a valid one, though it is very difficult to envisage any practically viable way of studying thought processes extended over many months that would truly eradicate the occurrence of retrospective reporting.

Having voiced the need to treat the conclusions of the previous investigation in a tentative manner it remains important to emphasise that the results are particularly encouraging in as far as they reveal a high degree of commonality in the design styles of the individual participants, who were engineers of varying ability pursuing designs within a range of technological subdomains of electronics. With, however, the overriding aim in mind of deriving a more extensive understanding of the cognitive processes involved in engineering design, it was decided that a second investigation could profitably be undertaken which would (a) use a more sophisticated method for studying ongoing though processes and (b) focus on engineers with a greater level of design experience and expertise. More specifically, then, it was decided that this second study should involve the collection of concurrent verbal protocols of individual professional engineers "thinking aloud" as they attempted a small-scale design task in a laboratory setting. Clearly the study would be rather less realistic than the longitudinal investigation described in chapter 4. However, the use of the think aloud technique was felt to afford the major advantage of providing a continuous and concurrent record of a specific act of thought thereby permitting more accurate study of underlying processes than could obtain form the use of diary or interview techniques.

In summary, then, the present study was clearly intended to build upon the conclusions of the previous investigation. In particular it was felt important to determine further the nature of any common overriding schema that might exist for design, as well as to assess the evidence for any individual difference in methods of working. In this latter respect it was felt that data obtained from professional engineers might reveal some interesting strategic differences in design style as have, for example, been implicated within the domain of software design (see, for example, Jeffries et al. 1981). Additionally, any tendency toward suboptimal design performance was generally anticipated as something that would show up more clearly in concurrent verbal protocols than in diary protocols because of both (a) the greater temporal density of the behavioural observations and (b) the more "direct" nature of the measure as an index of the locus of subjects' attention (see section 5.3.2).

5.2. METHOD

5.2.1. SUBJECTS

Six male electronic engineers served as subjects for this study. One of these subjects was a research assistant within Plymouth Polytechnic while the other five were professional designers from a locally based research and development company (Plessey Semiconductors, Roborough). All of these engineers had been selected for participation in this study on the basis of prior interviews which indicted that they had several years experience at working on design problems at fairly abstract conceptual levels (e.g. at the so called systems level). Engineers who

were rejected after the interview session were those whose design experience had primarily involved work at highly concrete conceptual levels within the electronics domain (i.e. levels where transistors, resistors and other concrete components tend to be the dominant objects that are considered).

5.2.2. DESIGN

No specific experimental manipulations were introduced since it was intended that the investigation should take the form of a set of case studies exploring the nature of expert strategies in engineering design.

5.2.3. TASK

The problem that was used in this study was intended to simulate a real-world design task and was therefore couched in terms of an informal "design specification" (see appendix B). This specification expressed a requirement for the engineer to design an integrated circuit in an image processing application. Certain of the circuits functional requirements were detailed as mathematical formulae while other functional requirements necessitated that the engineer first spend time in developing appropriate mathematical formulae. Clearly then, the problem was devised so as to address the nature of the design strategies employed in dealing with problems at highly abstract levels of the design hierarchy.

The design specification was also formulated in such a way that it presented a fairly well-defined set of requirements relating to the functionality of the chip. There was therefore no great need for the

engineers taking part in the study to flesh out the specification with a multiplicity of functional requirement of their own derivation, though there was certainly considerable room for subjects to attain the required functionality of the circuit in a number of alternative ways. Certain requirements of the chip apart from its functionality, however, were left implicit within the specification and were in need of definition by the subject in order for a worthwhile solution to be reached. For example, the problem specification did not overtly mention the fact that the chip should be capable of operating in real time, though this requirement was implied by the statement that the chip was required to "perform a computer vision task".

5.2.4. PROCEDURE

All subjects were studied individually in a quiet laboratory setting. Subjects was given spoken instructions indicating that they should "think aloud" throughout the design session. The importance of adhering to this requirement was emphasised and subjects were told that they would be prompted to continue verbalising if they dried up for any length of time. Subjects were requested to spend between one and two hours tackling the design task and were asked to carry out their pen and paper work on numbered blank sheets that were provided by the investigator.

A video camera set-up was used to make (a) visual recordings of each engineer's design workings and (b) audio recordings of their think aloud (TA) verbalisations. Back-up recording were also made of the TA reports using a portable sound recorder. Lapel microphones were used as pick-up devices as these enabled a very high quality of sound recording to be

attained - something that would obviously prove important when the transcription of verbal material was undertaken.

The protocol method for studying behaviour is frequently criticised informally as being an unnatural requirement to impose on the subject. Byrne (1983), however, has suggested that one criterion for deciding whether it is feasible to adopt this methodology in a psychological study is to determine whether the subject finds it just as easy to work at a task whilst verbalising as compared to remaining silent. In this regard, then, it is worth noting that the present engineers not only seemed happy enough working with the experimental requirement to think aloud but in a few cases even admitted that they tended to mutter a lot whilst designing anyway!

5.3. TRANSCRIPTION AND CODING OF DESIGN PROTOCOLS

5.3.1. TRANSCRIPTION SCHEME FOR DESIGN PROTOCOLS

The raw behavioural data obtained in the present study comprised (1) a continuous audio recording of each subject's TA verbalisation (2) a continuous video recording of each subject's pen and paper manipulations and (3) original copies of the actual sketches and notes that each subject produced. It should be noted that prior to any transcription or encoding of these raw data, back-ups were made of all video recordings. During this back-up procedure a time code (depicting hours, minutes, seconds and frames) was overlaid into the corner of each video frame so as to permit a subsequent time-based analysis of the material.

The audio recordings of TA verbalisations were transcribed verbatim into a typed format. All pause information was incorporated into the TA protocols during this transcription process by employing the coding scheme depicted in table 5.1 below. It was decided that the subjects' TA protocols would be used as the main starting point for various qualitative analyses whilst the visual material would be used as a supplementary source of data.

PAUSE DURATION	ENCODING
<= 2 seconds	..
> 2 seconds and < 4 seconds	...
>= 4 seconds and < 8 seconds	....
>= 8 seconds	....

Table 5.1: Scheme for encoding pauses in subjects' think-aloud reports

5.3.2. DERIVATION OF TAXONOMIC SCHEME FOR CODING PROTOCOLS

A comprehensive coding scheme for categorising designers' verbalisations and external manipulations was devised from a detailed preliminary analysis of protocol content. It is important to note, however, that the derivation of this coding scheme was not a theoretically neutral endeavour as it was inevitably influenced by a variety of interrelated factors.

One such factor was the investigator's own inclination to view design from the perspective of contemporary research on cognition. This perspective resulted in a tendency to incorporate theoretical notions

into the coding scheme such as "mental modelling" which stem from recent research on human problem solving and thinking. A second and related factor that coloured the formulation of the coding scheme was the investigator's existing understanding of the kinds of activities that were likely to arise in design situations. For example, the previous study of undergraduate engineers had indicated that solutions concepts tend to be externalised onto paper as models taking a variety of forms (i.e. diagrams, mathematical expressions, written notes and graphs). Clearly, then, it was felt to be desirable to include categories within the coding scheme that reflected such a diverse range of external modelling activity.

The primary factor influencing the derivation of the protocol coding system, however, was certainly the actual "theory" of verbal reporting that was adopted as an overarching framework for interpreting the data set. The most penetrating exposition of such a theory appears to be that of Ericsson and Simon (1980, 1984) which was discussed in detail in section 3.1.2. Ericsson and Simon's basic proposal is that concurrent verbal reports provide details of the information heeded by the subject in that they reflect the current contents of short-term memory. From this viewpoint, then, it is essential to treat verbal reports as reflecting the "products" of cognitive processes as opposed to a subject's description of such underlying processes. What this means is that verbalisations come to be treated as similar to any other kind of behavioural data (e.g. latency measures or error rates) which likewise require analysis and interpretation by the investigator. The point is, then, that it is only possible to *infer* cognitive processes from concurrent verbal reports (and other kinds of extended behavioural

traces) on the basis of a detailed and systematic analysis of the actual content of the behaviour as well as its more global structural characteristics (e.g. the order in which verbal units were produced).

Within the present study, the final coding scheme that was developed for classifying designers' behaviour is shown in figure 5.1 and can be seen to be in the form of a two-tiered taxonomy of design activities. Any activity depicted on the right-hand side of this hierarchical categorisation system may be viewed as a more abstract characterisation of the subset of activities depicted on the left-hand side of the taxonomy.

5.3.3. EXPLANATION OF CATEGORIES IN TAXONOMIC SCHEME

The present section aims to provide a detailed description of the categories that were incorporated into the protocol coding scheme. Each generic category (coded A through to M in figure 5.1) will be discussed in turn and a description provided of both (1) the characteristic type of design behaviour implied by the higher-level label itself, as well as (2) the more detailed types of behaviour implied by the lower-level labels. Before progressing to a description of this encrypting scheme, however, it initially seems worthwhile briefly considering the theoretical concept of a "mental model", which, as has already been noted above, played a vital role in its derivation.

From a preliminary analysis of protocols, then, it seemed clear that the present professional engineers - like the undergraduates in the previous study - could usefully be conceived of as constructing, evaluating and manipulating ***mental models*** of their design problems and solutions. It

Figure 5.1: Hierarchical taxonomy for encoding design activities

A1 SELECTING SUBPROBLEM(S) A2 RE-SELECTING SUBPROBLEM(S)	A SELECTING SUBPROBLEM(S)
B1 UNDERSTANDING FUNCTIONAL REQUIREMENTS B2 UNDERSTANDING INPUTS	B DEVELOPING BASIC MODEL OF PROBLEM OR SUBPROBLEM
C1 GENERATING MATHEMATICAL SOLUTION CONCEPT(S) C2 INTEGRATING MATHEMATICAL SOLUTION CONCEPTS	C DEVELOPING MATHEMATICAL SOLUTION MODEL FOR PROBLEM OR SUBPROBLEM
D1 GENERATING ABSTRACT HARDWARE SOLUTION CONCEPT(S) D2 INTEGRATING ABSTRACT HARDWARE SOLUTION CONCEPTS	D DEVELOPING ABSTRACT HARDWARE SOLUTION MODEL FOR PROBLEM OR SUBPROBLEM
E1 SPECIFYING PERFORMANCE CONSTRAINT(S) E2 SPECIFYING RESOURCE USAGE CONSTRAINT(S) E3 SPECIFYING DESIGN PROCESS CONSTRAINT(S)	E SPECIFYING DESIGN CONSTRAINT(S)
F1 EVALUATING MATHEMATICAL SOLUTION CONCEPT(S) F2 PROVING MATHEMATICAL SOLUTION CONCEPT(S) F3 APPLYING KNOWLEDGE OF HARDWARE POSSIBILITIES/CONSTRAINTS TO MATHEMATICAL SOLUTION CONCEPT(S)	F EVALUATING MATHEMATICAL SOLUTION MODEL
G1 EVALUATING ABSTRACT HARDWARE SOLUTION CONCEPT(S) G2 APPLYING KNOWLEDGE OF DETAILED HARDWARE POSSIBILITIES/CONSTRAINTS TO ABSTRACT HARDWARE SOLUTION CONCEPT(S)	G EVALUATING ABSTRACT HARDW SOLUTION MODEL
H1 EVALUATING PROGRESS H2 SUMMARISING PROGRESS	H MONITORING PROGRESS
J1 WRITING MATHEMATICAL EXPRESSION(S) J2 SKETCHING FUNCTIONAL BLOCK DIAGRAM(S) J3 SKETCHING GRAPHICAL MODEL(S) J4 MAKING NOTES	J EXTERNALISING MODELS ONTO PAPER
K1 READING PROBLEM SPECIFICATION K2 ATTEMPTING TO RECALL K3 QUESTIONING SELF K4 QUESTIONING INVESTIGATOR K5 READING HELP SHEETS K6 TIME-FILLING VERBALISATION	K SEARCHING FOR INFORMATION
L1 COMMENTING TO INVESTIGATOR L2 EXPLAINING TO INVESTIGATOR	L COMMUNICATING TO INVESTIGATOR
M1 STATING PLAN M2 STATING INTENTION M3 RE-STATING INTENTION M1 STATING INTENTION TO DEFER WORK	M STATING PLANS AND INTENTIONS

is important to stress that in the immediate context of the present study the term mental model is being used to refer to a ***mental representation*** of some aspect of the design problem or its solution which is constructed by the engineer during an attempt at the set task. This use of the term is, therefore, very much in line with the "mental models" notion of Johnson-Laird (e.g. 1983) who uses it to refer to a mental representation which is built up and exists as a "structural analogue" of a real-world situation. It is important that the reader adheres to this conceptualisation of the term since a multiplicity of rather different uses of the mental models notion - often incorporating the idea of mental simulation and prediction of events - abound in contemporary psychology (see, for example, Kahneman & Tversky, 1982; de Kleer & Brown, 1983). Mental modeling as espoused by these latter authors is clearly relevant to theories of design problem solving as was explicated in the context of the previous study (see section 4.5.5). For example, mental simulation was implicated as being used by student engineers for the purpose of determining the behavioural characteristics of their technical design solutions. The notion of mental modelling in this simulation sense is, however, not being ascribed to in the present behavioural taxonomy, though some consideration of its possible value in theories of design will be dealt with later in this chapter (see the general discussion section).

CATEGORY A: SELECTING DESIGN SUBPROBLEM(S)

During their design work subjects were sometimes heard to make comments that indicated the explicit selection of one or more design subproblems to work on. As in the undergraduate study, the term "subproblem" was again used in the present analyses to refer to the existence of a

functional requirement or a set of functional requirements which needed to be met by a design solution. Any verbal data, then, that revealed the selection of subproblems were viewed as falling within category A and were treated as being specific instances of either SELECTING SUBPROBLEM(S) (category A1) or RE-SELECTING SUBPROBLEM(S) (category A2). The latter activity occurred whenever a subject explicitly re-selected a subproblem that had previously been selected but had not actually been pursued. A few examples of verbal units falling under category A are as follows:

> "The first problem to solve is the N coordinate pairs along the line vector .. which is problem (A) ..." (SELECTING SUBPROBLEM. DS's protocol. Start time 00.19.52)

> "Right. So we're going to try and find the ends of the normal vectors ..." (RE-SELECTING SUBPROBLEM. IH's protocol. Start time 00.33.36)

CATEGORY B: DEVELOPING BASIC MODEL OF PROBLEM OR SUBPROBLEM

Subjects produced many verbalisations during their design work which indicated an activity broadly categorisable as DEVELOPING BASIC MODEL OF PROBLEM OR SUBPROBLEM (category B). Subjects were viewed to be engaging in this activity whenever they worked on a problem that was poorly understood or ill-defined (i.e. inadequately represented mentally) and attempted to convert it into a more clearly understood and well-structured state.

The notion of a basic model was introduced in section 4.5.2 and appears to be a useful concept for describing the structured, personalised representation that an engineer develops of (1) given problem or subproblem information (e.g. relating to functional requirements and input parameters) as well as (2) any inferred information concerning the

problem (e.g. relating to constraints on the device's functionality). It is assumed here (cf. sections 4.5.2 & 4.5.8) that the basic model provides the starting point for the development of design solutions that take the engineer into ***deeper*** levels of the design hierarchy and that without a well-developed basic model the engineer will have great difficulty proceeding with a design solution.

The initial analysis of the present protocols indicated that the development of a basic model of either the original design problem or any subsequent subproblem could involve two types of lower-level activities which were termed UNDERSTANDING FUNCTIONAL REQUIREMENTS (category B1) and UNDERSTANDING INPUTS (category B2). These two activities quite obviously related to the comprehension/representation of respectively (a) the functional requirements expressed by a problem or subproblem and (b) the input parameters expressed by a problem or subproblem. Some examples of category B behaviours are as follows:

> "OK. We're going to trace along a line vector - whatever such a thing is - but it's between two points - arbitrary points X and Y - which I'll assume are defined as complete pixels .. And we're going to trace along between those two points, a number .. of coordinate pairs .." (UNDERSTANDING FUNCTIONAL REQUIREMENTS. JF's protocol. Start time 00.03.36)

> "Yeah. Right well I'm going to assume in that case that ... that funny squiggle, whatever it is - a lambda or l - is that distance there .. The stepping factor - I don't see it anywhere else .... I'll assume it's that distance. Now then ... I can generate .. a vector that's a normal to that vector .. mmnh, the vector product thing .. So .. vector product .... (UNDERSTANDING INPUTS, UNDERSTANDING FUNCTIONAL REQUIREMENTS. JC's protocol. Start time 00.39.49)

Before concluding this discussion of category B behaviours it is worth noting a few points concerning the actual externalisation of basic models onto paper (see also category J below). The subjects in the

present study were generally seen to produce external representations of such models in one of three ways. Firstly, basic models were sometimes externalised as textual descriptions which itemised and separated out any functional requirements expressed within a selected problem or subproblem. Often subjects referred to these functional requirements idiosyncratically as being either "operations", "tasks" or "problems". Secondly, basic models were sometimes externalised as a combined block diagram plus textual description of functional requirements expressed within a selected problem or subproblem. As usual in the latter type of representation, labelled blocks would be used to depict separate functional requirements and links between blocks would be used to denote functional connectivity. Thirdly, basic models were sometimes externalised as abstract graph representations of the functional requirements expressed within a selected problem or subproblem. Often such a graph would be in a form similar to that of the diagram depicted in the actual problem statement (refer to appendix B).

CATEGORIES C AND D: DEVELOPING MATHEMATICAL SOLUTION MODEL FOR SUBPROBLEM and DEVELOPING ABSTRACT HARDWARE SOLUTION MODEL FOR SUBPROBLEM

Turning now to consider activities relating to the development of technical design solutions to subproblems, the initial analysis of subjects' comments indicated that they were developing solution models at two conceptually distinct levels of design detail, i.e. (1) at a mathematical level and (2) at an abstract hardware level. Instances of any behaviour directed toward the attainment of the former type of a mathematical design solution were designated by the term DEVELOPING MATHEMATICAL SOLUTION MODEL FOR SUBPROBLEM (category C) whilst instances

of behaviour directed toward the attainment of the latter were categorised by the term DEVELOPING ABSTRACT HARDWARE SOLUTION MODEL FOR SUBPROBLEM (category D). Similar to basic models, technical solution models may usefully be viewed as representations which are developed and altered over time.

In developing solution models, then, the present engineers were always seen (a) to work from a well-defined basic model of the current subproblem and (b) to expand the functionality expressed by the subproblem into a more detailed level of the design hierarchy. It should be noted, however, that in this design study, solution models were never seen to be developed that went to levels of design detail any lower than an abstract hardware level. In other electronic design situations, though, categories of solution model applicable to more detailed design levels than this are clearly conceivable (e.g. solution models relating to the design of, say, gate arrays).

In the case of engineers developing <u>mathematical</u> solution models for subproblems, the occurrence of two distinct types of behaviour was revealed by subjects' comments and these behaviours have been respectively termed GENERATING MATHEMATICAL SOLUTION CONCEPT(S) (category C1) and INTEGRATING MATHEMATICAL SOLUTION CONCEPTS (category C2). The former behaviour related to the generation of one or more mathematically oriented solution ideas or concepts for attaining the desired functional requirement (or set of functional requirements) expressed by the subproblem. In the present study, for example, Pythagoras theorem formed the basis of a solution model for many subjects in regard to the subproblem of how to calculate the length of a line vector. Behaviours categorisable as integrating mathematical

solution concept(s) occurred when subjects combined newly generated mathematical concepts with existing mathematical concepts to form a consistent and integrated solution model. An example of a verbal unit encodable as category C behaviour is shown below:

> "No. Do Pythagoras .... So .. Uh .... We call this t1 and t2 .... t1 is equal to .. uhm .. (end x minus .. sx), and that's the (unintelligible word) of that - if the line's going the other way .... So .. and t2 is this, equal to (ey minus .. sy) ... and then .. the desired length L1 is obviously equal to the square root of (t1 squared plus t2 squared) .... So .... And then each point ... divide by, that by N ...." (GENERATING MATHEMATICAL SOLUTION CONCEPTS, INTEGRATING MATHEMATICAL SOLUTION CONCEPTS. JM's protocol. Start time 00.36.11)

In the case of engineers developing <u>abstract hardware</u> solution models for a subproblem, again two generic types of behaviour were seen to occur which were termed GENERATING ABSTRACT HARDWARE SOLUTION CONCEPT(S) (category D1) and INTEGRATING ABSTRACT HARDWARE SOLUTION CONCEPTS (category D2). The former behaviour related to the subject producing one or more abstract hardware concepts (e.g. "registers" and "adders") relevant to the attainment of the required functionality expressed by the subproblem. Behaviours that were categorisable under category D2, i.e. integrating abstract hardware solution concept(s), occurred when subjects combined abstract hardware solution concepts in order to form a consistent and integrated solution at this particular level of design detail. An example of some verbalisation encodable as category D behaviour is as follows:

> "OK .. We have all this .. All right .. Stack for .. uh .... x ... Y pairs ... Well for the nesting .. nesting, 'cos you've got two levels .. and then you've got .. an adder .. for x and for y .. So it's like .. two adders .. +++ You've got the increment .. increment .. register .." (GENERATING ABSTRACT HARDWARE SOLUTION CONCEPTS, INTEGRATING ABSTRACT HARDWARE SOLUTION CONCEPTS. DS's protocol. Start time 00.44.39)

Mathematical solution models were externalised by subjects as either textual descriptions of mathematical ideas or, more commonly, as mathematical expressions (e.g. formulae and equations). Abstract hardware solution models, on the other hand, tended to be externalised as block diagrams - where labelled blocks and connections represented abstract hardware concepts. At this abstract hardware level externalised solution concepts were often labelled explicitly by subjects as structures (e.g. "registers") which possessed implicit functionality (e.g. "data storage").

Clearly, many issues arise from the previous characterisation of model development behaviour outlined in relation to both basic models and solution models. Some questions that might be asked, for example, concern (a) the mechanisms that underly the storage and retrieval of models (b) the ways in which the dynamic characteristics of models are accommodated and (c) the manner in which alternative solution models to a subproblem are handled. These kinds of issues clearly warrant further consideration. Because of the clear theoretical level of explanation that any answers must possess, however, any attempt at dealing with such issues will be left to the general discussion section of this report where a detailed theoretical model of professional design processes will be introduced and explained.

CATEGORY E: SPECIFYING DESIGN CONSTRAINTS

In the present study the information presented within the problem statement (i.e. the problem "givens") related to the functional requirements and input characteristics of the to-be-designed artifact. The problem specification therefore contained no explicit information

about design constraints other than functionally oriented ones. For example no mention was made of the desired speed of the chip's operation or of its required power consumption characteristics. When, however, subjects were pursuing the development of either basic models or solution models they periodically stated assumptions concerning these kinds of design constraints. Any verbalisations of this latter kind were encoded under the behaviour category SPECIFYING DESIGN CONSTRAINT(S) (category E).

The engineers attempting the present design exercise were in fact heard to specify three distinct types of design constraint (other than functional requirements) and these were termed performance constraints, resource usage constraints and design process constraints. Performance constraints refer to requirements that a designed artifact must satisfy when it is functioning (e.g. speed of operation, heat generation and the like). Resource usage constraints refer to requirements that a designed artifact must satisfy in terms of resources that are used in its manufacture or its operation (e.g. the area the device must occupy or the power that it consumes). Design process constraints relate to limitations on the actual process of design itself such as the time available to produce the design. The three lower-level behaviours that related to the specification of these three different types of constraint were termed SPECIFYING PERFORMANCE CONSTRAINT(S) (category E1), SPECIFYING RESOURCE USAGE CONSTRAINT(S) (category E2) and SPECIFYING DESIGN PROCESS CONSTRAINT(S) (category E3) and examples of such behaviours are presented below:

> "Well all right .. it's a vision task .. So you've got to do it quite fast .." (SPECIFYING PERFORMANCE CONSTRAINT. DS's protocol. Start time 00.33.09)

"Goodness only knows how you would want to work that out! .... And goodness only knows how you would figure it out in advance .. So assuming that I haven't got time to figure out that." (SPECIFYING DESIGN PROCESS CONSTRAINTS. JF's protocol. Start time 00.08.39)

"I've now put down a lot of resources .. You'd have to check if you can use them in parallel or not .." (SPECIFYING RESOURCE USAGE CONSTRAINTS. JO's protocol. Start time 01.34.42)

Since the specification of design constraints is an important aspect of developing a well-defined representation (i.e. a basic model) of an ill-defined problem or subproblem, category E behaviours could really be subsumed under category C. This was not done, however, since it was felt to be important to maintain the richness of the taxonomic scheme and not collapse too many categories together.

CATEGORIES F AND G: EVALUATING MATHEMATICAL SOLUTION MODEL FOR SUBPROBLEM and EVALUATING ABSTRACT HARDWARE SOLUTION MODEL FOR SUBPROBLEM

The comments made by the present engineers revealed that solution models were being evaluated at both (1) mathematical design levels and (2) abstract hardware design levels. Two high-level categories were therefore devised to encode activity relating to the evaluation of solution models and were designated by the terms EVALUATING MATHEMATICAL SOLUTION MODEL FOR SUBPROBLEM (category F) and EVALUATING ABSTRACT HARDWARE SOLUTION MODEL FOR SUBPROBLEM (category G).

Considering initially the behaviour termed EVALUATING MATHEMATICAL SOLUTION MODEL FOR PROBLEM OR SUBPROBLEM (category F), the preliminary analysis of protocol data indicated that three specific forms of this activity were occurring. The first of these has been termed EVALUATING MATHEMATICAL SOLUTION CONCEPT(S) (category F1) and related to the evaluation (often just a positive or negative judgement) of either a

single mathematical solution concept that had been generated or to an integrated set of mathematical solution concepts (see example verbalisation below).

> "That's right. So that's the first formula .. OK. That should be pretty easy to do .... (EVALUATING MATHEMATICAL SOLUTION CONCEPTS. JO's protocol. Start time 00.35.51)

The second lower-level behaviour of PROVING MATHEMATICAL SOLUTION CONCEPT(S) (category F2) involved the subject *formally* proving the adequacy of the solution model that was being developed. The subject was generally seen to accomplish this kind of systematic proof by simulating the functionality of the solution model. An example verbalisation that illustrates this behaviour of proving a mathematical solution model is presented below:

> "Let's just quickly try .. another 0,0 point, but say a 3,2 .... and put .. N equals 5 again .. 2, 3, 4, 5 ... Uh, the first X step is X2 .. which is (3 - 0) over 4 .. equals 3 over 4 .. X step .. Y step is .. (2 - 0) over 4, which is a half .. So the first step is 0,0 then it's a half, then it's 3 over 4 ... 1.5 .. 2.25 .. 3 .. 1, 2, 3, 4, 5." (PROVING MATHEMATICAL SOLUTION CONCEPTS. JO's protocol. Start time 00.35.03)

The third lower-level behaviour of APPLYING KNOWLEDGE OF HARDWARE POSSIBILITIES/CONSTRAINTS TO MATHEMATICAL SOLUTION CONCEPT(S) (category F3) involved the engineer evaluating a partially or fully developed mathematical solution model using technical knowledge of possibilities and constraints at lower levels of the design hierarchy, i.e. at hardware levels. It should be noted that the term "constraint" has been used here to refer to any *attribute* (e.g. functional, aesthetic or whatever) of a detailed hardware concept and not to a constraint in terms of a "constraint requirement" that a solution concept must meet (see foregoing category E). An example of some verbal data which were

considered to relate to the activity of applying knowledge of hardware in the evaluation of a mathematical solution concept is presented below:

> "Can't really use the hypotenuse +++ Or could do actually Could use a square root table .. Ah square roots expensive! .... But I could use a table. Yes .. Could .. however use a table in ROM +++ Yeah .. Let's see. 512 by 512 .. Yeah .. Sum of the squares. Oh dear! Could be a bit expensive in memory. 512 by 512 ... Yes that's too much .. No can't do that .. Too much memory required" (APPLYING KNOWLEDGE OF HARDWARE POSSIBILITIES/CONSTRAINTS TO MATHEMATICAL SOLUTION CONCEPT. DS's protocol. Start time 00.08.40)

Turning now to consider the high-level activity termed EVALUATING ABSTRACT HARDWARE SOLUTION MODEL FOR PROBLEM OR SUBPROBLEM (category G), it was seen that two types of such evaluation behaviour were occurring in the present design situation. The first type was termed EVALUATING ABSTRACT HARDWARE SOLUTION CONCEPT(S) (category G1) and was applied to produce a straightforward positive or negative assessment of either a single abstract hardware solution concept that had been generated or to an integrated set of abstract hardware solution concepts. The second type of process, termed APPLYING KNOWLEDGE OF DETAILED HARDWARE POSSIBILITIES/CONSTRAINTS TO ABSTRACT HARDWARE SOLUTION CONCEPT(S) (category G2), involved the designer using his technical knowledge of detailed hardware possibilities and constraints to make an assessment of the viability of a partially or fully developed abstract hardware solution model. Examples of verbalisations reflecting these two kinds of evaluation activity are presented below:

> "Right, in order to get .. the system to run fast enough to be .. well as fast as I can make it .. I can get a Sine/Cos Generator that outputs some new data on .. on every clock cycle for the system. I can get a video line store that outputs a new pixel on every cycle of the system .. Therefore I can run the accumulator flat out .. Therefore I need a Coordinate System Generator that works on every cycle .. (APPLYING KNOWLEDGE OF DETAILED HARDWARE POSSIBILITIES TO ABSTRACT HARDWARE SOLUTION CONCEPTS. JF's protocol. Start time 00.38.38)

CATEGORY H: MONITORING PROGRESS

The term MONITORING PROGRESS (category H) was devised to encode verbalisations which indicated the occurrence of any activity relating to the self-monitoring of progress. Perhaps the most useful way to conceptualise this behaviour is as corresponding to a "self-reflective" mode of thought where the engineer is effectively taking a step back from his current activity to provide a high-level assessment or overview of his progress. Frequently, then, verbalisations of this kind involved the engineer using a first person pronoun in order to express success, failure or frustration with respect to his design progress.

Two specific types of progress-monitoring behaviour were considered to be occurring in the present design situations and these were designated by the terms EVALUATING PROGRESS (category H1) and SUMMARISING PROGRESS (category H2). The occurrence of the former behaviour was indicated by comments expressing an evaluation of progress that had been made. Such comments were generally distinct from those expressing an evaluation of specific technical models (covered by categories F and G), though it must be admitted that the investigator sometimes experienced difficulty in attaining a consistent choice between categories H and categories F or G. The occurrence of the behaviour termed SUMMARISING PROGRESS was indicated by the subject providing verbal summaries of what had been achieved and these comments often related to technical models that had been developed. Some examples of comments reflected the behaviour of monitoring progress are presented below:

> "Getting bogged down with the geometry here. But, however .... We're possibly going to run over the time" (EVALUATING PROGRESS. JO's protocol. Start time 01.07.02)

> "Uhm .. I'm getting a bit .. a bit bogged down with this! - as to

what I'm actually ending up with and whether that's the right way to go about calculating that .. vector .. Uhm .." (EVALUATING PROGRESS. JC's protocol. Start time 21.37.00)

"So I've now got my .. N ... coordinate pairs .. along .. line segment" (SUMMARISING PROGRESS. JO's protocol. Start time 00.36.24)

"Right so we have Register File .. which has got each of the coordinate pairs inside it on the main line" (SUMMARISING PROGRESS. IH's protocol. Start time 01.17.02)

CATEGORY J: EXTERNALISING MODELS ONTO PAPER

In tackling problems and subproblems, subjects were observed to produce on paper a multiplicity of sketches, notes and jottings. Any ongoing generation of such visible products of the design process was considered to be a behaviour broadly categorisable as EXTERNALISING MODELS ONTO PAPER (category J). This activity, then, was considered to occur whenever a subject converted any mental model that he was currently developing into a permanent external form.

Basic models and solution models were seen to be externalised by subjects in a variety of different forms including graphs, text, block diagrams as well as various permutations of these. Four specific categories were therefore devised to encode any activities reflecting a process of externalising models onto paper. The first of these categories, termed WRITING MATHEMATICAL EXPRESSION(S) (category J1), referred to a process that tended to predominate during the development and evaluation of mathematical solution models for subproblems. Mathematical expressions that were seen to be written by the present subjects ranged from abstract formulae (e.g. $y = mx + c$) depicting mathematical relationships through equations containing one or more specific values to single mathematical symbols and numbers.

A second category termed SKETCHING FUNCTIONAL BLOCK DIAGRAM(S) (category J2) was devised to encode the activity of externalising a model on paper as an interconnected set of functional blocks. As has already been noted previously, the block diagram representation was seen to be used by subjects to depict basic models as well as abstract hardware solution models. In the former case labelled blocks denoted separate functional requirements expressed within a problem or subproblem. In the latter case functional blocks denoted separate solution concepts - usually functional submodules - generated to meet the desired functional requirements expressed within a problem or subproblem.

The third category termed SKETCHING GRAPHICAL MODEL(S) (category J3) was used to encode the activity of externalising an abstract graphical model onto paper. This type of model - which resembled the diagram shown within the problem statement - was frequently produced by subjects engaged either in the process of developing a basic model or in the processes of developing and evaluating mathematical solution models. Generally, graphical models tended also to contain mathematical information (e.g. concerning coordinates, angles, distances and the like) and they were sometimes annotated with text.

A final category, MAKING NOTES (category J4), was used in the present context in a restricted sense to refer to the activity of writing on paper some verbal text that either was distinct from or integrated with any diagrammatic representation. Thus, for example, the activity of externalising an annotated block diagram would be encoded as reflecting a process of making notes as well as a process of sketching a functional block diagram.

CATEGORY K: SEARCHING FOR INFORMATION

An important role for behaviours relating to knowledge acquisition and information search in design was implicated by the data obtained in the undergraduate study and such behaviours were also strongly evident in this present study of professional designers. Any verbalisations that subjects produced which appeared to reflect activities relating to the search for information (whether internally or externally represented) were categorised with the high-level label SEARCHING FOR INFORMATION (category K).

Six lower-level categories were defined in order to encode different kinds of information-search behaviour, and those which most commonly occurred were READING PROBLEM SPECIFICATION (category K1), QUESTIONING SELF (category K3) and QUESTIONING INVESTIGATOR (category K4). ATTEMPTING TO RECALL (category K2) was sometimes evident if a subject was striving to recall information from mind but was finding this difficult whilst READING HELP SHEETS (category K5) was - perhaps surprisingly - a behaviour that was almost never observed to occur. Behaviour that was categorised as being TIME-FILLING VERBALISATION (category K6) also occurred very rarely in the present study. This latter finding arguably serves to indicate the relative ease with which the subjects coped with the think-aloud requirement that was imposed upon them. Some examples of verbalisations that subjects produced which related to information-search activity are provided below:

"I have no idea how fast this thing has to be .. (unintelligible here) perform the following computer vision task ... How fast would you want a computer vision task to go to? .. How fast? .... How fast shall we make it go? ... (QUESTIONING SELF. JF's protocol. Start time 00.14.32)

"He says design an integrated circuit - What sort of length does

he mean - does he want us to go to? <Well not right down the hierarchy to gate level but maybe a few block> a few blocks of .. (QUESTIONING INVESTIGATOR. IH's protocol. Start time 01.08.02)

"X, Y .. Find a point ... Trying to remember some geometry .... Uhm ..." (ATTEMPTING TO RECALL. JO's protocol. Start time 00.38.02)

CATEGORY L: COMMUNICATING TO INVESTIGATOR

Many verbalisations that subjects produced during the design session appeared to be made for the purpose of communicating with the investigator either socially or to convey explanations for design decisions. Such verbalisations have therefore been categorised as reflecting a verbalisation process termed COMMUNICATING TO INVESTIGATOR (category L). The two detailed categories which were devised to encode these verbalisations were termed COMMENTING TO INVESTIGATOR (category L1) and EXPLAINING TO INVESTIGATOR (category L2). The former category covered any socially oriented comments including, for example, observations, opinions and jokes about the experimental set-up, the problem or the nature of design. The latter category covered comments aimed at explaining anything - though usually design decisions - to the investigator. Such verbalisations tended to have a highly introspective feel to them and usually involved the engineer expressing the rationale behind a design decision or providing a description of his own preferred design strategies and techniques.

CATEGORY M: STATING PLANS AND INTENTIONS

During their design sessions subjects often expressed their intended actions with regard to the pursual of a selected problem or subproblem. Comments such as "I'm going to sort of go in and actually again specify the inputs, outputs and internal processing" (see JO's protocol) were

treated as reflecting a straightforward verbalisation behaviour that was termed VERBALISING PLANS AND INTENTIONS (category M).

Four lower-level instances of such verbalisations were revealed in a preliminary analysis of protocols. Firstly, then, the term STATING PLAN (category M1) was devised to encode comments expressing a set of intended actions relating to the pursual of a selected problem or subproblem. The terms STATING INTENTION (category M1) and RE-STATING INTENTION (category M1) were devised to encode comments expressing, respectively, a single intended action or an intended action that had previously been expressed but which had not immediately been undertaken. Finally the term STATING INTENTION TO DEFER WORK (category M3) was instigated to encode comments expressing the intention to defer further work on a subproblem. An example of an intentional statement is provided below:

> "I'm going to sort of go into .. uhm - now I think I've understood the problem - just go in and actually again specify the inputs, outputs and internal processing. Just at a very high level." (STATING INTENTION. JO's protocol. Start time 00.10.58)

5.3.4. APPLICATION OF THE TAXONOMIC SCHEME TO PROTOCOLS

The more detailed categories in the coding scheme (see figure 5.1) were used in its application to the think-aloud protocols derived in the present study. For each protocol, verbalisations were encrypted in a chronological ordering and were taxonomised only after a thorough analysis of both their information content and their immediate context. The process of encoding protocols, then, involved the investigator determining which category within the encoding scheme most nearly

captured the ongoing design behaviour reflected in the content of the verbalisation.

The application of the coding scheme resulted in each protocol being segmented into a sequence of verbal "units" of varying time duration, each of which was encoded as either a single design behaviour or an inextricably interwoven grouping of design behaviours. After protocols had been encoded, the start and end times of each verbal unit were added into the protocol. The six fully encrypted protocols (see appendix B) provided the basis for all subsequent data analyses.

It is briefly worth repeating that whilst the taxonomic system was devised primarily with the aim of encoding verbal behaviour it was often possible to relate a subject's comments to other overt activities going on at the same time (e.g. writing, drawing, shuffling around of paper and the like) such that the scheme was not just restricted to the encoding of verbalisations.

It is also important to note here that a reliability check was carried out in order to assess the author's consistency in encoding protocol data using the taxonomic scheme. A single protocol was chosen at random form the full set of six and recoded approximately one year after the original data encoding had been undertaken. Prior to carrying out the recoding of the selected protocol, all codes that had originally been applied to each verbal segment were deleted (by someone other than the author) and replaced with a number that indicated how many activity codes classified that particular verbal unit. Out of the 201 activities that were subsequently recoded within the complete protocol, only 18 (i.e. 9%) were misclassified (note that encoding confusions seemed not

to exhibit any obvious patterns). This high degree of agreement between first and second encoding supports to the view that the taxonomic scheme could be applied to design protocols with a high degree of internal consistency by the deviser of the scheme - and presumably by anyone else who was trained in its use.

5.4. DISCUSSION OF DESIGN PROTOCOLS

5.4.1. INTRODUCTION TO DISCUSSION

Section 5.3, with its emphasis on the encoding scheme that was devised for taxonomising design behaviour, has presented a rather static and simplistic rendition of the distinct ***types*** of activity that were evident in subjects' protocols. What this means, then, is that very little coverage has actually been given to the more dynamic characteristics of design behaviour in terms of its broader ***organisational*** and ***structural*** features. Interesting issues that arise in this regard concern, for example, (1) the way in which different activities predominate during particular phases of a subject's design session (2) the manner in which different design behaviours are seen to co-occur or inter-relate, and (3) the general pattern of behavioural transitions that arise during design problem solving. As a case in point, one of the few organisational features of the engineers' activity that was noted previously in regard to these issues was the common tendency for subjects initially to formulate abstract design solutions to subproblems before then proceeding to generate more concrete design solutions. Clearly, the nature of any patterns and sequences occurring in ongoing design behaviour is crucially relevant to

one of the main themes of the present research programme, i.e. the possible existence of a general schema for electronics design common to all engineers in its essential respects.

In the next two sections, then, coverage will be provided of the follow-on analyses that were undertaken of encoded protocols in order to derive an improved understanding of the dynamic dimensions of design problem solving. Section 5.4.2 presents an extensive ***qualitative*** analysis of a single encoded protocol (for subject JO) which was randomly chosen from a set of five (JC's protocol having been barred from possible selection owing to this subject's obvious difficulty in making any real headway with a solution to the problem). The detailed case discussion of JO's protocol not only allows for organisational aspects of design activity to be considered but also permits examples of error and inconsistency to be pinpointed and their consequences traced. Within the discussion of this protocol, continual reference is made to the detailed semantic content of JO's design work and frequently excerpts of protocol data are actually included. Such direct use of this material, however, is thought to be no bad thing for it serves both to (a) give the reader a feel for the actual conceptual knowledge that subjects applied in tackling the present electronic design problem as well as (b) convey something of the reality of design problem solving in all its richness and flexibility (cf. the presentation of the Design Behaviour Graphs in chapter 4).

Following on from the discussion of JO's design work, section 5.4.3 presents some ***quantitative*** analyses that were undertaken on all six encoded protocols. These quantitative analyses, which involved computation of the distribution and time course of design activities,

were undertaken primarily so as to provide a clearer indication of the way in which subjects' design behaviour changed over the duration of a design session. As will be seen, these latter analyses indeed proved very useful for comparing and contrasting different subject's design strategies and enabled some conclusions to be drawn regarding the nature of individual differences in design style.

It is worth stressing that the following two sections do not aim to provide any strong theoretical interpretation of the nature of the cognitive processes underlying design behaviour. Such a theoretical analysis of findings is something that is felt to be most profitably left to the general discussion section of this chapter which will strive to draw out and synthesise a model of design processes that is consistent with both the present data set as well as that derived in the previous undergraduate study.

5.4.2. CASE ANALYSIS OF JO'S DESIGN PROTOCOL

To facilitate the following discussion of JO's design activity the protocol was segmented into eight equal sections based on a division of the total design time. For convenience these protocol sections are henceforth referred to as "Design Periods" and are numbered in a chronological ordering from one through to eight.

DESIGN PERIOD 1 FOR JO: (00.00.00 to 00.12.38)

JO's activities during Design Period 1 were oriented toward his gaining a complete, accurate and structured basic model of the problem specification. He was observed systematically to read and re-read the problem statement, each time extracting information relating to the

chip's functional requirements - which he sometimes termed "operations" - as well as its input and output parameters.

In outline, JO began the design session by stating his intention to read through the problem statement ("first of all I'm just going to read through the .."). Subsequent to this initial familiarisation read through the problem, JO commented that he had "got the general idea" and he explained that he was next intending "to go into some more specifics" and "draw out the problem". On the next pass through the specification JO actually restructured the given information into a personalised set of notes and sketches which detailed the inputs to the chip and which itemised its functional requirements. These notes (see sheet 1 of JO's design work in appendix B.2.1) together with the verbalisation suggest that JO was conceptualising the overriding problem as involving a set of high-level subproblems relating to the design of separate functional modules of the chip. This suggestion is lent further weight by the fact that JO actually referred to the functional requirements of the chip as numbered subproblems. An illustration of this is the following comment made by JO: "So problem one: Generate coordinate pairs .... Generate the appropriate coordinates for a line vector drawn at a normal to the given line vector". The three basic subproblems that JO defined during this phase of design work actually mapped onto the three numbered sets of functional requirements detailed in the problem specification.

After this second phase of developing a basic model of the problem statement JO stated his intention to cycle through the problem yet one more time ("I'm going to sort of go into .. uhm - now I think I've understood the problem - just go in and actually again specify the inputs, outputs and internal processing. Just at a very high level").

The end of Design Period 1 saw JO beginning this third and final pass through the problem statement.

DESIGN PERIOD 2 FOR JO: (00.12.38 to 00.25.16)

At the beginning of Design Period 2 JO was pursuing his final derivation of a basic model of the problem specification. Once again it was clear that JO was continuing to model the overriding problem as a set of well-defined, conceptually distinct but interacting subproblems relating to abstract functional modules - or what he termed "tasks" or "operations" - of the chip. This time, however, the three original subproblems that he had defined were subdivided further into subproblems relating to component "tasks" or "operations". For example subproblem two that JO had identified during his second pass through the problem specification was seen to be divided into a set of interdependent component subproblems ("So that's divided it down into three tasks there +++ M pairs, separated by 1, on the N lines").

This more detailed subdivision of functional requirements is depicted in the notes and sketches that JO produced during his final pass through the problem specification (see sheet 2 for JO in appendix B.2.1). This external model of the problem information is clearly a more elaborate and structured version of the model that JO had developed earlier (sheet 1) and was the final basic model of the overriding design problem that JO produced on paper. Interestingly, in the latter stages of deriving this basic model, JO noted an inconsistency in the given problem specification itself ("Mr C----- has given me M equals 5 points as an example, but he wants to sum.. summate over i equals zero to M which is actually 6 points. But not to worry I'll carry on. I think M should

possibly have been 4 there +++"). The fact that JO picked up this inconsistency - although not a particularly consequential one - reflects the advantages of the thorough approach that JO was adopting to represent the problem information. Generally JO was both very thorough and highly structured in his approach to developing a basic model of the given design problem. One particular comment that he made during this Design Period suggests that this is his customary style of tackling new design problems ("I normally spend quite some time checking through these things").

Mid-way through Design Period 2 JO stated his intention to derive solutions for the subproblems or tasks that he had identified during the derivation of a basic model of the design problem ("I'm going to have to try and find out how I'm actually going to do this now - these tasks"). He proceeded to generate a basic plan for tackling the set of subproblems which involved "Delving into each problem in turn. Looking at it probably from a .. trigonometric point of view". JO next selected the initial subproblem to work on ("The first problem to solve is the N coordinate pairs along the line vector .. which is problem (A)" and decided that it was necessary to pursue the component subproblem of how to calculate the actual length of the line vector ("what I need to know .... is the length of the line vector +++ What calculations need to be performed to do that?"). Interestingly, JO was heard to question "Is there an easy way of doing this?" which suggests that he was looking for a simple mathematical solution that would both be "easy" for himself to derive and "easy" for the chip to implement.

JO began work on this subproblem by sketching a graphical model which included hypothetical coordinate information (see sheet 3 in appendix

B.2.1). On the basis of this graphical representation of the subproblem JO generated a mathematical solution concept involving Pythagoras theorem for generating the length of the line vector ("+++ actual length is, uh, root X squared plus Y squared ... X squared equals 4 plus 9 .... Uhm .. Which is root 13"). JO briefly checked his workings for this example and concluded that it was sound ("Ah that's it! .. That's root X squared plus Y squared for that"). It seems, however that JO was somewhat unhappy with this mathematical solution model for the subproblem although he decided that he would stick with it for the time being ("I've got to think of some .. easy formula for calculating the length of the line segment .. I'm sure there must be an easier way. But, however"). He proceeded to state his plan to run through a few example sets of hypothetical data in order to prove the validity of this mathematical solution model ("I'll just try it for two examples to .. uh .. prove by induction") and having done this he concluded by saying "That's right! It's the right formula - has to be!"). At the end of Design Period 2 JO returned to the subproblem of actually generating the N coordinate pairs along the line vector ("Next thing we've got to have is N points along that line").

DESIGN PERIOD 3 FOR JO: (00.25.16 to 00.37.54)

The start of Design Period 3 saw JO sketching a graphical model - incorporating hypothetical coordinate information - for the subproblem of generating the N coordinates along the line vector (see lower half of sheet 3 in appendix B.2.1). Although JO initially appeared to be aiming for a trigonometric solution concept ("Got to take the angle into account +++ Can't remember any bloody trigonometry") he soon decided that this was unnecessary and generated a simple mathematical expression

for calculating the stepping distance between coordinate points along the line vector ("Aha! +++ Knowing the length +++ and the number of points - which is 5 +++ we can divide the length ++ over (N - 1)".

The latter mathematical expression, though rather imprecise, clearly paved the way for JO to develop a complete mathematical solution model to the subproblem of generating N points along the line vector. Working from this expression, then, JO proceeded - via further graphical sketching - to come up with a more detailed equation set (see large circled section on sheet 4 in appendix B.2.1) for calculating stepping distances in the x and y directions along the line vector ("So I have a formula .... Coming across to a formula to find these N coordinate points +++ The X .. step is (X2 - X1) over (N - 1) .. Y step .. equals (Y2 - Y1) over (N - 1)"). Notably this set of equations for calculating stepping factors rendered redundant the need to calculate the length of the line vector - a subproblem that JO had previously spent some considerable time working on during Design Period 2 and for which he had developed a mathematical solution concept involving Pythagoras theorem.

JO proceeded to prove these mathematical concepts using hypothetical coordinate data and he then moved on to develop the concepts into a final iterative formula (see sheet 4 in appendix B.2.1) for calculating the actual N coordinates along the line vector ("The first point is X1, Y1 .. 2nd, 3rd, 4th, 5th obviously has to be X2, Y2 .. That's X1 plus ... (X2 - X1) over (N - 1) ... That's the X .. and the Y1 plus (Y2 - Y1) over (N - 1) .. and that's actually an iterative formula"). JO was next heard to evaluate this mathematical solution model for the "N coordinate pair" subproblem by applying his knowledge of potential hardware possibilities ("that I know will come down to some sort of accumulate

function .. That's thinking about it in hardware .... That's quite easy in fact"). He seemed, however, to be unsure of the validity of the mathematical expressions for he again stated his intention to prove it using more sample data ("I'll try it through a simple example). This proof undertaken, JO uttered a positive evaluation of the mathematical solution ("That's right. So that's the first formula .. OK. That should be pretty easy to do").

The final part of Design Period 3 saw JO selecting a new high-level subproblem to turn his attention toward (i.e. how to generate "M coordinate pairs separated by 1 ... aligned at the normal"). He began to sketch a graphical model of the subproblem but halted this activity in order to question the investigator regarding the study requirements ("Are you more interested in the hardware at the end or in how someone solves these higher level problems?). JO's comments attendant to this question are interesting since they suggest that his preferred strategy for tackling design problems involves a top-down, breadth-first approach ("++ I mean I could sort of on a few hunches blast into drawing out block diagrams .. but I would normally prefer to solve all of the .. you know, the mathematical problems at the top level first").

DESIGN PERIOD 4 FOR JO: (00.37.54 TO 00.50.32)

At the start of Design Period 4 JO was continuing to work on the subproblem of generating normals to the original line vector. He was clearly attempting to solve this subproblem using a geometric solution concept ("Trying to remember some geometry") and subsequently generated a solution idea that related to the "slope" of the normal ("OK .... I can work on slopes - but, uh. Yes I need +++ the slope of the line"). JO

stated his intention to prove this concept ("Just try an example") and proceeded to elaborate - with coordinate information - the graphical sketch that had been started in Design Period 3 (see sheet 5 in appendix B.2.1) and to externalise mathematical formulae.

Directly after these activities JO was not heard to make any explicit evaluation of the mathematical solution model that he was developing but rather stated his intention to try yet another proof ("Let's just .... try this one"). Again then, a graphical model was constructed and a proof worked through (see lower half of sheet 5 in appendix B.2.1). This time JO evaluated his current progress rather negatively ("Getting a bit stuck on this one here") and moved on to try yet another proof of the solution concepts that had been generated and developed. This new proof of the current mathematical solution model, however, brought JO some success ("That's the slope of the line .. That's minus the slope .. So .. slope (unintelligible here) ... Y equals X ..and Y equals -X .... minus 1 .. Y equals 1, X equals 1, Y equals -1 +++ That's what we want!"). JO then stated his intention to derive a formula to express more clearly his mathematical solution model ("Let's ... get a bit more ... Derive a formula for this") and proceeded in his attempt to generate this formula during the latter part of Design Period 4. He appeared, however, to have little success in this endeavour for he ended the Design Period still without any precise expression for calculating normal coordinates (see sheet 6 in appendix B.2.1).

Generally, throughout Design Period 4 JO appeared to be having difficulty in making any significant headway with the development of a mathematical solution model for the subproblem of generating normals to the line vector. The model that he was developing was based - it appears

- on the idea of rotating the slope of the original line vector to produce the normals to the vector. It should be noted that JO's verbalisations throughout this Design Period were technically dense in terms of their mathematical content. Additionally JO frequently appeared to be confused as he developed his mathematical solution ideas - suggesting perhaps that working memory limitations were having some affect as he struggled with a difficult mathematical problem. These factors clearly led to difficulties with the interpretation and encoding of protocol material and it would therefore seem prudent to treat the foregoing analysis of JO's activities for Design Period 4 in a somewhat tentative manner.

DESIGN PERIOD 5 AND DESIGN PERIOD 6 FOR JO: (00.50.32 TO 01.03.10) AND (01.03.10 TO 01.15.48)

Design Period 5 and Design Period 6 are dealt with in conjunction here since an analysis of protocol material revealed that JO was undertaking a great deal of repetitive and relatively uninteresting design work. For this latter reasons it is also felt to be wholly unnecessary to provide any lengthy and detailed discussion of the activities arising in these Design Periods and only a basic overview of JO's work has therefore been attempted. The caveats noted above concerning the analysis of activities for Design Period 4 are also relevant to the discussion of the present Design Periods; again the content of JO's verbal reporting was both heavily mathematical and often seemingly confused as he continued to strive toward a solution for the subproblem of generating normals to the original line vector.

The main focus of JO's behaviour during Design Periods 5 and 6 related

to his attempt to prove the mathematical solution concept of rotating slopes to generate normals. JO was clearly persisting with this mathematical solution concept despite mounting difficulties, and the externalised models that he produced during this Design Period clearly depict the extent of his fixation with this solution idea (see bottom of sheet 6 and top of sheet 7). Indeed he seemed to realise himself that the attainment of a proof was taking a considerable amount of time ("It's a pretty long proof this. But, however") and during these Design Periods various comments betrayed JO's mounting frustration with the solution approach he had adopted ("OK ... I've screwed this up a bit +++ There's a simpler formula for these bloody points on a line but I can't remember! +++ Getting bogged down with the geometry here").

Just over halfway through Design Period 6 JO was heard to mention the possibility of abandoning his chosen solution path ("OK. I could approach this in a very crude ... crude way instead of trying to do something clever") and indeed JO seemed at this point to take a step back from the details of his current solution attempt in order to look for another solution path. He soon commented that he felt he might have been working with an incorrect assumption ("Ah! I might have made a .. an assumption here +++ Up to now I think I've been sort of going around in a circle a little bit +++ Yeah. I've been assuming to date that the l was actually .. a vector step. But I presume - as it's not stated there - that it's done in both the X and Y direction .. I was trying to derive a formula that would give me the sine and cosine of l to add onto it"). The end of Design Period 6, then, saw JO working to attain a new mathematical solution to solve the subproblem of generating coordinates along normals to the main line vector. As it transpired, this new

solution was not only readily attained (see bottom of sheet 8 in appendix B.2.1) but was also seemingly rather more elegant than the solution originally attempted.

DESIGN PERIOD 7 AND DESIGN PERIOD 8 FOR JO: (01.15.48 TO 01.28.26) AND (01.28.26 TO 01.41.08)

At the start of this Design Period JO selected his third major high-level subproblem ("We want to fetch the nearest pixel .. from memory .. and summate it's value according to the expressions below generating two data items"). This subproblem encompasses the functional requirement for the integrated circuit to fetch pixel values from memory and summate them according to two given mathematical formulae. As JO began his work on this subproblem he first ensured that he had a clear grasp of the given requirements for this section of the chip and only then proceeded to generate and integrate - in very rapid and efficient manner - a set of high-level hardware solution concepts (including multipliers, accumulators, a ROM and a look-up table) to meet the functional specification (see sheet 9 in appendix B.2.1).

At this point in Design Period 7 (i.e. having just generated a schematic hardware solution for this third major functional module of the chip) JO was seen to move on to a phase of design work that took all existing solution models - most of which were at a mathematical level - toward a hardware level of design detail. During the second half of this Design Period, then, JO was seen (1) to treat each mathematical solution model that he had developed as a new subproblem, i.e. as a subproblem expressing mathematical operations that needed to be attained in hardware, (2) to generate an integrated abstract hardware solution model

for each such subproblem such that the model could meet the required mathematical functionality and (3) to integrate - as they were generated - all abstract hardware solution models for the various mathematically-based subproblems. The composite solution model that was produced as a result of this highly breadth-first design approach can be seen on sheet 10 in appendix B.2.1 and clearly existed as a high-level, flow-oriented, hardware solution model for the whole integrated circuit design problem that had been set.

The initial part of JO's final design period saw him stating his intention to develop a slightly more detailed hardware characterisation of the complete integrated circuit than the one that had just been attained ("We can get a bit more specific now"). This phase of hardware solution development - like the phase in Design Period 7 - involved JO adopting a highly fluent and integrated breadth-first design approach in which he (1) took sections of the existing, composite hardware solution model that had just been developed (sheet 10) and (2) translated these sections of the model into more elaborate and detailed hardware concepts within a new composite solution model (see sheet 11 in appendix B.2.1).

THE GLOBAL CHARACTERISTICS OF JO'S DESIGN ACTIVITY

The foregoing analysis of JO's design protocol indicates that his behaviour exhibited a considerable degree of structuredness throughout all periods of the design session. One dominant aspect of this structured design approach appeared to be JO's repeated application of a problem reduction strategy in tackling design problems and subproblems. In operation this strategy initially led to the division of the given problem specification into three fairly well-defined high-level

subproblems relating to conceptually distinct functional modules of the desired integrated circuit. Later on this problem reduction strategy again appeared to be applied at all levels of design detail - whether mathematical or abstract hardware - in order that solutions for problems might be factorised into sets of new sub-problems for which more detailed solution models could be generated. Of interest in this regard was the observation that the order in which JO selected subproblems to work on reflected his concern with the derivation of solutions for a full set of subproblems at a single level of design detail before then moving on to tackle subproblems at a new more detailed level. This kind of balanced, breadth-first development of design solutions from mathematical levels through to hardware levels presumably enabled JO to deal more readily with the functional interdependency of subproblems and their solutions.

Whenever JO selected any particular problem to work on that was in an ill-defined state he was always seen to spend time gaining a clearer representation of the problem (i.e. the functionality goals and input parameters that it expressed) by means of reading, understanding and assimilation activities. These activities usually involved some basic inferencing from and elaboration of the problem information and always seemed to occur at the problem's current level of detail. This kind of problem definition and structuring behaviour was most predominant in the case of JO tackling the problem specification itself, though such behaviour did occur periodically throughout the whole of JO's design session.

Whenever JO's current design problem had been successfully represented in a well-defined state, JO was next customarily observed to exhibit a

clear pattern of behaviour involving the repeated generation and integration of solution concepts and the subsequent evaluation and/or proof of the resulting solution model. This generate-integrate-evaluate sequence seemed to occur much more fluently for JO at more detailed design levels, i.e. during the development of hardware solution models. Certainly at the mathematical level of his design work, JO often appeared to be scrabbling around and generating any mathematical concepts that seemed relevant to the task at hand - with the consequence that he spent most of his time working at proving the viability of solution ideas at the mathematical level. On the whole, though, JO's mathematical solution models appeared to end up being complete and accurate enough to meet the functionality expressed by the given design specification.

The reason why JO appeared to be more adept at working at hardware design levels was no doubt because hardware concepts were more central to his range of design expertise whilst mathematical concepts were rather more peripheral. Certainly JO appeared to be working well within the bounds of his expertise during the hardware modelling stages of his design work. Still, however, it is worth noting that irrespective of the design level at which JO was working, he was rarely seen to display any inclination to generate and develop alternative solution models for particular subproblems with the aim of achieving optimal designs. Indeed JO was often often observed to stick fixedly to initial solution ideas even when their viability started to become highly suspect. Unviable solution paths were generally only seen to be abandoned by JO when headway was obviously becoming completely unlikely.

Some further consideration is given to the organisational aspects of

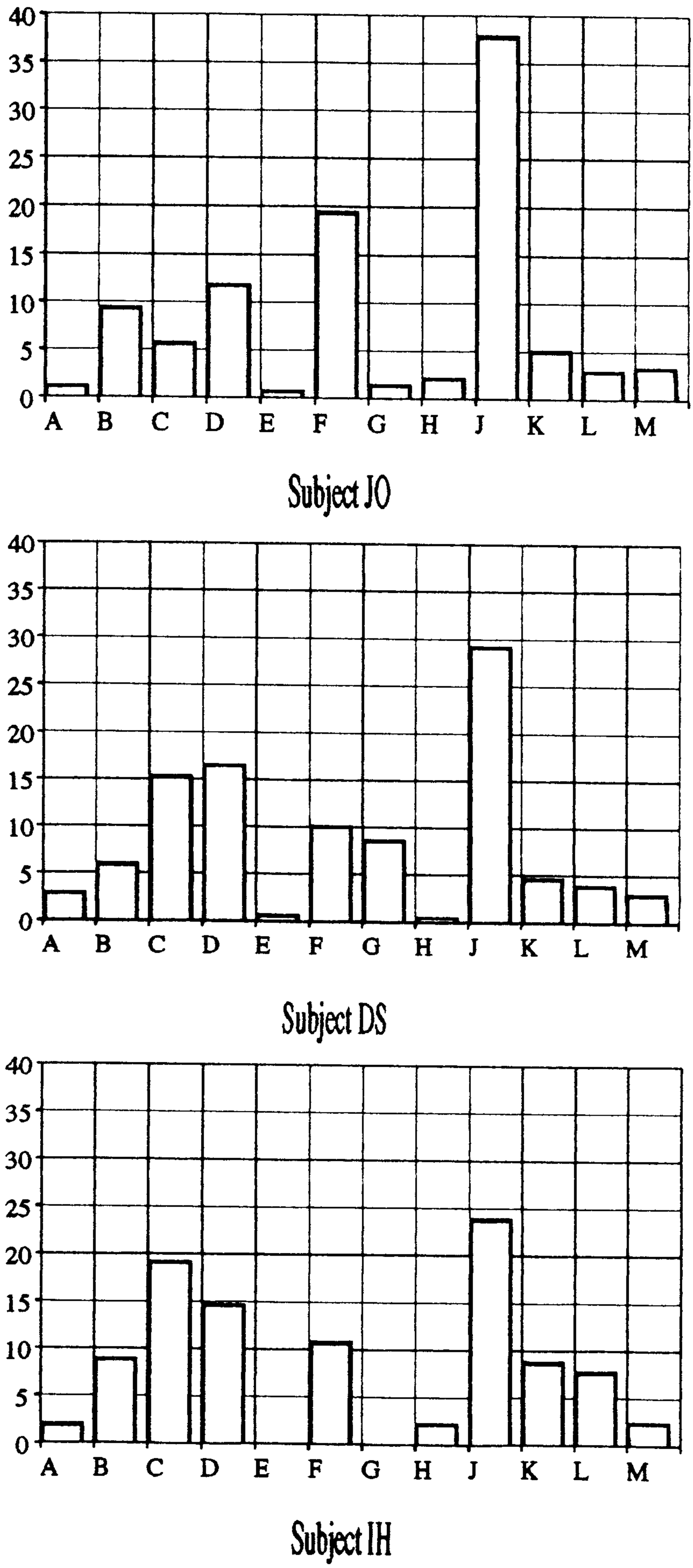
subject's design activity (including that of JO) in the following section which details some quantitative analyses which were undertaken on designer's encoded protocols.

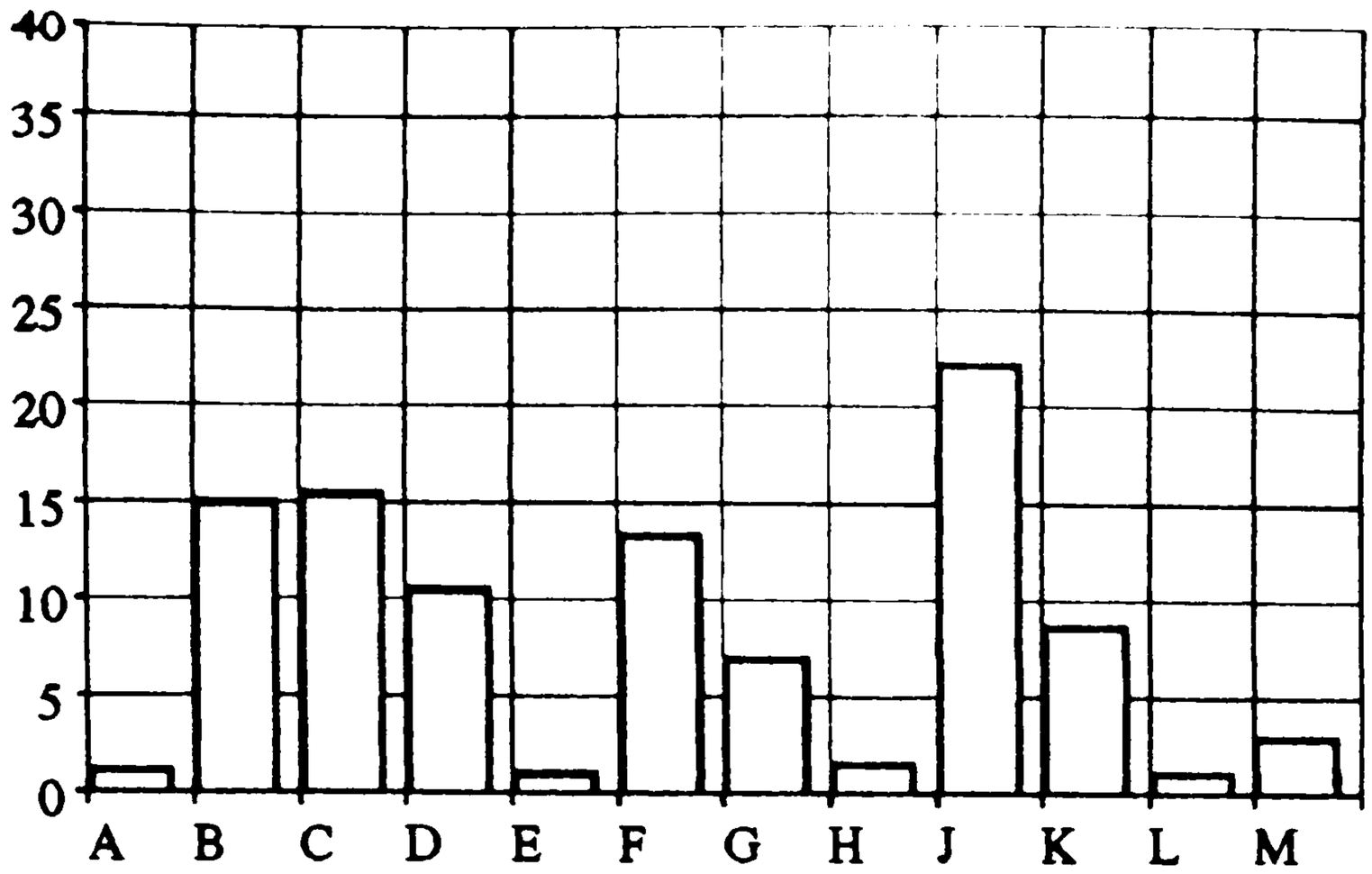
5.4.3. QUANTITATIVE ANALYSES OF DESIGN PROTOCOLS

One type of analysis that initially seemed worthwhile pursuing on subjects' encoded protocols was a straightforward assessment of the time that each engineer spent engaged in each of the twelve different high-level design activities detailed in the behavioural taxonomy (see section 5.3.3). Such an analysis was felt to be useful for (1) pinpointing the types of activities which either predominated or were minimally apparent during subjects design sessions and (2) providing a means for assessing whether there were commonalities across subjects in respect to the distribution of their activities over the design sessions. The results of these analyses were plotted graphically as bar charts (see figure 5.2) which each show what percentage of a subject's total design time was spent engaged in different behaviours (labelled A to M as in the the behavioural taxonomy).

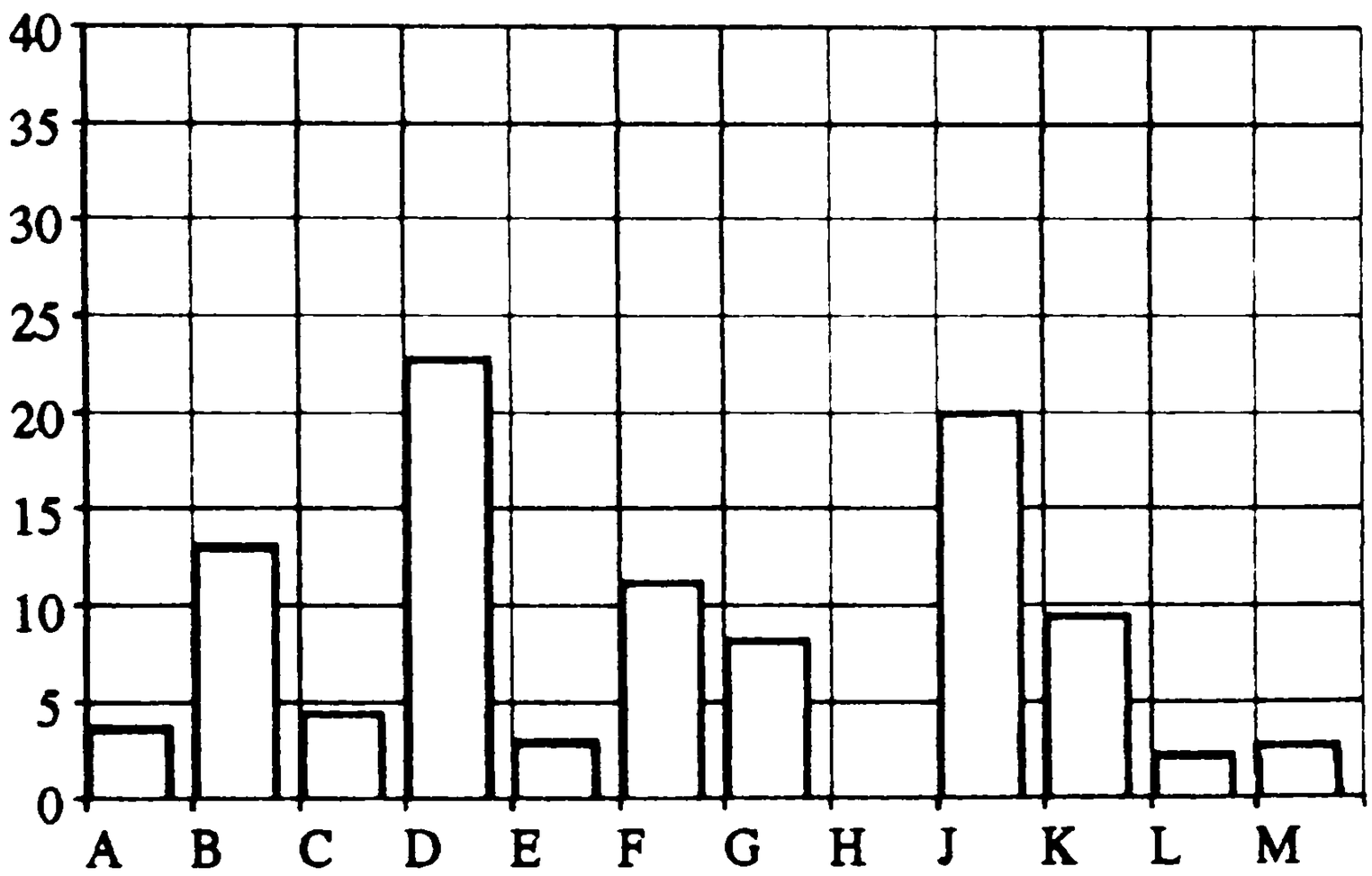
Looking first at the kinds of behaviour that ***predominated*** during subjects' design sessions, it emerged that the duration of time spent externalising models onto paper (activity J) generally occupied a large proportion of subjects' total design time. Once again, then, this result points to the important role played by external modelling in the process of engineering design - a theme that will be pursued in section 5.5.6 where some consideration will be given to the idea that external modelling provides a way for the engineer to overcome inherent processing limitations of working memory.

Figure 5.2: Bar charts depicting the percentage of each subject's total design time engaged in behaviours A to M.

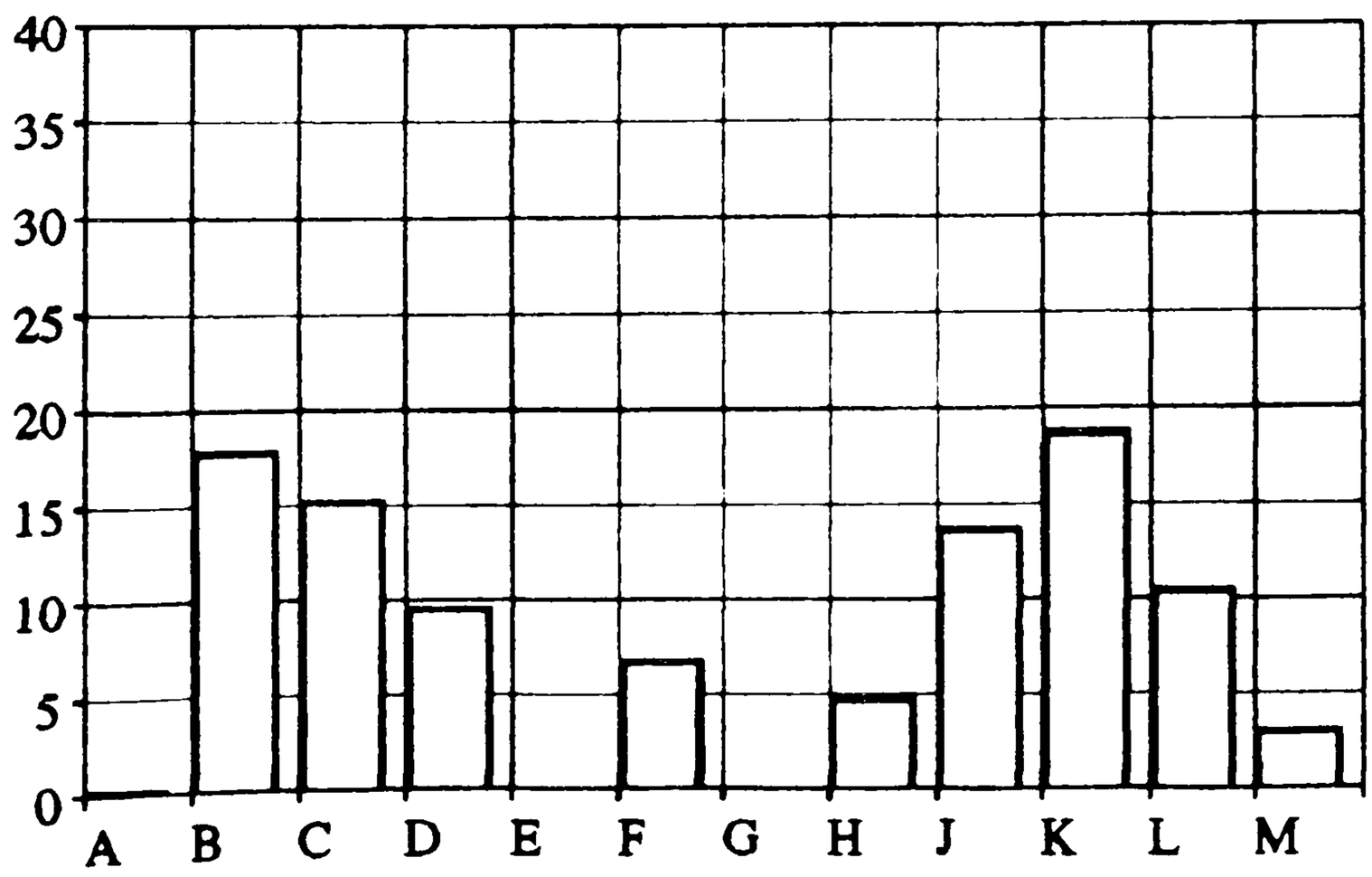




Subject JM



Subject JF



Subject JC

Looking further, then, at behaviours which appeared to dominate subjects' design effort, the bar charts below indicate that for the majority of the engineers the development and evaluation of mathematical solution models (activities C and F respectively) as well as the development (but not the evaluation) of hardware solution models (activity D) were activities of some importance. What these latter results seem to suggest is that, firstly, subjects were spending a large part of their design sessions working at a ***mathematical*** level of design abstraction and that, secondly, although the engineers proceeded to generate hardware solution models for design problems they generally seemed to engage in little - and sometimes no - evaluation of these hardware solutions. These latter findings may be interpreted as reflecting that engineers were getting rather de-motivated toward the latter part of their design sessions and were just glad to generate hardware ideas without bothering to evaluate them. There is, however, little evidence to support this suggestion and indeed the subjects seemed highly motivated throughout their design work. On the other hand, it seems that a more plausible case can be made for the argument that the present engineers were more familiar and more confident with hardware solution ideas than with their mathematical solution ideas and were therefore less inclined to pursue elaborate evaluations and simulations of their hardware models. In this regard it is noteworthy that many of the subjects often seemed to be struggling with their solution ideas at a mathematical level and seemed to iterate the concepts that they generated through a number of improvements before proceeding to hardware levels. One further activity which, not surprisingly, showed up with some prevalence in subjects' protocols was that of developing basic models of problems and subproblems (Category

B). Interestingly there was some indication that those subjects that spent a greater amount of time than others in developing basic models of their problems were the ones who were also having the most difficulty in making progress with their solutions. Subject JC, for example, who was one of the more junior and inexperienced members of the subject pool that were studied (a) spent a considerable amount of time in trying to understand the overriding design specification and (b) was seen to make very little progress with a design solution throughout the design session.

Turning, now, toward a consideration of those behaviours that generally seemed to feature only minimally in subjects design work, there is perhaps little surprise in observing that these included the activities of (1) selecting design problems (2) stating plans and intentions (3) searching for information and (4) communicating to investigator. Whilst these behaviours were, however, only briefly engaged in by subjects this is certainly not to say that they played only a superficial role in subjects design problem solving. Indeed behaviours such as subproblem selection clearly had a crucial impact on the coordinated development of solutions during the ongoing process of design. One final point worth briefly noting concerning these latter types of activities, is that subject JC, who had considerable difficulty in tackling the design problem, was the subject who showed the greatest tendency to spend time searching for information and communicating with the investigator. Clearly, though, there is nothing profoundly important in the observation that subjects who have difficulty in designing will devote considerable time to understanding the problem through whatever means are available.

At this point it is worth moving on to discuss the results of some further analyses of the protocol data that were undertaken. This time, however, the focus was on the way in which the distribution of high-level design activities changed as a function of the time course of the ongoing design session. In order to carry out such an analysis, each subject's protocol was segmented into eight sections based on a division of the total design time. As noted above during the discussion of JO's protocol, the resulting time divisions were referred to as Design Periods - each being numbered chronologically from one through to eight. Each subject's protocol, having been split up into eighths, was then subjected to a computational analysis that calculated the percentage duration of time spent engaged in behaviours A to M for each Design Period (see tables 5.2 to 5.7 below). During this computational procedure it should be noted that whenever a behaviour (or a set of behaviours) straddled Design Periods, then the duration of those behaviours was split equally between Design Periods.

To aid the interpretation of the results of these computational analyses, certain percentage values within the tables below have been circled whilst others have been left uncircled. Glancing across any row will reveal that for any particular design behaviour (1) the circled values indicate those Design Periods where the subject spent relatively more time engaged in the behaviour and (2) the uncircled values indicate those Design Periods where the subject spent relatively less time engaged in the behaviour. Thus, for example, looking at behaviour B (developing basic models) for Subject JO, it can be seen that he engaged in this activity much more in Design Periods 1, 2 and 7 than in other Design Periods. This suggests that JO was rather more concerned about

Behaviour	Design Period 1	2	3	4	5	6	7	8	Total (%) time spent
A	0%	4%	3%	0%	0%	0%	3%	0%	1.08%
B	37%	15%	1%	0%	0%	6%	13%	0%	9.29%
C	0%	3%	14%	14%	0%	13%	0%	0%	5.61%
D	0%	0%	0%	0%	0%	0%	51%	42%	11.78%
E	0%	1%	0%	0%	2%	0%	2%	1%	0.66%
F	0%	8%	33%	47%	34%	33%	0%	0%	19.34%
G	0%	0%	0%	0%	0%	0%	2%	9%	1.37%
H	1%	5%	2%	0%	3%	5%	1%	0%	2.10%
J	26%	35%	34%	32%	55%	36%	41%	41%	37.78%
K	28%	18%	5%	3%	2%	1%	16%	0%	5.00%
L	4%	8%	6%	0%	0%	4%	2%	2%	2.89%
M	4%	4%	2%	4%	4%	1%	2%	4%	3.10%

Table 5.2: Percentage duration of activities in each of eight Design Periods for Subject JO.

Behaviour	Design Period 1	2	3	4	5	6	7	8	Total(%) time spent
A	3%	0%	0%	9%	87%	2%	0%	2%	2.83%
B	31%	8%	0%	0%	8%	0%	0%	0%	5.87%
C	4%	12%	0%	14%	2%	0%	0%	0%	15.22%
D	0%	0%	0%	0%	0%	26%	41%	49%	16.42%
E	2%	0%	0%	0%	0%	2%	0%	1%	0.58%
F	0%	31%	0%	12%	0%	21%	0%	0%	10.01%
G	0%	0%	0%	0%	0%	38%	12%	18%	8.57%
H	8%	0%	0%	0%	1%	0%	2%	0%	0.46%
J	23%	29%	0%	41%	1%	7%	20%	24%	29.12%
K	30%	10%	0%	12%	0%	4%	0%	1%	4.55%
L	0%	22%	0%	3%	1%	0%	0%	5%	3.86%
M	5%	8%	100%	10%	0%	0%	25%	0%	2.52%

Table 5.3: Percentage duration of activities in each of eight Design Periods for Subject DS.

Behaviour	Design Period 1	2	3	4	5	6	7	8	Total(%) time spent
A	1%	0%	2%	3%	0%	2%	22%	5%	1.95%
B	31%	0%	19%	0%	4%	7%	0%	0%	8.81%
C	9%	18%	14%	17%	52%	0%	0%	41%	19.11%
D	0%	0%	0%	0%	0%	40%	42%	0%	14.57%
E	0%	0%	0%	0%	0%	0%	0%	0%	0.00%
F	1%	24%	8%	38%	4%	1%	0%	11%	10.69%
G	0%	0%	0%	0%	0%	0%	0%	0%	0.00%
H	0%	1%	13%	1%	3%	1%	1%	0%	2.15%
J	4%	47%	13%	29%	24%	20%	3%	22%	23.75%
K	20%	5%	20%	4%	7%	17%	21%	5%	8.70%
L	25%	1%	9%	4%	4%	12%	0%	14%	7.70%
M	8%	4%	2%	3%	2%	0%	0%	1%	2.56%

Table 5.4: Percentage duration of activities in each of eight Design Periods for Subject IH.

Behaviour	Design Period 1	2	3	4	5	6	7	8	Total(%) time spent
A	0%	1%	3%	3%	4%	2%	0%	0%	1.16%
B	48%	28%	9%	15%	0%	11%	0%	0%	14.98%
C	0%	19%	0%	4%	12%	23%	33%	0%	15.41%
D	4%	12%	41%	4%	0%	9%	0%	34%	10.46%
E	0%	0%	0%	11%	0%	0%	0%	0%	1.00%
F	0%	0%	0%	2%	78%	8%	43%	11%	13.30%
G	4%	6%	10%	6%	0%	6%	0%	17%	6.91%
H	0%	1%	0%	4%	8%	0%	0%	11%	1.51%
J	17%	31%	26%	23%	0%	26%	15%	27%	22.14%
K	23%	3%	11%	25%	1%	5%	2%	0%	8.64%
L	2%	0%	0%	0%	0%	5%	1%	0%	1.07%
M	3%	1%	0%	2%	5%	6%	6%	0%	3.42%

Table 5.5: Percentage duration of activities in each of eight Design Periods for Subject JM.

Behaviour	Design Period 1	2	3	4	5	6	7	8	Total(%) time spent
A	9%	1%	0%	0%	0%	19%	0%	0%	3.60%
B	45%	50%	6%	0%	7%	0%	0%	0%	12.99%
C	0%	0%	17%	3%	0%	0%	11%	0%	4.37%
D	0%	7%	3%	0%	22%	60%	30%	32%	22.72%
E	0%	14%	6%	0%	0%	3%	0%	0%	2.88%
F	0%	0%	6%	41%	40%	0%	6%	0%	11.15%
G	0%	0%	12%	0%	7%	11%	9%	63%	8.13%
H	8%	0%	0%	0%	0%	0%	0%	0%	0.00%
J	12%	3%	43%	35%	13%	0%	29%	0%	19.92%
K	31%	14%	2%	9%	8%	1%	3%	5%	9.36%
L	1%	10%	0%	2%	0%	0%	0%	0%	2.18%
M	2%	0%	4%	9%	2%	5%	11%	0%	2.70%

Table 5.6: Percentage duration of activities in each of eight Design Periods for Subject JF.

Behaviour	Design Period 1	2	3	4	5	6	7	8	Total(%) time spent
A	0%	0%	0%	0%	0%	2%	0%	0%	0.21%
B	33%	24%	44%	2%	0%	38%	0%	0%	17.73%
C	3%	0%	7%	34%	14%	0%	19%	40%	15.41%
D	0%	0%	5%	0%	61%	3%	0%	0%	9.74%
E	0%	0%	0%	0%	0%	0%	0%	0%	0.00%
F	0%	0%	6%	28%	0%	5%	12%	7%	6.69%
G	0%	0%	0%	0%	0%	0%	0%	0%	0.00%
H	8%	3%	13%	3%	8%	0%	12%	0%	4.74%
J	0%	11%	7%	26%	0%	23%	5%	31%	13.61%
K	28%	13%	14%	4%	17%	29%	37%	13%	18.69%
L	36%	40%	3%	0%	0%	0%	3%	1%	10.38%
M	0%	10%	0%	4%	0%	0%	12%	17%	3.36%

Table 5.7: Percentage duration of activities in each of eight Design Periods for Subject JC.

developing basic models of problems during the initial phases of his design session than in the later phases (cf. section 5.4.2 above). Such a result is clearly understandable since JO was obviously confronted with an overriding design problem that needed a high degree of understanding and structuring before an attempt at a solution could be made. Additionally, a look at the distribution of category B behaviour for the other subjects reveals some close similarities in that such behaviour seems to dominate the initial phases of their design sessions relative to later Design Periods.

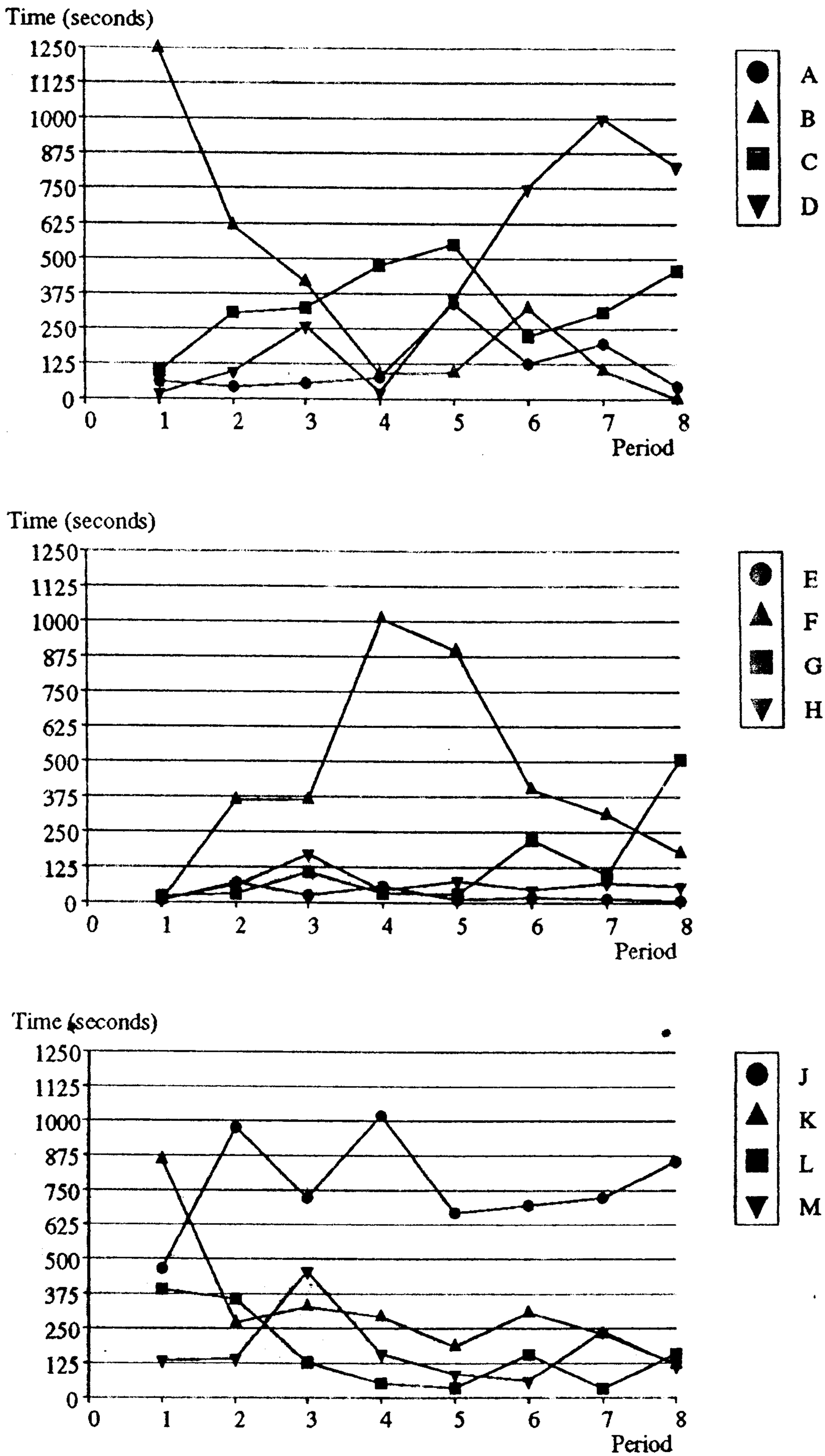
Looking through tables 5.2 to 5.7 in the manner outlined above reveals a large number of commonalities in the way that behaviours change over the time course of subjects' design sessions. Firstly, for most subjects the time spent developing mathematical solution models (activity C) tended to be fairly high over Design Periods 3 and/or 4 and/or 5 then declined over the remaining Design Periods. On the other hand, it is generally apparent that the time subjects spent generating hardware solution models was low - and frequently at zero - for the first four or five Design Periods but then increased markedly over the remaining Periods of the design session. Additionally, it seems that activities relating to (a) the evaluation of mathematical solution models (activity F) and (b) the evaluation of hardware solution models (activity G) generally predominated - as might be expected - in Design Periods that just lagged behind the phases of model development at the related level of design abstraction. Taken together, then, these latter results clearly support the view that the present professional engineers were developing solutions to their design problems in a highly top-down, breadth-first manner (i.e. through firstly a mathematical layer of design abstraction

and then on to a more concrete layer of hardware concepts). This result supports evidence derived from studies of software engineers that likewise indicate that professional practitioners within this domain develop their programs in a top-down, breadth-first manner (see section 2.3.2).

Some further common trends in design behaviour that are apparent from the activity distributions presented in tables 5.2 to 5.7 are that subjects are more inclined both to search for information and to communicate with the investigator at the beginning of the design session rather than at latter phases of their work. This seems again to support the proposal that the engineers were going through an elaborate phase of problem definition for the overriding design problem involving the search for information that could expand their understanding of the given design specification.

The commonality in the activity distributions of different individuals that was apparent from the comparison of tables 5.2 to 5.7 suggested that it was legitimate to combine subjects' data in order to pursue certain further computational analyses. Firstly, then, a straightforward analysis was undertaken of the total combined time (in seconds) that subjects spent engaged in activities A to M as a function of Design Period (one through to eight). The results of this analysis were plotted graphically (see figure 5.3) and, as expected, these graphs bear out the common activity trends that have already been described above. What emerges particularly clearly from these graphs is (a) the high level of problem definition behaviour (i.e. activities B, K and L) that occurs during the the initial Design Periods of a design session (b) the high level of mathematical modelling behaviour (i.e. activities C and F) that

Figure 5.3: Graphs showing the total time (in seconds) that subjects engaged in activities A to M as a function of Design Period.

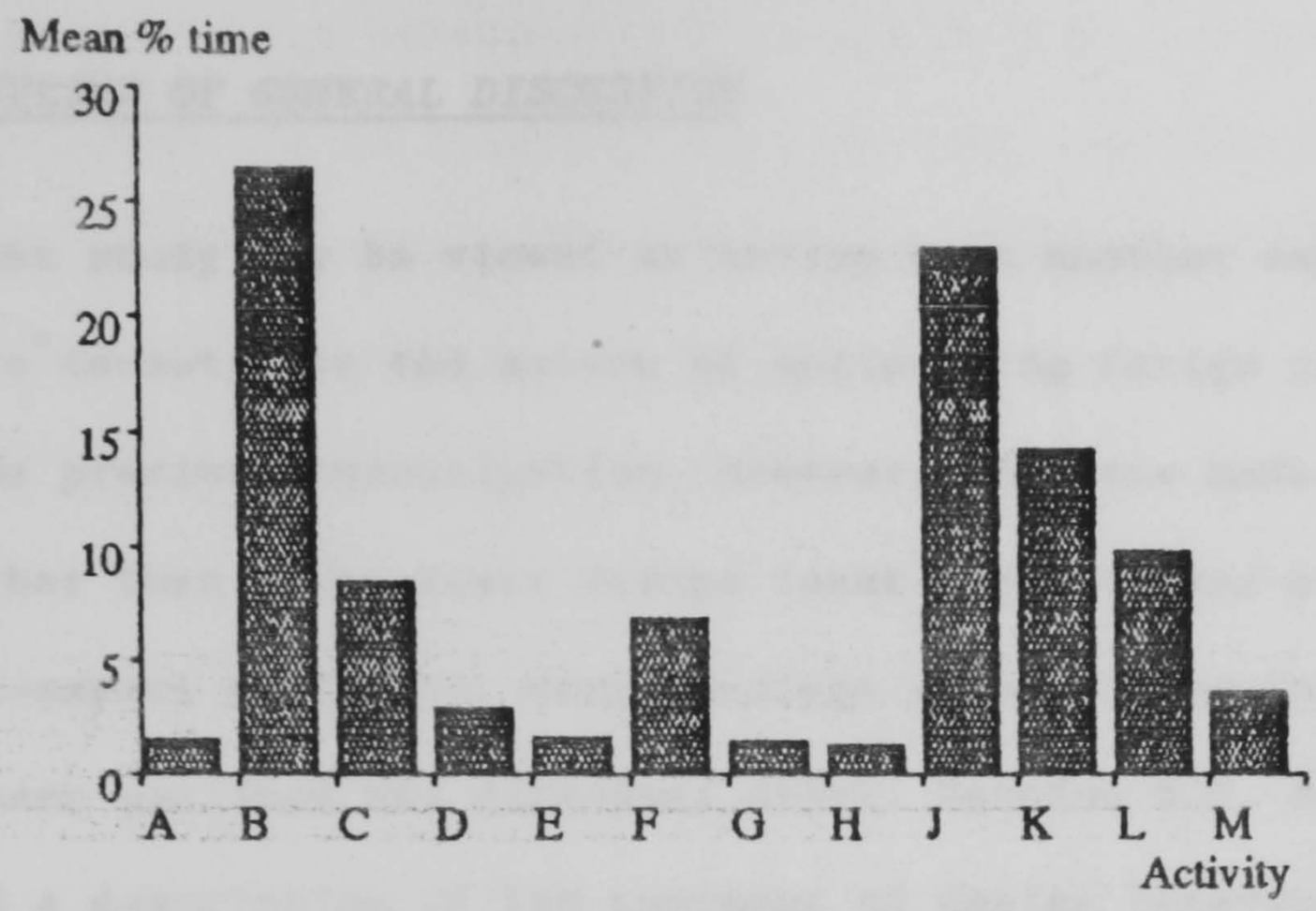


occurs during the middle Design Periods and (c) the high level of hardware modelling behaviour (i.e. activities D and G) that occurs during the latter Design Periods.

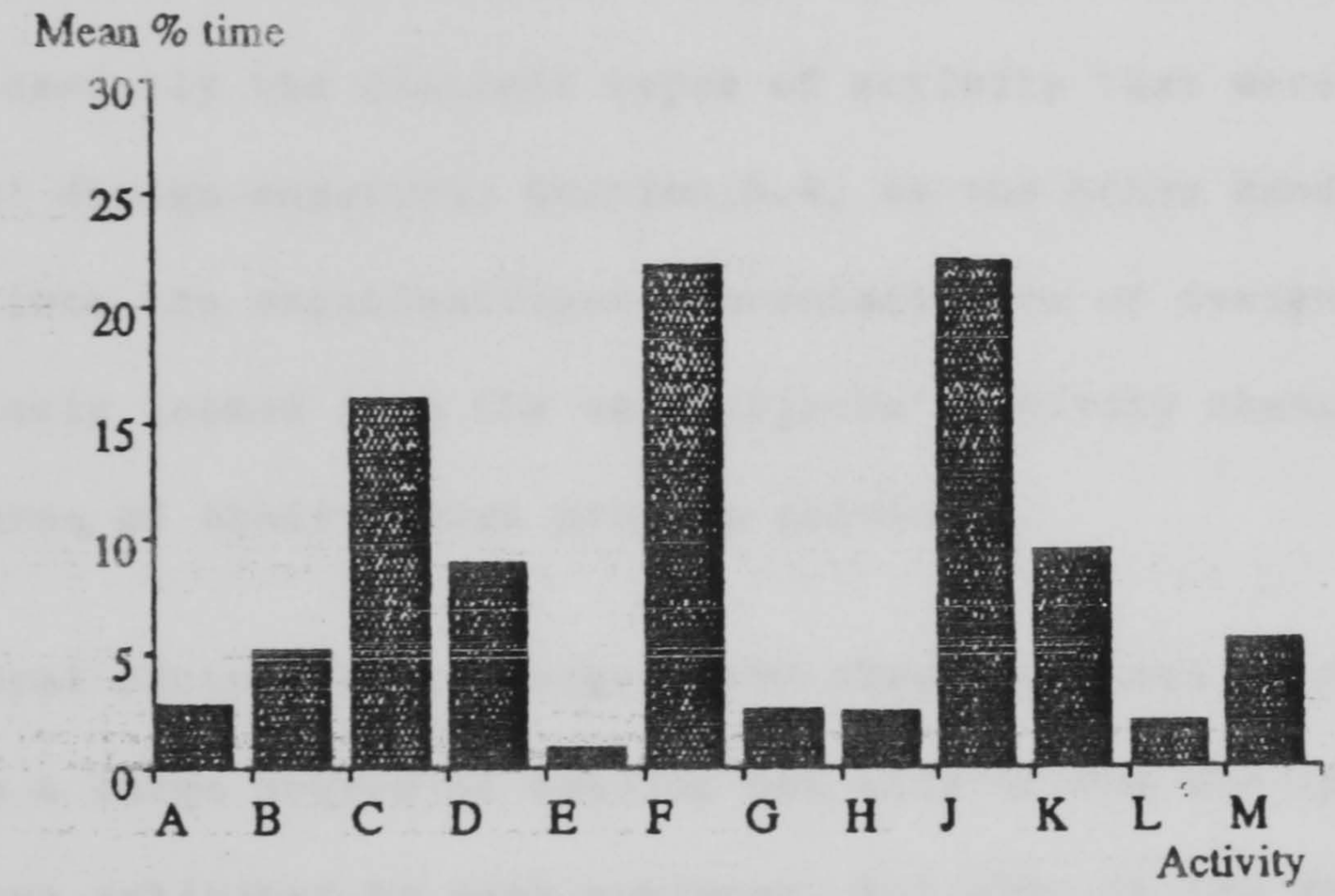
These latter results point to the fact that the present design sessions had an essentially ***three-phase*** structure involving transitions from problem definition activities (Phase 1) through mathematical modelling activities (Phase 2) to hardware modelling activities (Phase 3). This three-phase organisation of design behaviour is neatly captured in the bar charts presented in figure 5.4 which were computed on subjects' pooled data and reflect a division of design sessions into three (rather than eight) Design Periods. Each graph depicts the mean percentage time that subjects spent engaged in design activities A through to M for Design Periods 1, 2 and 3 respectively, and the activities that dominate in each design phase are readily apparent.

To conclude this section, then, it is clear that the full set of quantitative analyses that were undertaken on design protocols revealed some important aspects of design behaviour that were common across the six professional subjects who were studied. The major aspects of these commonalities in design behaviour will be further explored in the next general discussion section of this chapter which moves on to attempt a more theoretically-oriented interpretation of professional design behaviour in terms of the actual cognitive processes that underly activity.

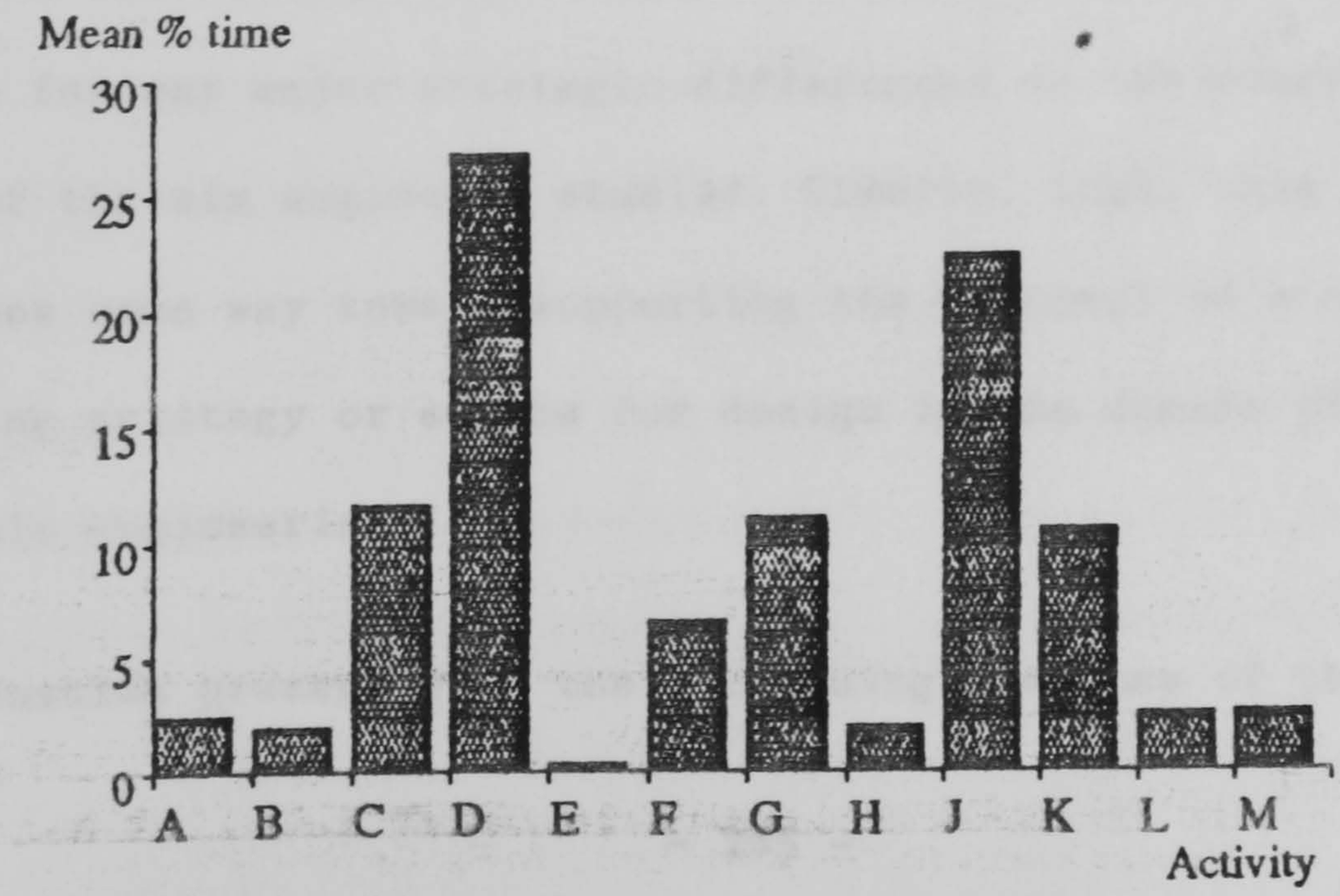
Figure 5.4: Bar charts showing the mean percentage time that subjects spent engaged in design activities A to M as a function of a three-way division of design sessions.



Phase 1: Problem Definition



Phase 2: Mathematical Modelling



Phase 3: Hardware Modelling

5.5. GENERAL DISCUSSION

5.5.1. OUTLINE OF GENERAL DISCUSSION

The present study may be viewed as having been another exploratory attempt to investigate the nature of engineering design processes. Unlike the previous investigation, however, this one made use of small-scale rather than large-scale design tasks and focussed on expert rather than semi-expert engineers. Many findings arising from this study have already been outlined and discussed above. Section 5.3, for example, presented a description of the taxonomy of design behaviours that was derived from a content analysis of protocol data and which seemed to capture usefully the distinct types of activity that were arising during subjects' design sessions. Section 5.4, on the other hand, provided some insight into the organisational characteristics of design behaviour and particularly looked into the way subjects' activity changed over the time course of their design problem solving.

The general picture that emerges from these analyses is one that portrays a large degree of overlap not only in the the specific design behaviours exhibited by each engineer, but also in the way these behaviours were structured. Indeed there seemed to be little evidence in the data for any major strategic differences in the overriding design styles of the six engineers studied. Clearly, then, this latter finding again goes some way toward supporting the proposal of a common underlying strategy or schema for design in the domain of professional electronic engineering.

The discussion presented in the continuing sections of this chapter,

then, stands largely as an attempt to draw out and clarify the major similarities and differences that were seen to arise in the present professionals' design behaviour. As in the general discussion section of the last chapter, it is likewise the case here that emphasis will be placed upon the theoretical interpretation of findings in terms of the *underlying cognitive processes* believed to be responsible for design behaviour. One important aim of this theoretically-oriented analysis will be the attempt to formulate a logically consistent model of engineering design processes that not only encapsulates the principal features of an expert design strategy but also remains largely consistent with the model presented in the context of the undergraduate study.

5.5.2. DESIGN PROBLEM SOLVING AND THE PROBLEM REDUCTION STRATEGY

In the study presented in chapter 4, subjects were considered to have a "design problem" whenever the goal state that they needed to attain was a model of a desired artifact or a model of some element of a desired artifact. Under this conceptualisation of a design problem, then, the "goal state" was considered to be a *structure* (i.e. object or component) in a hierarchically organised, multi-dimensional search space, the dimensions of which were defined in terms of *requirements* that the desired structure had to satisfy (see section 4.3.2). Moreover, in the specific domain of electronic engineering, it was believed that the requirements which constrain such a search space to the greatest degree are *functional* requirements, i.e. those relating to what the desired structure *has to do*. This characterisation of a design problem is neatly captured by the design specification given to

the engineers in the present study. This specification, then, is readily classifiable as a design problem since (1) it encapsulates an overriding goal that relates to the attainment of a model of an integrated circuit (i.e. a structure) and (2) it expresses a large number of image-processing requirements (i.e. functional requirements) that the integrated circuit should be able to perform.

The qualitative analyses undertaken in the present study revealed that "problem reduction" was a dominant aspect of the subjects' design approach, so lending further support to the evidence presented in chapter 4 for the use a problem reduction strategy by electronics designers. As was explained in this previous chapter, such a strategy appears to be particularly useful for dealing with complex, multifaceted design problems since its application produces a set of essentially independent *subproblems* which can be dealt with one at a time - so reducing the demands on the cognitive system. What was particularly noticeable about the use of problem reduction by the present engineers, however, was that the strategy not only led to (a) subproblems being generated whose goals related to the attainment of a model of *structures* which would meet existing functional requirements (as in the undergraduate study), but also led to (b) subproblems being generated whose goals related to the attainment of a model of more concrete *functions* which would meet existing functional requirements. In the former case, then, as described earlier in this section, the problem-space being searched through can be thought of as comprising "structures" (i.e. hardware components) that attain various requirements on a number of dimensions, with the dominanting dimension being functionality (see figure 5.5 for a diagrammatic representation of such

a search space). In the latter case, however, the problem space being searched through can be viewed as a comprising "functions" (e.g. mathematical operations) that attain various requirements on a number of dimensions, with the dominanting dimension again being functionality (see figure 5.6).

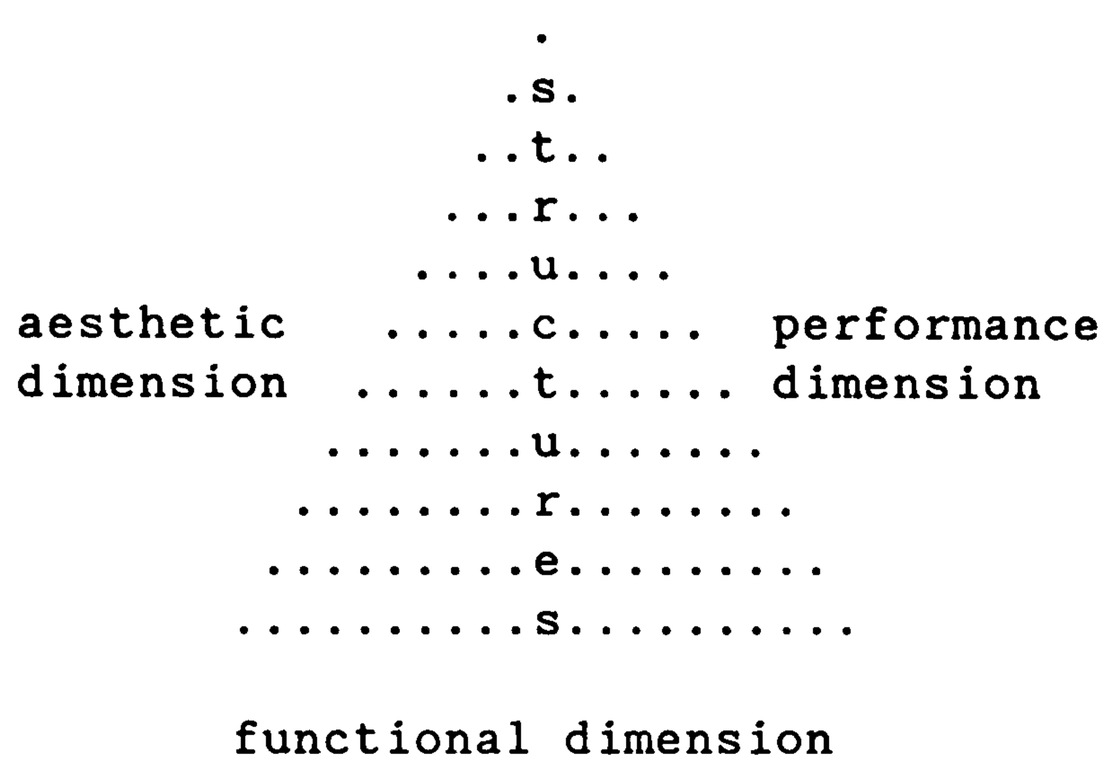


Figure 5.5: Example of a sub-space of structures that meet design requirements on three different dimensions

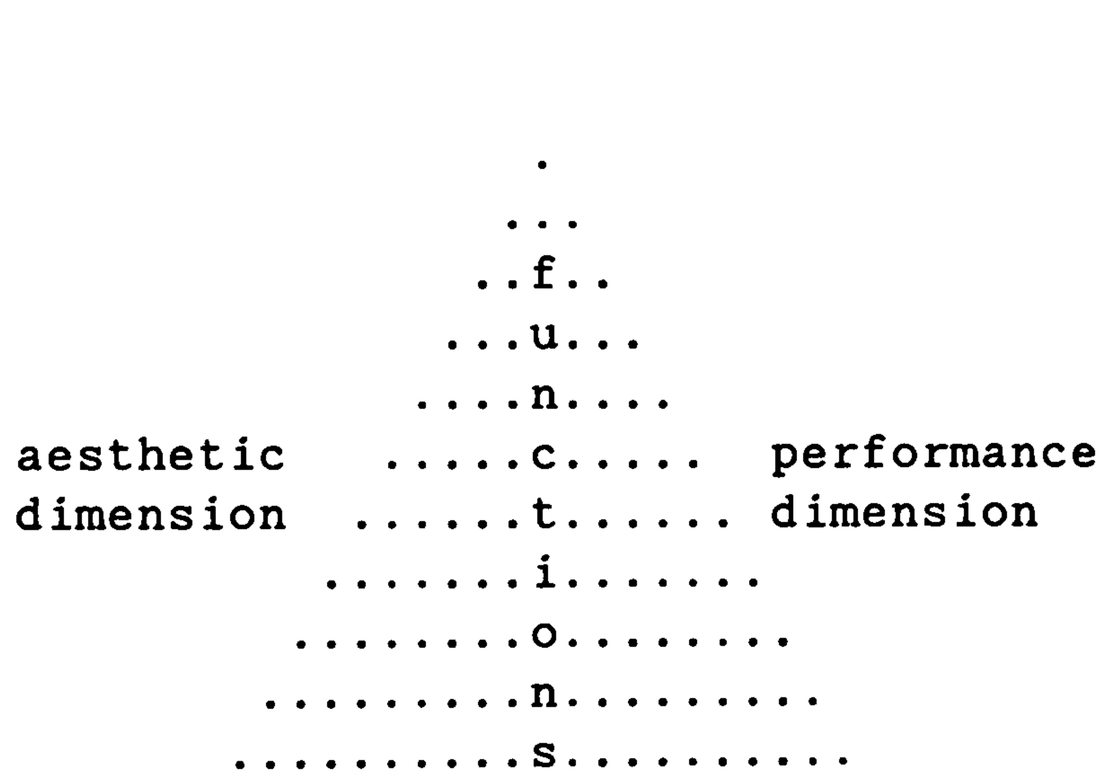


Figure 5.6: Example of a sub-space of functions that meet design requirements on three different dimensions.

The previous distinction concerning the two kinds of problem space that the present engineers appeared to be searching through for design

solutions is clearly interesting. More than its interest value, however, is the fact that the distinction provides an important extension to the understanding of design that was derived in the diary-based study. In this regard it is worth noting that it was only through the use of a mathematically-oriented design problem in the present study that evidence for the existence of these two kinds of design sub-space was found. This certainly serves to show that the use of restricted types of engineering design problem is likely to produce at best only a partial understanding of design problem solving in all its diverse manifestations. As an aside, the one factor which may account for the failure to observe the search for functional solutions in the student study may well be that the design projects being undertaken required little or no understanding and development of abstract mathematical functionality (i.e. there was little scope for the search and generation of purely functional models of subproblems). This lack of any necessity for the mathematical modelling of solutions certainly had the consequence that subjects could rapidly converge upon hardware-oriented solutions to design subproblems from initial functional requirements.

The discussion of problem reduction in design continues in the next two sections with some consideration of the way in which subproblems were seen to be generated in the present study from, respectively (a) basic models of a problem or subproblem (b) mathematical solution models of a problem or subproblem and (c) abstract hardware solution models of a problem or subproblem.

5.5.3. SUBPROBLEM GENERATION FROM BASIC MODELS

In the previous study it was observed that subjects, having attained a

structured mental representation of the functional requirements expressed within a design problem or subproblem would then proceed to generate a model of design structures that attained the expressed functional requirements. It has already been noted above that the engineers in the present study were seen to pursue searches from basic models for either (a) more detailed functional solutions or (b) more detailed structural solutions. Another important observation that derived from the present study with regard to ***basic models,*** however, was that the functional requirements expressed within a basic model would sometimes actually be ***directly*** decomposed into a number of ***subproblems*** of subfunction design (i.e. prior to the generation of any form of solution model whatsoever). This finding is again distinct from observations made in the undergraduate study which indicated that the only path from a basic model was to the generation of a solution model.

One point during the present subjects design process at which this method of problem reduction from a basic model predominated was when the engineers had gained a well-defined representation of the initial design specification itself. Subsequent to attaining such a basic model, then, the subjects were always seen to decompose the structured set of functional requirements expressed within this specification into relatively independent and well-defined subproblems of sub-function design. Usually subjects were observed to break the overriding specification into ***three*** major subproblems which related to the three separate functional aspects of the desired Integrated Circuit which correspond to the itemised functions actually numbered in the original specification (see appendix B). This reduction of the original design

specification is captured very well in a verbal sequence that is presented below which was produced by subject DS:

> "Right. OK. How do we calculate the N coordinate pairs along the line vector? Well let's work that out. Let's work that out .. Uhm .. N coord pairs along vector. Right .... Actually let's be sensible ... uh, and do like a top level thing of this thing ... Right we need to do that .. What else do we need to do? .. For each, uh, N .. do .. M .. coord .. pairs along .. normal vector.... Right .... Do (a). And (b) .. calculate .. two data items for each normal vector .. OK .. g(x) .. and .. h(x) .. Right. So, we generate those lot and for each one of those we go along that and generate those results. That's a lot better. So that's all we need to do." (DS's protocol. Start time 00.02.50)

What is clear in the verbal excerpt above (and is similarly exhibited in other protocols) is that subproblems were always derived from a splitting up of the functional requirements expressed within a higher level problem or subproblem. These subproblems therefore always expressed functional requirements at the ***same*** level of design detail as the functional requirements expressed in the original problem or subproblem and did not encompass any development of the expressed functionality into ***more detailed*** functional or structural levels of the design hierarchy. This latter activity of taking a technical design to increasingly more concrete levels of function or structure reflects a process of solution model development, the characteristics of which are discussed later on in the section 5.5.5. It is finally worth noting here that many basic models were seen to form the starting point for the development of ***solution models*** rather than just being decomposed into subproblems. This process of solution model development form basic models (see section 5.5.3) was, however, something that predominated well after the initial phase of most subjects' design work.

5.5.4. SUBPROBLEM GENERATION FROM SOLUTION MODELS

In the present study subproblems were not only seen to be generated from ***basic models*** but they were also seen to be generated from ***solution models***. As was mentioned in section 5.5.3 above, this latter observation of subproblem generation from solution models corroborates evidence for this use of this particular problem reduction method by engineers in the previous undergraduate study. What is important to note in the context of the present study, however, is that since a solution model for a subproblem could exist either as a set of interrelated functional entities or as a set of interrelated structures, then it was possible for subproblems to be generated from either a splitting up of functions or from a splitting up of structural modules. Subproblems generated by means of either of these forms of problem reduction were invariably seen to be in a well-defined state and therefore enabled the engineer subsequently to pursue a technical solution model. More detailed aspects of this process of solution model development are discussed in the next section.

5.5.5. SOLUTION MODEL DEVELOPMENT AT DIFFERENT LEVELS OF DESIGN ABSTRACTION

As has already been noted, in developing solution models the present engineers were always seen (a) to work from a subproblem that possessed a well-defined basic model of functional requirements, inputs and constraints, and thence (b) to generate solution concepts that took the design to more ***detailed*** levels of the design hierarchy. Importantly, then, whenever subjects were in possession of a well-defined basic model for any problem or subproblem they appeared to make a procedural

decision - albeit an implicit one - as to whether it was useful (1) to decompose the basic model into further subproblems of subfunction design (see section 5.5.4 for a discussion of this process) or (2) to utilise the basic model to guide the search for and development of a solution model at a functional/mathematical level (see section 5.5.3) or (3) to utilise the basic model to guide the search for and development of a solution model at a level of hardware structures (again see section 5.5.3).

One important finding concerning the process of solution model development that emerged form the present analyses was that the engineers always developed solution models at abstract design levels before then proceeding to develop more concrete refinements of solution concepts through levels of increasing design detail. It should be noted, however, that in the present study, solution models were never seen to be developed that went to levels of design detail lower than an abstract hardware level. In other electronic design situations, of course, categories of solution model applicable to more detailed design levels than abstract hardware structures are clearly conceivable (e.g. solution models relating to the design of, say, transistor based circuits or gate arrays).

As was noted above in section 5.4.3, the observation that professional engineers develop solution models breadth-first for a whole layer of subproblems at any particular level of design abstraction supports similar findings that derive from the domain of software design (see Jeffries et al, 1981; Adelson & Soloway, 1986). The expert's top-down, breadth-first design strategy is generally considered to be superior to the novice's top-down, depth-first strategy in as much as it enables the

designer to cater more readily for the functional interdependency that exists between solutions at a particular design level (e.g. allowing the engineer to exploit any similarity that emerges in the solution models to different subproblems).

5.5.6. MODELLING IN ELECTRONICS DESIGN: SOME FURTHER CONSIDERATIONS

Like the previous study, the present one again indicated that electronics engineers make use of a wide variety of external models when engaged in their design work. The kinds of external models that were produced in this study included: (1) mathematical expressions (e.g. formulae and equations); (2) two-dimensional line graphs (which generally included some additional mathematical/symbolic notation); (3) written notes (e.g. lists of functional requirements); (4) block diagrams of interconnected "functional" modules (often depicted as mathematical operators); and (5) block diagrams of interconnected "hardware" structures. As was noted in the context of the previous study, it would seem sensible to assume that the physical representations (i.e. of basic models and solution models) that subjects were producing on paper would be related in essential respects to internal representations that they were developing. The point is, then, that for any type of external model that was being developed with pen and paper there would be some form of isomorphic mental model which was likewise being developed in the mind of the designer. Furthermore, if the additional assumption that mental models are constructed and manipulated within some form of limited capacity ***working memory*** is posited, then physical models would clearly provide an essential form of "external memory" which would free working memory for further

information handling (see also section 3.5.5).

In this latter regard, one aspect of modelling activity in design that was particularly evident in the present design protocols was the very high degree of ***interdependence*** that existed between (1) the activity of developing mental models (N.B. mental models were reported in verbal terms within the TA protocol) and (2) the activity of externalising models on paper. That these two activities should overlap so inextricably clearly supports the idea that subjects' mental models were occupying some form of working memory system whose capacity is limited to a few unchunked items of information (e.g. a four or five design concepts being built up into a solution model for a problem). It is worth noting in passing that this kind of ongoing dialectic between the engineer's cognitive system and external forms of memory was an all pervasive aspect of the subjects design process. As such, then, it is clear that for CAD tools to work effectively as design assistants they must provide as seamless a link as possible between the engineer's mental models and the overt representations of these mental models (see chapter 6 for further discussion of this point).

So far in this chapter, mental modelling has been conceptualised as an activity involving the construction of fairly static mental representations of technical design problems and their solutions. That is to say, although such mental models are viewed as being representations that engineers construct and change over time, the models are still only mental analogues of real world states of affairs (e.g. sets of design objects, hardware components, mathematical functions and the like). In the previous chapter, however, the notion of mental modelling was also discussed in a rather different and more

dynamic sense to this, that is, as a means for engineers to "simulate" the behaviour of a device or system that they are designing (see section 3.5.5). Ways in which such a simulation might be undertaken were suggested to involve the engineer (a) running hypothetical data through a technical design solution or (b) engaging in cause-effect reasoning about interactions between component modules of a design system. Such simulation of a system's operations would clearly provide a useful mechanism for the engineer to evaluate the goodness of fit of a solution model to the functional requirements and other constraints expressed by a problem. What is perhaps surprising in the context of the present study was that the design protocols picked up very little evidence for subjects engaging in such mental simulation of device behaviour. Indeed the nearest that the present subjects seemed to come to pursuing design simulations was during the activity that was termed as "proving mathematical solution models". Whilst engaged in this activity, then, a subject would strive for a formal proof of a functionally-oriented solution model (e.g. a mathematical model) by running hypothetical values of, say, coordinates and slopes through the mathematical expressions within the model (see following verbalisation).

> "Uhm, I'll just try it for two examples to .. uh .. prove by induction .. 1,0 .. and just say for example that's .. 3,3 .. So that a variant 2, and that's a variant 3 .. So using Pythagoras, that's, uh, root 4 plus 9 which is root 13. Now that's the same as having an X1, Y1 which is 3,3, an X2, Y2 which is 1,0 .. Uhm .. subtracting (X1 - X2), (Y1 - Y2) .. (X1 - X2) is 2 .. is 3 .. And it's actually the square root of those two squared - which is 4 plus 9 - That's right! .. Ignoring signs .. Uh, let me just try it one more time. It should be .. Let's see .. Check .. the X coordinate .. 3 .. Yes, (3 - 1)". (JO's protocol. Start time 00.22.36).

Looking at the issue of why the present subjects generally failed to pursue mental simulations of device behaviour, the most plausible

explanation seems to revolve around the tendency for the engineers generally to avoid doing *any* truly systematic or thorough evaluation of solution concepts at all - at least at an explicit level. As was mentioned earlier in the chapter, many of the evaluations that the engineers did make of solution concepts just seemed to involve simplistic assertions as to their adequacy or inadequacy (e.g. "That's OK" or "That's good" or "That won't work"). Interestingly, on the few occasions when subjects *did* engage in more elaborate evaluations of solution concepts (apart, that is, from the activity of proving mathematical models), this evaluation process generally displayed some intriguing characteristics which are neatly captured by another sense in which the term "mental simulation" is often used in the psychological literature. That is to say, the professional engineers seemed to engage in a process of evaluation that involved a ***forward exploration*** from a current solution (actually a "depth-first" exploration through levels of increasing detail) in an attempt to determine whether it was likely to be successful at lower levels of the design hierarchy. This kind of forward reasoning through the consequences of a particular design decision was seen (1) to take the engineer rapidly depth-first to highly concrete design levels (i.e. specific hardware components and their attributes) and (2) to result finally in a positive or negative assessment of the adequacy of the solution concept or model being evaluated. Often implementation considerations were seen to be given particular weight during this modelling process (e.g. what it would actually cost to implement the solution model as hardware).

This latter kind of mental modelling is clearly rather fascinating, for it suggests that whilst the predominant design strategy of the

professional engineer is a breadth-first one that revolves around developing an integrated design solution a layer at a time, there is still potential for the engineer to engage in a process of depth-first assessment of abstract solution concepts. An example of some verbal data that implicated such a process of depth-first modelling is presented below and occurred when DS was evaluating a mathematical solution model for a subproblem:

> "Dividing is a pain .. So .. I have to .. And a possible answer ... is to .. multiply by 1 over number .. Yeah .. so we need a table .. So recip' table we need, and shifts .. It's going to be a bit of a (unintelligible word) to do that type of thing .. Well it depends what the number of N are .. 'cos you might not need that much memory to do it - maybe .. you could do a division table perhaps .. I mean you're not dividing a thousand by a thousand - it can't be a million bits ... Righty ho!". (EVALUATING ABSTRACT HARDWARE SOLUTION CONCEPTS, APPLYING KNOWLEDGE OF DETAILED HARDWARE CONSTRAINTS TO ABSTRACT HARDWARE SOLUTION CONCEPTS. DS's protocol. Start time 00.33.23)

In summary, then, this section has discussed three types of mental modelling process that appear to play some role in design problem solving. One form of mental modelling is considered to involve the construction of fairly static representations of design problems and their solutions. Other, more dynamic types of modelling process appear to be implicated, however, when engineers *evalaute* design solutions. These kinds of modelling involve the designer pursuing either (a) mental simulations to investigate the operational characteristics of integrated design solutions or (b) mental simulations that look at the future consequences of maintaining specific design solutions. Some further considerations concerning mental modelling in engineering design will also be discussed in section 5.5.7 below which looks into the effects that cognitive limitations appear to have on subjects' performance.

5.5.7. TENDENCIES TOWARD SUBOPTIMAL PERFORMANCE IN DESIGN

An analysis of the content of the present designers' work revealed the existence of a variety of inconsistencies and errors in the models that were being developed. Such errors appeared to be equally likely to occur in any phase of the design process, though some appeared to relate to the encoding of given problem information whilst others seemed to relate to the subsequent generation and evaluation of solution-oriented information.

In the present section, then, an attempt is made both to (1) characterise more clearly the types of error that were arising in subjects' design sessions and (2) consider the likely causative factors which led to the defective design performance exhibited by these professional subjects. One proposal that will be made is that cognitive factors which induce the "selective" encoding and processing of information (cf. Evans, 1989) were in all likelihood constraining the exhibition of design competence by the professional engineers studied. For example, capacity limitations of working memory might be expected to affect performance when concurrent attention to several items of information is required (e.g. in constructing and evaluating a detailed mathematical model of a problem). Specific examples of subject's tendencies toward suboptimal design performance will be presented below and will be discussed in relation to this notion of selectivity in information processing.

ERRORS IN THE REPRESENTATION OF INFORMATION

Some of the present subjects showed a marked tendency to formulate

incorrect or imprecise representations of the information contained within the actual problem statement. An error in the encoding of problem information appeared to underly a mistake made by DS which manifested itself during the development of the mathematical solution model detailed at the top sheet 3 of his design workings (see appendix B.2.2). DS was heard to state that an angle "alpha" which he had included in a mathematical solution model was the same as the parameter "k" contained in the problem specification. DS exact verbalisation was as follows:

> "Right. That's my angle ... alpha ... And we've got an alpha there as well .... So we can calculate alpha because alpha is the same as k .. " (DS's protocol. Start time 00.21.38)

DS's assumption that angle alpha and parameter k were the "same" is incorrect. How he derived this mistaken belief can only be speculated upon, however, since the above verbalisation actually contains DS's first mention of "k". Indeed this very observation suggests that the "k=360/M" equation detailed in the problem statement resided only momentarily in the focus of DS's attention during his phase of problem encoding. It is likely that the "k" was merely represented in DS's basic model of the overriding problem in an abstract sense as "having something to do with angles". Subsequently, then, when DS was actually dealing with "angles" in his solution ideas, parameter "k" was cued by association with "angles" and was equated (incorrectly) with the most salient angle (i.e. alpha) that was being dealt with.

A very similar type of problem encoding error appeared to underly a mistake made by JF in connection with the g(x) and h(x) equations presented within the specification. It can be seen form the specification (see appendix B) that these two equations contain elements

Sin(k.i) and Cos(k.i) respectively. JF certainly made some form of encoding of these elements for they were mentioned in a verbalisation produced when he was deriving a basic model of the overriding design problem:

> "Cos of (k.i) .... What an odd thing to calculate ....
> k.i. Well k is that and i is that .. Right .. OK .."
> (JF's protocol. Start time 00.12.45)

What is interesting, however, is that JF was never again heard to mention the product "k.i" but instead referred to the equations as containing the elements "Cos(theta)" and "Sin(theta)". What is even more interesting is that prior to these references to the equations, JF had already started using a "theta" symbol to denote an important angle in a mathematical solution model he was developing. What seems to have happened is something very similar to what transpired with DS's "alpha" and "k". The indications are (1) that the product "k.i" as detailed in the specification had been simplisticly represented by JF as having something to do with "angles" and (2) that through this prior association, "k.i" had incorrectly come to be linked with the angle "theta" that existed in a mathematical solution model that was being developed. Interestingly JF's angle "theta" (which he was equating with "k.i") was the same DS's "alpha" (which he initially equated with "k"). At this point it is worth noting that later on in his design session, DS was actually seen to be working with the assumption that his "alpha" was equal to the product "k.i" rather than just "k" (refer to lower half of sheet 5 of DS's design workings in appendix B.2.2). It seems, then, that both DS and JF were working with identical misconceptions which seemingly had their origins in an initial failure to adequately represent problem information. The mistake seems to have arisen from

associations made (implicitly) between information that was in fact associated only in very abstract terms, (i.e. the information "related to angles").

A final point that is worth noting in regard to subject's errors in information representation is that most subjects, having developed a basic model for the overriding problem specification, generally failed to refer back to the specification during later stages of their work. This latter observation suggests that the engineers were not checking their mental representation of the problem with the original specification as their solution models progressed to more detailed design levels - something that would seem to be invaluable for attaining a complete and accurate design solution. Indeed the mistakes made by DS and JF were, in both cases, propagated through to more detailed design levels such that the final integrated solutions that these subjects developed were highly inaccurate in terms of their image processing characteristics.

ERRORS IN THE PROCESSING OF INFORMATION

Design errors not only occurred during the encoding of the problem specification but also arose during the subsequent generation, integration and evaluation of solution concepts and ideas. One good example of such an error was provided by a subject who was engaged in elaborating (as a new functional block diagram) a previous solution model for the overall design problem (which again existed on paper as a block diagram). In this process of solution elaboration the engineer was seen actually to miss out two fundamentally important iterative loops that existed in his original diagram. This clearly resulted in an

inaccurate design solution, although this error might well have been spotted by the designer had he attempted to check the consistency of the design expansion which he had undertaken. Again this error appears to have arisen through limitations of the cognitive system in terms of working memory or attention span restrictions.

Evidence for the occurrence of notational inconsistencies was also prevalent in the design data of these professional subjects. They appeared to be particularly prone to (a) using different symbols and abbreviations to refer to the same design parameter (switching between upper and lower case letters was especially common) and (b) using the same symbol or abbreviation to refer to different design parameters. While the former type of notational inconsistency rarely seemed to cause any problems, the latter type was seen to lead to some fundamental design failures. An example of such an inconsistency was provided by a designer who, within the space of several minutes, used a "theta" symbol to refer to two different design parameters. At a later point these two parameters were incorrectly being considered as one and the same, leading to a basic design mistake which was propagated through to lower levels.

It is worth pointing out here that the present engineers' mathematical problem solving appeared to be generally very clumsy and inefficient indeed in terms of the consistency and rigour with which mathematical notation was used. Whilst such inconsistent use of notation does again seems to implicate some failings of a working memory system, it is also likely that many of the subjects - having been formally trained as engineers rather than mathematicians - have simply not been taught about the importance of using notation rigorously.

"SATISFICING" IN DESIGN

One important area in which the present professional engineers seemed to show some particular tendency toward suboptimal performance was in the actual processes of searching for and evaluating technical solution concepts. That is to say, the subjects displayed (1) a general failure to search for alternative and potentially more optimal solution concepts - particularly at high levels of design abstraction where it would prove the most useful to do this, (2) a marked inclination to stick religiously with initial solution ideas that had been generated, even when these concepts eventually proved to be increasingly difficult to pursue, and (3) only superficial and highly unstructured modelling and assessment of competing alternatives at times when such options were actually considered.

Once again, then, these findings are comparable with evidence that derived from the previous undergraduate study which indicated that subjects were expending only limited cognitive effort in the generation and evaluation of alternative solutions ideas - a tendency that was given the generic term "satisficing" in light of the fact that it implicated that designers were accepting solutions that were merely satisfactory rather than anywhere near optimal. The fact that this tendency is something which is also evident at superior levels of design skill to undergraduate levels is interesting for it suggests that either (1) subjects are generally motivated toward maintaining original solutions ideas, or that (2) a more basic set of cognitive limitations are giving rise to the effect. In regard to the first possibility, it is noteworthy that time pressure would be likely to have a profound effect on the motivational set of engineers, leading, for example to a

conscious strategy in which the designer trades off the cost to search for an optimal solution - in terms of time and effort - against the benefit of attaining a satisfactory solution as rapidly as possible. An argument against this possibility, however, is that the subjects in the present study were asked simply to take the problem "as far as they could" within the given period of time - which was actually quite considerable (i.e. between one and two hours).

The second proposal mentioned above in regard to the origin of satisficing, i.e. that it is due to some basic cognitive limitations, seems to be particularly plausible in light of the high information processing burden that design problem solving appears to place on the human cognitive system (particularly working memory resources). In this regard, it has repeatedly noted that engineers spend a large proportion of their design time actually externalising their design models onto paper - presumably to compensate for such memory limitations. Furthermore a common finding in the general literature on human thinking and problem solving (see chapter 2 for some coverage of this research) is that people find it very difficult to pursue two or more avenues of exploration during their search for solutions to problems and indeed pursue solution paths in a highly selective and sequential manner. Such evidence lends credence to the view that cognitive limitations may well lie at the heart of the satisficing tendency in design. Further evidence for cognitive limitations as being responsible for satisficing in design will be looked at in the final discussion chapter of this thesis. For the time being, however, it worth addressing a third possible cause of the phenomenon that arose in the present study.

This third causative factor relates to the fact that the present

subjects, whilst being mainly "systems engineers" who commonly dealt with design problems at high levels of design abstraction (including mathematical levels) were still not especially expert at mathematics. The idea that stems from this consideration is that the engineers may well have possessed a somewhat limited conceptual knowledge of mathematics which had the consequence that they were generally struggling to find "any" mathematically based solutions whatsoever, and clearly having found one potential solution were reluctant to give it up. Some evidence for this stems from the indication that (1) subject JC failed to make much headway with the design problem at all, (2) all subjects were at times struggling with the mathematical aspects of their problem solving and (3) all subjects showed somewhat more inclination to generate and evaluate (albeit superficially) alternative solution ideas when they were working at more hardware-oriented levels of their designs (i.e. at levels where engineers would be expected to possess the greatest quantity and variety of conceptual knowledge).

All in all, then, there is clearly a need to pursue further studies that look specifically at the issue of satisficing in design and which attempt to pinpoint the precise origins of the tendency. In this latter regard it would be particularly appealing to try and unconfound the effects on satisficing of (a) motivation (b) cognitive limitations and (c) quantity and variety of domain concepts possessed. Further considerations regarding such future work will be outlined in chapter 6.

5.5.8. A GENERIC SCHEMA FOR ELECTRONICS DESIGN?

At this point in the general discussion it finally seems appropriate to consider the implications of the present findings for the existence of

any common strategy or schema for electronics design. Bearing in mind that the first study provided considerable grounds for viewing undergraduate design behaviour as highly similar across the several subjects studied, it seems sensible to review the findings of the present investigation firstly in terms of what they say about a common schema for ***professional design,*** and only then in terms of what they say about any strategic similarities and differences between different levels of design expertise (i.e. between professional and undergraduate electronics designers).

Firstly, looking at the possibility of a generic schema for professional electronics design, the results of the present study (see especially section 5.4.3) indicated quite clearly that there was a considerable degree of commonality not only in (a) the general types of activity that engineers engaged in, but more significantly in (b) the actual organisation and sequencing of these design activities over the time course of a design session. Whilst, then, this finding strongly implicates a common schema for design at a professional level of ability, it is interesting that there were subtle divergences from such a common strategy across the professionals studied. Some engineers, for example, seemed to spend rather less effort than others in developing a well-defined basic model of the original problem specification. Similarly, some engineers such as JM, tended to exhibit more flexibility in their design style than their colleagues, whilst still maintaining a basic top-down, breadth-first, problem reduction approach in their design problem solving. Over and above such minor divergences from a generic design style, though, the fundamental view is one of a common overriding strategy to design across the professionals studied.

Secondly, then, looking at the issue of expertise in design, it is clear that the results of the present study revealed both differences between professional and undergraduate designers as well as a striking number of similarities in their methods of working. Clearly the breadth-first design strategy of experienced engineers implies that these engineers possess a more sophisticated understanding of how best to tackle design problems. How such a polished strategy develops from the depth-first approach of the novice, though, is something that is clearly open to speculation and empirical investigation. In most other respects, however, the results of this study have indicated a high degree of overlap in the design styles of both company engineers and undergraduate engineers. One point worth noting in this regard is that the verbal data obtained in the present study were often very much more detailed than those obtained in the undergraduate investigation. Although, then, such detailed data have made it possible to formulate a more *elaborate* understanding of the nature of cognitive processes in engineering design, it is, of course feasible that an equally elaborate - and very similar - understanding od design processes would likewise derive from the use of the think-aloud technique with undergraduate engineers. In this respect there clearly remains a definite need to pursue such a protocol based study of more novice electronics designers in order to be able to generalise the present findings to lower levels of design expertise.

As in the previous chapter, an attempt is made below to encapsulate many salient aspects of a professional engineer's procedural knowledge about how to design in terms of a set of high-level production rules. Clearly, this schema is only an attempt to capture the commonalities of different

subjects' design approaches and can in no way cater for the individualistic features of any single engineer's design strategy. Additionally, in as far as the schema by definition is only formulated so as to capture the engineers general approach to design it also fails to address the details of the processes by which specific technical knowledge of design objects and their attributes is continuously retrieved and integrated into the ongoing design solution.

Many of the rules that are depicted within this final schema for engineering design are, not surprisingly, identical to rules that were developed within the context of the undergraduate study. To avoid having to repeat the explanations of such rules, however, only minimal description of their functionality will be provided below, the reader being directed to section 4.5.8 for fuller accounts of their cueing conditions and the actions that follow on from their invocation. Furthermore, it is important to appreciate that a number of the productions that are presented are either completely new or are at least different in certain critical respects to the productions that were detailed in the last chapter. That is to say, some of the rules below were either (1) devised specifically in order to capture aspects of professional engineering design or (2) based to some extent on the undergraduate rules but containing differences that bring them in line with critical features of professional design. It should be noted that any productions of the professional design schema presented below that are identical - or very nearly identical - with productions in the undergraduate schema will be marked as such by the placing of an asterisk as a superscript attached to their defining rule number (see, for example, the problem selection production labelled as R1 below). As

a final point, it should be noted that the schema contains several productions that enable the system to contend with highly mathematical and functionally-oriented aspects of a design problem (something which the undergraduate schema failed to address) and it is believed that this has been undertaken in a way that does not compromise the generality of the schema to different kinds of design problem to the one that was presented to subjects in the actual study.

FINAL SCHEMA FOR ENGINEERING DESIGN

PHASE 1: PROBLEM DEFINITION

Phase 1a: Developing basic model of problem

STEP 1

```
R1*   IF (no current problem exists)
      AND (problems on agenda)
      THEN (select highest priority problem and remove from agenda)
      AND (make highest priority problem current problem p)
```

STEP 2

```
R2*   IF (p is current problem)
      AND (basic model of p does not exist)
      THEN (set goal to develop a well-defined basic model of p)
      ELSE (move to STEP 8)
```

STEP 3

```
R3a*  IF (goal is to develop a well-defined basic model of p)
      THEN (use own technical knowledge to expand understanding of
           functional requirements and constraints expressed in p
           at current level of abstraction)

R3b*  IF (Rule R3a is successfully applied)
      THEN (assert that p has a well-defined model)
      AND (move to STEP 8)
```

STEP 4

R4a* IF (goal to develop well-defined basic model of p)
THEN (seek technical advice of person who set the problem so as to expand understanding of functional requirements and constraints expressed in p)

R4b* IF (Rule R4a is successfully applied)
THEN (assert that p has a well-defined model)
AND (move to STEP 8)

STEP 5

R5a* IF (goal to develop well-defined basic model of p)
THEN (seek technical advice of a domain expert so as to expand understanding of functional requirements and constraints expressed in p)

R5b* IF (Rule R5a is successfully applied)
THEN (assert that p has a well-defined model)
AND (move to STEP 8)

STEP 6

R6a* IF (goal to develop well-defined basic model of p)
THEN (use technical literature to expand understanding of functional requirements and constraints expressed in p)

R6b* IF (Rule R6a is successfully applied)
THEN (assert that p has a well-defined model)

STEP 7

R7 IF (p does not have well-defined model)
THEN (fail and exit)
ELSE (set goal to assess usefulness of decomposing basic model of p directly into subproblems)

Phase 1b: Generating subproblems from basic model of p when useful

---

STEP 8

R8a IF (goal to assess usefulness of decomposing basic model of p directly into subproblems)
THEN (use understanding that has been gained of functional requirements and constraints expressed by p to assess usefulness of reducing functional requirements into subproblems of subfunction design)

R8b IF (R8a is successfully applied)
THEN (use understanding of problem to reduce p into subproblems of subfunction design AND use understanding to give priority ratings to these subproblems)
AND (add subproblems that are identified to agenda)
AND (assert that no current subproblem exists AND return to STEP 1)
ELSE (set goal to develop solution model for p)

PHASE 2: SOLUTION MODELLING

Phase 2a: Developing and assessing solution model for p

STEP 9

R9a IF (goal to develop solution model for p)
THEN (use understanding that has been gained of functional requirements and constraints expressed by p to assess usefulness of developing a functional/mathematical solution model for p)

R9b IF (R9a is successfully applied)
THEN (set goal to develop functional/mathematical solution model for p)
ELSE (set goal to develop structural/hardware solution model for p AND move to STEP 11)

STEP 10

R10a IF (goal to develop functional/mathematical solution model for p)
THEN (search for functional/mathematical solution concepts at next level of design detail directed by information about important functional requirements and constraints expressed in basic model of p AND generate and externalise concepts sequentially AND assess potential of concepts by applying heuristic knowledge of lower level possibilities AND maintain and integrate concepts assessed as having potential)

R10b IF (application of R10a results in solution model for p)
THEN (set goal to evaluate solution model for p AND move to STEP 12)
ELSE (fail and exit)

STEP 11

R11a IF (goal to develop structural/hardware solution model for p)
THEN (search for structural/hardware solution concepts at next level of design detail directed by information about important functional requirements and constraints expressed in basic model of p AND generate and externalise concepts sequentially AND assess potential of concepts by applying heuristic knowledge of lower level possibilities AND maintain and integrate concepts assessed as having potential)

R11b IF (application of R11a results in solution model for p)
THEN (set goal to evaluate solution model for p)
ELSE (fail and exit)

STEP 12

R12a* IF (goal to evaluate solution model for p)
THEN (check that solution model satisfies functional requirements and constraints as expressed in basic model of p AND ensure potential connectivity of solution model to other solution models at same level of design detail)

R12b* IF (Rule R12a is successfully applied)
THEN (assert that solution model for p is satisfactory AND move to STEP 16)

STEP 13

R13a* IF (solution model for p is unsatisfactory)
THEN (use understanding of solution model for p to assess viability of refining unsatisfactory solution model into satisfactory solution model for p)

R13b* IF (Rule R13a is successfully applied)
THEN (assert that solution model for p is viable)

STEP 14

R14* IF (solution model for p is both unsatisfactory and unviable)
THEN (reject solution model)
AND (augment basic model of p with understanding derived from rejected solution model for p)
AND (set goal to develop solution model for p AND return to STEP 9)

STEP 15

R15* IF (solution model for p is unsatisfactory but viable)
THEN (use understanding of functional requirements and constraints expressed in basic model of p to search for new solution concepts that satisfy failed aspects of solution model for p and generate AND integrate AND externalise solution concepts for p)
AND (set goal to evaluate solution model for p AND return to STEP 12)

Phase 2b: Complete processing of p, generating subproblems as necessary

---

STEP 16

R16* IF (solution model for p is satisfactory)
AND (solution model for p is not at most detailed design level)
AND (solution model for p exists as two or more solution concepts)
THEN (use understanding of problem and solution to decompose solution model into separate subproblems which relate to the design of separate solution concepts AND use understanding to give priority ratings to these subproblems)
AND (add subproblems that are identified to agenda)
AND (assert that no current problem exists)

STEP 17

R17* IF (solution model for p is satisfactory)
AND (solution model for p is not at most detailed design level)
AND (solution model for p exists as a single solution concept)
THEN (treat solution model for p as expressing a problem)
AND (add new problem to agenda)
AND (assert that no current problem exists)

STEP 18

R18* IF (solution model for p is asserted to be satisfactory)
AND (solution model for p is at most detailed design level necessary for the present application)
THEN (assert that no current problem exists)

STEP 19

R19* IF (no problems on agenda)
THEN (succeed and exit)
ELSE (return to STEP 1)

*INTERRUPT RULE**

IF (external social requirements demand that a problem other than current p to be given high-priority status)
THEN (halt progress on current p)
AND (delete p as current problem and add p to agenda)
AND (set high-priority status problem as current problem p AND return to STEP 2)

As in the previous chapter, a description is provided here of the intended processing characteristics of each of the rules that were considered to make up the final schema for engineering design. Since many of the rules of this final schema coincide with those of the undergraduate schema only limited explanation will provided of these. More space will instead be devoted to an account of rules that have been newly added to the schema.

R1 is the rule that enables the production system actually to get started on a the task of developing a solution for a given design problem. In brief, an initial design problem - as well as any subsequently identified subproblems - are again viewed as being listed on an agenda from which they can be selected and tagged as the "current" problem to be focussed on by the designer. It may be recalled from the previous chapter that problems and subproblems are considered to reside on the agenda in a sequence determined by a process of prioritisation.

Subsequent to any particular problem being selected as the current one to work on, R2 is invoked if the problem is not yet represented mentally, i.e. if the designer has not yet attempted to construct a ***basic model*** for the problem. In the present study, for example, this would have been the case when the engineers were initially tackling the given design specification which needed to be read and thoroughly understood. It is worth repeating here that a basic model for a problem is conceived of as a structured set of information concerning (1) the input and output parameters of the desired design artifact or module of the artifact, (2) any functional requirements that the artifact/module must satisfy and (3) any other constraint requirements that must be met. In general, the major focus of the designer during this phase of problem definition actually tends to be on the functional requirements expressed by the problem since functional requirements actually exist as the major constraints on the desired goal state, i.e. it is useless for the engineer to attain a design solution that doesn't do what it was required to do.

During the process of basic model development it is clear that the

cueing of domain-specific technical knowledge (e.g. concerning the engineering terms used in the specification) must play a fundamental role in the formulation of the basic model. R2 should therefore be viewed as encompassing processes which elaborate and embellish the problem information at the current level of design detail by the incorporation of appropriate information cued from memory. It is also clear that the basic model of a problem or subproblem must be representation that will be continually changing as new information is added or as inappropriate information is discarded.

R7 will move the ongoing processing toward step 8 whenever a basic model for a problem exists which is well defined. Step 8 has been included within the professional design schema in view of evidence which indicated that the present subjects sometimes engaged in a process of generating subproblems <u>directly</u> from a basic model of a problem (i.e. without first developing a solution model from the basic model). What was seen to occur in such situations (refer also to section 5.5.3) was that the engineer would simply split up the functional requirements detailed within the basic model of a problem such that these functional requirements were themselves treated as expressing distinct - though interrelated - subproblems.

R8, then, provides the mechanism for a procedural decision to be made as to whether the functional requirements expressed within the basic model "can" or "cannot" be usefully decomposed into problems of sub-function design. If R8 is successfully applied then R8b leads the engineer to use his or her existing understanding of the functional requirements expressed within the basic model in order to subdivide these requirements sensibly into a set of subproblems of sub-function design.

It is further assumed that these subproblems, once identified, are each given a priority rating (based on technical and/or social considerations) and are added sequentially to the "end" of the problem agenda in "forward" order of their priority (i.e. the most important problem is added to the end of the agenda first). The view that subproblems are added to the agenda in this manner is an assumption that has been made (see below) in order to enable the occurrence of the characteristic top-down, breath-first problem selection and solution development exhibited by professional designers.

If the latter process of subproblem generation from a basic model has been undertaken then the control of processing will return to step 1 (i.e. the design schema will be reinvoked). Alternatively, if R8a has not been successfully applied (in that the functional requirements expressed within the basic model of the problem were not viewed to be usefully decomposable into problems of sub-function design) then processing moves on to step 9. At step 9 another procedural decision is made, this time relating to whether it is more useful to develop a solution model for the problem at a functional level (e.g. at a level of mathematical operations) or at a hardware level (e.g. at a level of electronic structures and components). Subsequent to the procedural decision at step 9 the control of processing passes on either to step 10 (if a functionally-oriented solution model is the most useful kind of solution to be developed) or to step 11 (if a hardware-oriented solution model would be a better type of model to attain).

R10 details the procedure by which an engineer develops a functionally-oriented solution model for the current problem. The action condition of the production specifies a highly integrated set of activities that

involve the designer (1) searching a design sub-space for more detailed functional concepts (e.g. mathematical operations) that meet the functional requirements and constraints expressed in the basic model of the problem, as well as (2) generating and externalising all concepts but accepting only those that are judged to have potential for development at lower design levels, and (3) integrating such worthwhile concepts to form a complete solution model for the current problem. Subsequent to an integrated set of functionally-oriented solution concepts being generated, R11 b sets a final action for the complete solution model to be evaluated.

R11 is identical in operation to R10 with the exception that the designer is this time attempting to generate a set of hardware structures that meet the functional requirements expressed by the basic model of the problem. In this way, then, the designer may be viewed as searching a sub-space of hardware concepts rather than a sub-space of functional concepts (refer to section 5.5.2).

The professional designer's procedure for evaluating an integrated solution model for a problem is depicted within the schema as being identical to the undergraduate's (refer to section 4.5.8). This procedure is represented in steps 12 to 19. The rules within these steps encapsulate activities that (1) check whether the solution model satisfies the requirements and constraints expressed by problem, (2) ensure that the solution model interconnects appropriately with other solutions models at the same level of design detail and (3) determine the state of the solution model as satisfactory or unsatisfactory. One aspect of the professional designer's evaluation procedure that has not been captured within R11 is the activity of simulating the behaviour of

a design solution in order to assess its functional adequacy. Whilst subjects are certainly able to engage in such active mental simulation of solution ideas, at present it is unclear when such modelling is attempted and how it might be accomplished.

As in the undergraduate schema, if the solution model for the current problem is deemed to be satisfactory then it is decomposed into subproblems (see step 16) if this is viewed to be necessary. R16 ensures that all subproblems that are identified are given priority ratings based on technical and social considerations and that these subproblems are then added to the end of the problem agenda. As was noted above, prioritised subproblems are viewed as being added to the end of the problem agenda in "forward "order of their priority. This way of adding subproblems to the agenda represents a departure from the undergraduate schema in which problems were added to the agenda in "reverse" order of their priority. However, this assumption that (1) professional engineers add subproblems to the end of the agenda in a forward order of importance, when combined with an additional assumption that (2) subproblems are removed from the **front** rather than the end of the agenda, provides - in admittedly a totally *ad hoc* manner - a psychologically plausible way for the professional designers' schema to perform the characteristic top-down, breath-first problem selection and development of solutions. Whether any way exists to substantiate such speculations about this agenda mechanism in design problem solving is, however, questionable.

5.5.9.CONCLUDING COMMENTS

In conclusion, then, it is felt that the study presented here was a

worthwhile progression from the previous undergraduate investigation in as much as many of the insights gained expand the understanding of engineering design processes into the domain of the professional practitioner. One thing that is particularly noticeable when comparing the results of the present study with those of the undergraduate one is the more detailed understanding of design processes that was afforded by the use of the think aloud technique. This result clearly serves to indicate the superiority of the think-aloud approach over less sophisticated process-tracing methodologies such as cognitive diaries and interviews. On the other hand, however, it is also pleasing to see a large degree of similarity in the results of the two studies over and above methodological differences in the way the studies was undertaken. Finding such a level of commonality in the design styles of professionals and undergraduates provides encouraging grounds for the belief that there may be some generic schema for electronics design which develops with experience at design tasks. Likewise, the common evidence across expertise levels for tendencies toward suboptimal performance is also of interest as it suggests that the origin of these tendencies may well lie in some basic limitations of the cognitive system rather than in the skill level of the engineer. Further speculation along these lines, however, is perhaps more appropriately left to the final discussion chapter that concludes this thesis.

Chapter 6

Final Discussion

Chapter 6. Final Discussion

Page

6.1. INTRODUCTION TO FINAL DISCUSSION

6.1.1. Overview of final discussion..... 292

6.2. GENERAL SUMMARY AND SYNTHESIS OF FINDINGS

6.2.1. Commonalities between engineers..... 293

6.2.2. Differences between engineers..... 307

6.3. IMPLICATIONS OF THE PRESENT RESEARCH

6.3.1. Implications for problem-solving theory and research..... 311

6.3.2. Implications for Computer Aided Design..... 317

6.3.3. Implications for training and educating design skill..... 321

6.4. INTERVENTION APPROACHES TO IMPROVING DESIGN: A PILOT STUDY

6.4.1. Aims of the study..... 328

6.4.2. Method..... 331

6.4.3. Results and discussion..... 332

6.5. CONCLUDING COMMENTS

6.5.1. Future work and final comments..... 337

6.1. INTRODUCTION TO FINAL DISCUSSION

6.1.1. OVERVIEW OF FINAL DISCUSSION

Chapters 4 and 5 provided extensive and detailed accounts of the two major studies that were undertaken within the current research programme investigating cognitive processes in engineering design. Whilst some attempt was made in these previous chapters both to summarise important findings and to address their major theoretical and practical implications, the scope for pursuing such objectives was clearly somewhat limited. This final discussion chapter, however, provides the ideal opportunity for pursuing these objectives more fully. In section 6.2, then, a general recapitulation and synthesis of the salient findings stemming from the present research is attempted in which particular emphasis is placed on drawing out and discussing both the characteristic ***commonalities*** and the fundamental ***differences*** between the engineers studied in terms of (a) their basic design strategies and (b) their tendencies toward error and sub-optimal performance. This summary also looks at the present theoretical conclusions in the light of other contemporary psychological theories of design (as reviewed in chapter 2).

Whilst section 6.2 provides an integrative account of the cognitive processes underlying behaviour in domains where ***design*** is the key activity, section 6.3 broadens the discussion toward an examination of (1) the wider theoretical and methodological implications of the present work for problem-solving research in general, as well as (2) the all-important applied implications of the work for the development of

computer-based design aids and for the training and education of design skill. It may be recalled from chapter 1 that the possibility of improving design performance by means of computer-based support environments actually provided the main impetus for undertaking this research programme in the first place. In this regard then, a large part of section 6.3 is devoted to discussing how the present research findings fed in to the development of the Plymouth Engineers Design Assistant (PEDA) - a prototype version of which was implemented concurrent to the ongoing psychological research that has been presented within this thesis.

Finally - and still in relation to the issue of improving design processes - section 6.4 presents a brief description of a pilot study that was undertaken to investigate various forms of strategic intervention as being possible ways of countering tendencies toward sub-optimal performance in electronics design. As will emerge during the discussion of this pilot study, influencing design processes in a positive manner can prove an elusive goal for a number of reasons.

6.2. GENERAL SUMMARY AND SYNTHESIS OF FINDINGS

6.2.1. COMMONALITIES BETWEEN ENGINEERS

Understanding the nature of the commonalities between engineers in terms of the processes underlying their design behaviour formed one of the primary aims of the current research programme. At a purely theoretical level any evidence of similarities in cognitive processes across engineers is important in that it points to the possession of

some common procedural knowledge for design (i.e. some form of generic design schema or strategy). Similarly, any evidence of consistent tendencies toward biased or error-prone information processing is pertinent in that it suggests that either (a) designers have the underlying competence to produce optimal design solutions but such competence is susceptible to performance factors (e.g. motivation levels or basic cognitive limitations) which constrain its exhibition or (b) any underlying competence is itself deficient in certain crucial respects for the production of optimal designs (e.g. the actual rules that guide design activity may diverge from those required for optimal performance).

Considerable evidence did indeed arise in the present research indicating some major similarities in design processes across engineers - both within studies and between studies. In the latter case of comparing results *across* the two investigations, however, there is a clearly a need for caution in considering the causes of observed similarities - and differences - since any interpretation is confounded in terms of (1) the populations sampled at, i.e. undergraduates versus professionals (2) the nature and time scale of the given problems, i.e. long-term, real-world design projects versus short-term, laboratory design tasks and (3) the observational methodologies employed, i.e. cognitive diaries and interviews versus think-aloud protocols. Still, however, similarities emerged in the findings of the two studies which are certainly worth drawing out whilst at the same time bearing in mind the confounds just highlighted.

Firstly, then, the behavioural analyses undertaken provided definite grounds for maintaining the view that electronics design is a form of

"problem-solving" activity in as much as the protocols obtained in both studies readily lent themselves to interpretation within a problem-solving framework. That is to say, during the analysis of verbal behaviour it was apparent that designers - like individuals working in other problem-solving domains - were (1) engaging in activity that was directed toward the attainment of various goals and subgoals that were not immediately achievable (2) pursuing - at least at an explicit level - highly serial rather than parallel lines of solution development and (3) developing their solutions steadily by means of a gradual addition of concepts with little evidence of any rapid, intuitive leaps toward solution ideas.

It is worth reiterating here that the problem-solving conceptualisation of electronics design not only supports the views of other researches who have investigated different design domains (cf. Eastman, 1969a; Newell & Simon, 1972 Jeffries et al. 1981) but also supports a task analysis of this particular domain which indicates that any given electronics design specification can be thought of as expressing an overriding "goal" to attain a description or model (i.e. a "design") of an artifact that meets various requirements concerning functionality, structure, cost and the like. In this way, then, electronics design can be understood using traditional problem-solving terminology as involving a *search* for an artifact description in a multi-dimensional *problem space* the dimensions of which are defined in relation to the requirements that the artifact description must satisfy.

The two studies also provided considerable evidence which indicated that subjects' overriding design tasks were prototypical cases of "ill-defined" problems (cf. Simon, 1973). In particular the tasks being

attempted were always seen to require an extended phase of problem definition before the engineers could then proceed with the generation of solution ideas. Designers in the professional group, for example, were generally seen to spend on average the initial third of their design sessions engaged in activities that predominantly related to the understanding and structuring of problem information. It is worth reiterating here that phases of problem definition in design have been noted elsewhere in the literature (see, for example, Malhotra et al. 1980) and it is certain that such processing is vitally important for the engineer to make any subsequent progress with a design solution.

In the current studies, problem definition was considered to involve the designer building up a structured, mental model - or what was termed a "basic model" - of problem information. The process of developing such a basic model for any ill-defined design problem was seen to pivot around the understanding and structuring of the ***functional requirements*** that the to-be-designed artifact should meet. Other types of requirement which also tended to be clarified during basic model development included (a) requirements relating to the artifact's performance, resource usage, and structure (e.g. concerning its efficiency, reliability, power consumption, speed of operation, elegance and the like) and (b) requirements relating to the process of design itself (e.g. the availability of time, funds, tools and other resources). These latter kinds of requirement, however, were generally seen to be focussed on to a lesser extent than were functional requirements - a finding which suggests that functional requirements may be the more important for constraining a potentially vast search space of design concepts through which the engineer needs to search for a

solution. The point seems to be, then, that whilst it is clearly useless for the engineer to attain a design solution that doesn't do what it's supposed to do, there may often be some leeway as far as the attainment of other requirements is concerned.

Subsequent to the initial phase of problem definition for the overriding design specification, the present engineers were always then observed to use a simple factoring approach in tackling their design problems. This approach is neatly captured by the notion of "problem reduction" - a general problem-solving strategy which is useful for dealing with complex problems that can be easily decomposed into relatively independent subproblems. Problems in electronics design appear to be of this form since they generally express a number of functional requirements which can be readily factored into subproblems of subfunction design. For example, the splitting up of functional requirements expressed by the overriding design specification was seen to be an ubiquitous feature of subjects' design strategies in the second study.

A more common way in which problem reduction tended to be used by the present designers, however, involved the splitting up of the actual ***solution model*** that had been generated for a particular problem. Figure 6.1, for example, shows a solution model produced by a subject in the first study which, as can be seen, comprised a set of hardware concepts that were depicted as labelled and interconnected blocks. Having generated this solution model the subject then proceeded to deal with each abstract hardware concept as an independent design subproblem. In this particular case, moreover, the solution models that the subject then generated for each of these hardware subproblems were themselves

subsequently reduced into further sub-sub-problems and so on until a level of detailed electronics concepts was finally reached.

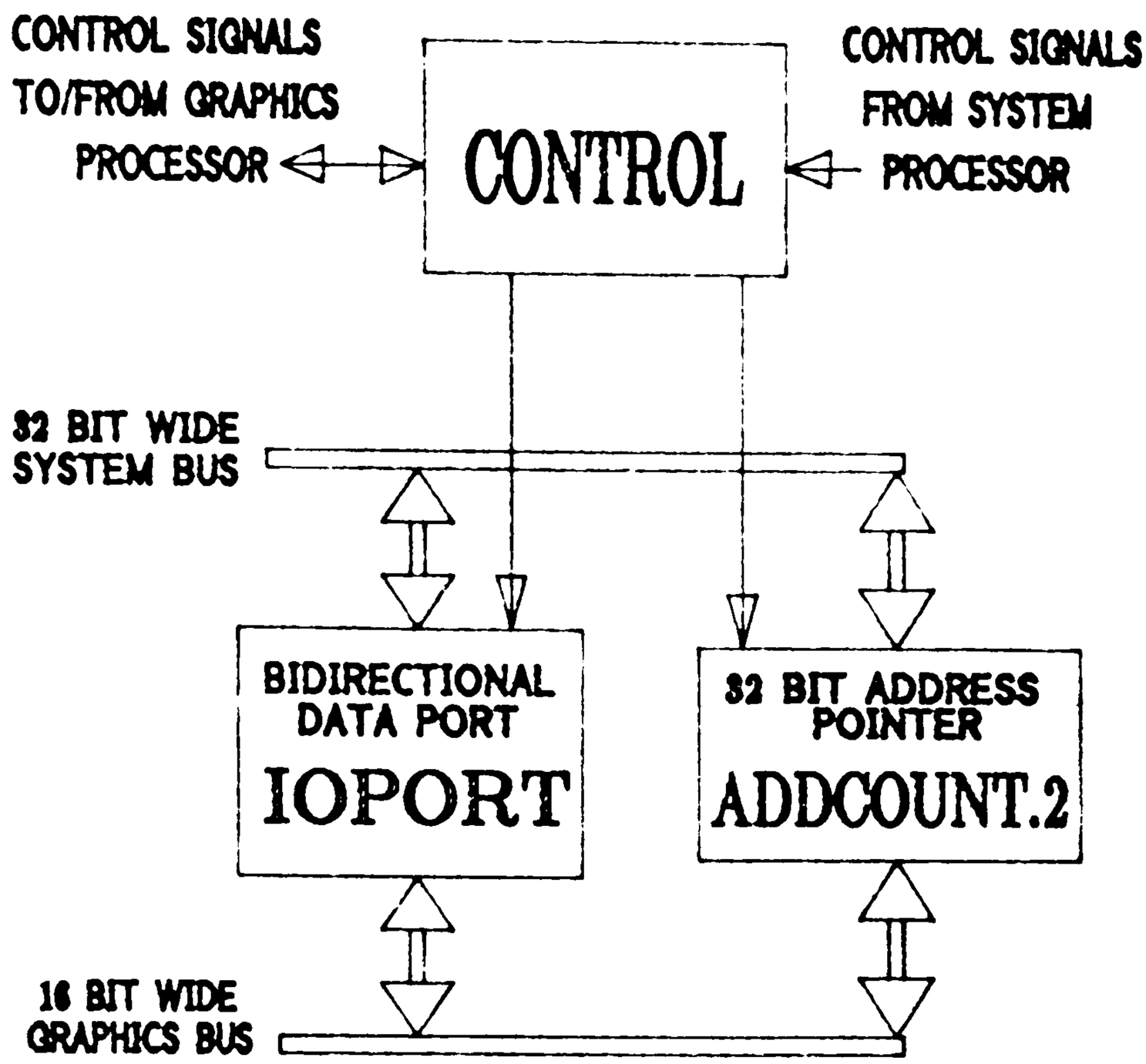


Figure 6.1: Top-level solution model generated by subject TS in undergraduate study

In electronics design, then, it generally appears that the processes of problem definition, solution model development and problem reduction occur in an iterative manner such that design problems and their solutions are transformed from an abstract conceptual level to increasingly more concrete levels of technical detail - i.e. the so-called *top-down* design method (see, for example, Mostow, 1985). It should be stressed that the adoption of a top-down approach to electronics design appears to be an eminently sensible strategy in as much as it exploits the inherently hierarchical nature of technical

concepts within the domain. The main point here - as noted earlier in the thesis - is that since an abstract solution to a problem represents a whole "class" of detailed solutions, then searching a space of abstract solution concepts is ***equivalent*** to searching a much larger space of detailed solution concepts though far less expensive in terms of the time and effort required. Evidence for the use of a top-down approach in design has also been obtained in the studies carried out Jeffries et al. (1981) and by Adelson and Soloway (1986) who investigated software design processes. The present results, however, appear to be the first to broaden this finding to the electronics domain.

The proposal that there may be some common form of procedural knowledge underlying electronics design activity (cf. Jeffries et al.'s, 1981 view of software design) is believed to have gained a fair degree support from the present research. Indeed, it is worth mentioning again that the results of the two studies provided a valuable framework for formulating a performance model which aimed to capture the basic control structure underlying electronics design behaviour. In this regard, then, a "design schema" for electronics engineering was written as a set of abstract production rules depicting the global processes that control and coordinate designers' mental movement between subproblems and their development and evaluation of design solutions. As a psychological construct it is assumed that the design schema develops as a result of an engineer's experience within the domain such that eventually the same basic strategy would be applicable to a wide range of electronics design problems having similar fundamental structures but differing in terms of their actual content. It should be stressed, then, that the design

schema is considered to be a "generic" knowledge structure in as much as it could be applied to any electronics design problem, subproblem or sub-sub-problem at any level of design abstraction. When applied in a design situation the design schema controls and coordinates (1) the order in which different problems are worked on and the definition and understanding of such problems (2) the search, retrieval and evaluation of solutions concepts and (3) the decomposition of problems or their solutions into subproblems relating to the design of component functional or structural modules.

The design schema that was formulated from the results of the present research is considered to pivot around the problem reduction strategy which engineers were observed to make use of repeatedly. Interestingly, problem reduction is a general purpose, domain-independent method for tackling problems in all kinds of fields - one of the so-called weak methods of Newell and Simon (1972). When, however, this weak method is augmented with procedures that ensure the effective retrieval and utilisation of domain specific technical knowledge, the outcome is a sophisticated higher-order knowledge structure that can function to produce solutions to a wide range of design problems. Indeed in this regard, one interesting idea that was proposed earlier in the thesis was that there might not only be a abstract, content-independent schema for electronics design but also a multiplicity of more content-dependent schemas which specific relate to technical domain concepts (e.g. hardware components and their attributes) and which contain procedural knowledge concerning how such concepts are to be integrated with other concepts and the like. Presumably, then, such "technical-concept schemas" would be invoked by information present within the basic model

of the current subproblem - in which case control of processing would temporarily pass to the technical schema and would then return to the overriding design schema. This notion of a hierarchical arrangement of embedded schemas in design certainly seems compatible with the data obtained in the present studies though the precise mechanism by which the exchange of control between schemas might be successfully attained is something that remains open to question.

Like the "schema" notion, another contemporary psychological concept that played an important role in the present research during the analysis and interpretation of data was that of "mental modelling". Indeed, the present engineers were considered to be engaging in three distinct types of activity that could be defined as forms of mental modelling. Firstly, then, subjects appeared to be constructing, manipulating and evaluating fairly static internal models of their design problems and the solutions to these problems. In this sense, then, the term mental model is being used in a similar way to that proposed by Johnson-Laird (e.g. 1983), i.e. to refer to a mental representation which is built up as a structural analogue of a real world state of affairs. In electronics design such models were primarily seen to relate to technical, domain concepts and included the "basic models" (i.e representations of functional requirements and other constraints) and the "solution models" (i.e. representations of integrated solution concepts such as hardware components or mathematical functions) which have already been referred to earlier in this section.

One further point of interest with respect to this latter type of modelling activity was that subjects not only seemed to be developing *technical models* of problems and solutions but also appeared to be

constructing *social models* that reflected factors originating in or relating to the social world (e.g. the goals of clients, collaborators and supervisors and the needs of end users). Technical and social models appeared to interact heavily during the design process in that design constraints of a social origin would often restrict technical solution possibilities. Whilst, then, the technical and social models can be viewed as conceptually distinct, it seems clear that (a) the design schema (see above) must permit a mapping between the two to occur and (b) that it is important for good design that such connections are indeed made. In the latter respect much anecdotal evidence exists to suggest that many examples of unsaleable products arise from technical design having taken its head without reference to social realities such as the needs of the eventual consumer of the product. Further implications of this technical-social distinction will be pursued in section 6.3.

Turning, now, to consider a second, and rather more dynamic sense of the mental modelling notion, it appeared that the present subjects were able to "simulate" mentally the behaviour of a device or system that they were designing in that they were sometimes seen to (a) run hypothetical data values through their technical solutions and (b) engage in cause-effect reasoning about the interaction between elements of their technical solutions. Such mental simulation appeared to serve either as a means for engineers to evaluate their solution ideas or to explain their characteristics to another person. Regarding the evaluation of solution models it should be stressed that the present engineers frequently appeared to undertake fairly unsystematic and cursory assessments of solution concepts which often took the form of simplistic

assertions as to their adequacy or inadequacy. It is, of course, possible that evaluations were occurring rapidly at an implicit level of thought which would thereby not be manifest within the protocol data. Whilst, however, such rapid pre-attentive processing might be expected to prove successful as far as the assessment of small-scale solution concepts were concerned, it does seem unlikely that it could be effective for the evaluation of more complex and multi-faceted solution models. The point is, then that any assessment of these latter kinds of of solution model would seem to necessitate the extensive use of external memory aids (e.g. pen-and paper) for any truly thorough assessment to be attained.

Continuing with the theme of mental modelling in the evaluation of design concepts, one interesting finding to emerge from the second study was that on the few occasions when subjects ***were*** seen to pursue more elaborate evaluations of solution models they usually did this by means of a rapid but explicit "forward exploration" from the current solution to assess its consequences at lower levels of design detail. This process of forward exploration which took a design concept rapidly depth-first through more detailed levels of the design hierarchy may be viewed as the third form of mental modelling that was displayed by the electronics designers studied. Its value in the design domain seemed to lie in the fact that it was a quick yet systematic way for the engineer to test the viability of maintaining a potential solution concept. It should be noted, however, in line with the satisficing tendency discussed above, that when any concept actually failed this - or indeed any other form of assessment - then the concept would usually be patched and improved rather than totally rejected.

So far in this section, then, many key commonalities between the engineers studied have been summarised, the majority of which seem compatible with existing psychological research that has been carried out within design domains other than engineering. What has not been dealt with yet in this summary section, however, is evidence that arose from the studies relating to generic tendencies toward sub-optimal performance in design. One particular finding of importance in this respect was that subjects in both studies were seen to exhibit a "satisficing" tendency (see, for example, Simon, 1981) in that they (1) often focussed selectively on the first satisfactory - or nearly satisfactory - solution generated for any problem (2) generally failed to search for any substantially alternative solution concepts with the aim of optimising design choices and (3) only undertook cursory and unstructured modelling and assessment of competing alternatives during the few times when such options were actually considered. Basically, then, subjects seemed to accept any solution for a problem that was "good enough" (e.g. cheap enough to implement or fast enough in operation) rather than looking further for a solution that, even if not "best" (i.e. cheapest, fastest), was at least somewhat "better" than the first solution generated. It is possible, of course, that these subjects were engaging in some form of rapid and covert exploration and evaluation of alternative solutions, although such intuitive processing, if it were occurring, clearly has an air of superficiality about it.

The notion of satisficing in design was introduced by Simon (1969) who advocated it as the basis of an acceptable procedure for finding a satisfactory solution to a problem in the absence of a method for finding an optimal solution. Whilst, however, this argument is

reasonable enough, it seems certain that in a competitive, profit-oriented design climate, design solutions that are merely satisfactory are rarely going to prove to be a truly cost-effective solution. With respect to satisficing in design it is finally interesting to note that evidence of such a tendency has also been observed in a the strategies of a group of mechanical engineers who were studied by Ullman, Stauffer and Dietterich 1987 (see also Ullman, Dietterich & Stauffer, 1988). The results of this study - which has only recently come to the attention of the present author - led Ullman et al. to conclude that: "There is no global search that constructs whole alternative (design) proposals in detail and evaluates them to select the best. Our designer subjects are satisficers, not optimisers".

It would seem, then, that the use of satisficing in design is a fairly global tendency that transcends a variety of expertise levels and problem types. Ullman et al. (1988) suggest that satisficing arises from the fact that multiple, competing solutions to a problem are too complex to be handled by the human cognitive system. These authors seem to be alluding, then, to the possibility that satisficing reflects the effect of working memory limitations which constrain underlying procedural competence (i.e. competence that is in principle capable of producing optimal design solutions). The idea that basic cognitive limitations lie at the heart of satisficing certainly seems very plausible given the high information processing burden that design appears to place on mental resources. As has already been stressed previously in the thesis, however, satisficing can also be interpreted in a variety of other ways. It is possible, for example, that the phenomenon reflects either (1) a simple lack of motivation to seek alternative solution concepts when a

congenial, satisficing solution has been attained or (2) a lack of technical domain knowledge of solution concepts which means that engineers will be thankful to stick with any solution ideas that they come up with or (3) the possession of procedural knowledge (e.g. rules within the design schema) that diverge from the procedures needed for the production of optimal design solutions as recommended by formal engineering design theory (see, for example, Hubka, 1982; Pahl & Beitz, 1985). Clearly, then, the range of possible determinants of the satisficing phenomenon in design are numerous and no firm conclusions as to its origin seem to be implicated by the present data set. As will be discussed in section 6.3.2, however, an understanding of the causes of satisficing is design is something that would certainly be crucially important if any attempt to alter the tendency in engineers were to prove successful.

Continuing with the theme of sub-optimal performance in electronics design, then, the use of the think-aloud technique in the second study was particularly successful in revealing a number of systematic weaknesses in behaviour including (1) errors in the representation of the given problem information and (2) notational inconsistencies during the modelling of design solutions. In this regard, it once again tentatively suggested that factors which induce the "selective" encoding and processing of information (see particularly Evans, 1989 for a discussion of such factors) may well have been constraining the exhibition of design competence by the professional engineers studied. For example, the lack of salience or vividness of given problem information would be expected to lead to failures to attend to important aspects of the specification, whilst capacity limitations of working

memory would be expected to have profound effects on performance when concurrent attention to several items of information is required (e.g. in manipulating mathematics, when investigating competing design options or while expanding designs to more detailed levels).

6.2.2. DIFFERENCES BETWEEN ENGINEERS

The issue of individual differences in engineering design was given rather limited coverage in the context of the previous studies since the interpretation of results tended to be directed toward the nature of commonalities in design problem solving. Clearly, however, the question of individual differences in design is an important one and an effort will be made here to pursue it to a greater degree than was attempted in previous chapters.

Dealing once again with the important theme of strategies in electronics design, it is worth reiterating a salient difference that emerged in the design styles of engineers participating in the two separate studies. That is to say whilst the engineers in the first study were generally observed to tackle their design problems in a top-down, ***depth-first*** manner, the engineers in the second study usually pursued their design problems in a rather different, top-down, ***breadth-first*** way. Again it should be stressed that although it is very tempting to attribute differences in the results of the two studies to individual differences between the skill level of the participants (i.e undergraduates versus professionals) caution must be taken in going to far along this line of argument since the results of the studies are confounded in terms of other crucial dimensions (i.e. type of problems given and methodology of observation). Still, however, the observed co-occurrence of these

contrasting design strategies with different levels of design expertise should not be dismissed too readily for it does map on to evidence from studies of software engineers (see particularly Jeffries et al., 1981) which points to depth-first methods predominating in novice programmers and breath-first method predominating in more expert programmers.

The contrasting depth-first and breadth-first design strategies are illustrated schematically in figures 6.2 and 6.3 which respectively show the sequence (i.e. numerical ordering) in which a hypothetical set of subproblems might be pursued by the adoption of one or other of these design approaches. Within these figures problems have been numbered hierarchically so as to indicate the higher level problems from which they have been derived. The breadth-first approach is generally considered to be a superior method in design since it enables the engineer to cater for the interdependency between design problems at a particular level of design detail (e.g. the solution for one subproblem may clearly constrain what is possible for other subproblems at the same design level). The general argument for how designers come to have such a polished schema for design (which includes the breadth-first method) is that the schema develops over time as a result of an individual's experience and feedback with a multiplicity of domain problems (i.e the individual learns that important dependencies often exist between different design subproblems).

Considering further the issue of individual differences in design strategy, however, it is interesting to note that engineers within a particular skill level frequently showed a degree of flexibility in

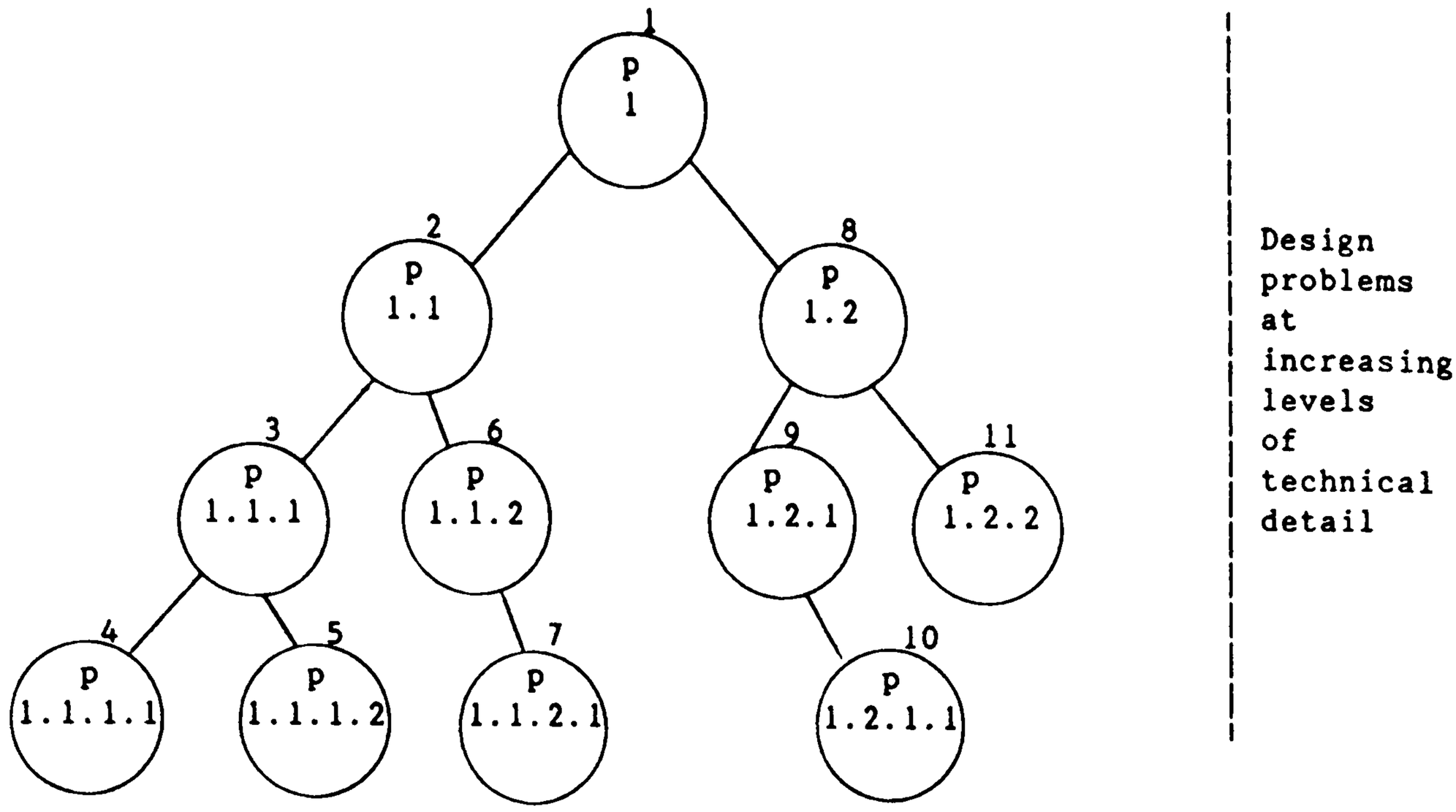


Figure 6.2: A hypothetical set of subproblems showing a typical depth-first order in which these might be tackled

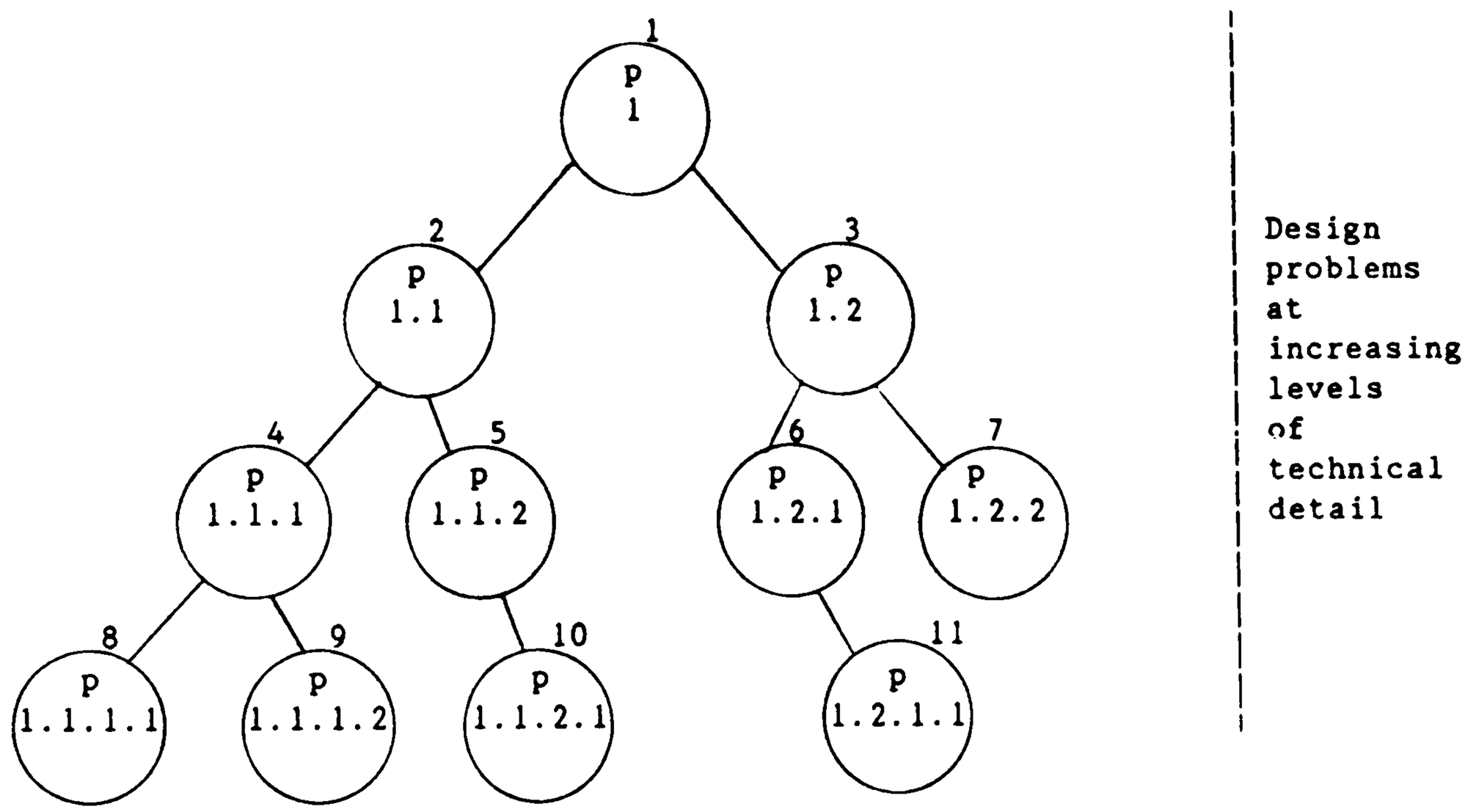


Figure 6.3: A hypothetical set of subproblems showing a typical breadth-first order in which these might be tackled

their adherence to rigid depth-first or breadth-first design approaches. One reason why divergences from a particular design approach

occur clearly relates to the nature of external influences on the individual, i.e. design problems might have to be re-prioritised as a result of requests by collaborators or supervisors (something that was commonly seen in the undergraduate study). However, in the laboratory-based professional study, design flexibility was observed even without the possibility of external, social factors having an influence. This suggests that some degree of design flexibility is an inherent aspect of the engineer's individual dialectic with the specific problem at hand and again this is likely to be tied up with the engineer's prior experience within the domain.

It should be stressed that the final schema for electronics design that was devised as a result of the two previous studies was specifically formulated so as to capture the nature of strategy differences between subjects. In particular, the agenda mechanism was incorporated into the schema partly so as to provide a means for the engineer to exert an individual influence on which design subproblems should best be tackled next. It may be recalled that the agenda provides the engineer with a current list of all unsolved design problems which each have an attached priority value (with priorities normally mapping on to a breadth-first or depth-first scheme). Flexibility is permitted, however, by the engineer being allowed to re-prioritise subproblems at various points during the design process. Clearly, then, the agenda (in association with relevant production rules within the schema) provides a psychologically plausible account of how divergences from a breadth-first or depth-first scheme may be implemented by the designer.

6.3. IMPLICATIONS OF THE PRESENT RESEARCH

6.3.1. IMPLICATIONS FOR PROBLEM-SOLVING RESEARCH AND THEORY

This section attempts to explore briefly some of the implications of the present programme of work for general research and theory relating to human problem solving. Perhaps the first point worth making in this regard, then, is that the studies undertaken here are rather different in their focus from a lot of other contemporary empirical work that has been undertaken within the general field of problem solving. That is to say, whilst research has certainly made a move away from the study of the well-defined, domain-free puzzles which dominated work in the 1960s and 1970s, research over the last decade appears to have remained fairly focussed on problem solving with either (1) tasks that relate to real-world domains of ability whilst still being essentially "well-defined" in nature (see, for example, the studies of physics problem solving by Chi, Feltovich & Glaser, 1981) or (2) tasks which whilst being "ill-defined" in nature are essentially domain free - the prototypical cases here being the classic "radiation problem" (see Duncker, 1945) and in its many recent variations (e.g. Gick & Holyoak, 1980, 1983; Keane, 1988; Holyoak & Koh, in press). In marked contrast to the focus of these latter kinds of research effort, however, the present research programme has investigated problem solving in a real-world domain (electronics design) in which the tasks also typically arise in an <u>ill-defined</u> form.

It is the relative novelty of the present studies (i.e. investigations of real-world problem solving with ill-defined problems) which underlines the potential importance of the work for problem-solving

research in general - not only at a theoretical level but also at a methodological one. Looking firstly, then, at the methodological side of things, it is particularly worth mentioning that in approaching a novel problem-solving domain such as engineering design, recourse to conventional psychological research methods (i.e. experimental studies with well-defined dependent variables) generally appears impractical. The point is, then, that in having to investigate a domain without any clear theoretical understanding of the kinds of processes that normally apply, one has little to guide the setting up of neat experimental tests of specific hypotheses. Certainly, the work at this phase has to be exploratory, which means that it ends up - as it did in the present studies - being (a) focussed on a few individual domain experts (b) concerned with the content of ongoing behaviour and (c) essentially descriptive and qualitative in nature. More than this, however, is the fact that it may often prove necessary to develop completely new kinds of investigative technique in order to study phenomena of interest. In the present case, for example, in order to investigate the problem solving of undergraduates tackling extended design projects, a structured diary had to be developed so as to monitor ongoing design activity. The research difficulties, however, often only truly begin after the actual acquisition of qualitative raw data, for it is then necessary to devise some way of encoding and analysising such material. Once again, in the present research a new type of method for converting subjects' diary comments into structured representations (or what were termed "design behaviour graphs") had to be developed and applied.

Also on the methodological front it is important to point out the valuable role that the "think aloud" method can play in exploratory

research within problem-solving domains. This was borne out in the present research by the apparently successful application of this methodology in the second major study which focussed on individual designers tackling small-scale design tasks. Still, however, the method is by no means an easy one to apply for it invariably requires extensive phases of work relating to (a) the transcription of protocols (b) the derivation and application of an encoding scheme and (c) the final theoretical interpretation of encoded protocols. Indeed it is the present author's belief that it is the often time consuming and arduous nature of carrying out exploratory research in domains where the problems are complex and ill-defined and where solution times are usually extensive which has contributed to investigators shying away from such work. In the present atmosphere where applied and interdisciplinary work is encouraged, however, it is likely that understanding within such domains will soon burgeon.

Turning now to look at some of the implications of the present studies for problem-solving *theory*, it principally seems worthwhile comparing the major theoretical conclusion of this work with existing theoretical ideas about cognitive processes in problem solving. One point that can immediately be made in this regard is the very valuable role that current theoretical notions played in the interpretation of the present design data. Indeed it appeared that much of the present engineers' behaviour could be interpreted usefully and informatively within the framework of well-established theoretical ideas about the nature of problem solving. As a case in point consider the "schema" concept which was introduced by Bartlett (1932) in the context of memory research and which has shown a marked resurgence in popularity (see Rumelhart, 1980;

Alba & Hasher, 1983; Holland et al. 1986 for discussions of the schema notion). The basic idea behind the concept (see also section 2.2.4) is that the schema exists as a knowledge structure containing abstractions of reasoning and problem solving procedures learned to be effective in contexts sharing structural features in common. The concept is, therefore, particularly useful for explaining how analogy can play a role in the acquisition of problem solving skill for domain problems that have similar structural features but which differ in terms of their specific content. In relation to the present studies, then, the indication was that experienced designers have acquired strategies and methods of working which can readily be described at a high level of abstraction in terms of a common design schema (see section 6.2.1 above).

The design schema is particularly interesting in that it appears to revolve around a "problem reduction" approach - itself a general-purpose method which research has indicated is frequently applied in tackling both ill- and well-defined problems that can be decomposed into relatively independent subproblems (see, for example, Voss et al., 1983a, 1983b). The finding that problem-reduction is used in domains where the tasks are ill-defined (such as design) is encouraging as far as a general-purpose view of problem-solving strategies is concerned (as espoused, for example, by Newell & Simon (1972). It should be stressed, however, that the present research has also indicated that there is much more to design than just the application of a general problem reduction method. For example, many schema rules must also control and coordinate phases of problem definition for design problems as well as the specialised processing (e.g. retrieval, integration and evaluation) of

domain specific conceptual knowledge relating to actual technical design solutions. Indeed the possibility that a designer's knowledge is organised as a hierarchically embedded set of technical schemas (see section 6.2.1) is certainly not challenged by the present research. The latter idea is particularly appealing as it allows for the domain-specific aspects of design skill to be catered for whilst allowing control and coordination of the design process to ultimately reside with the top-level "design schema" itself. In the balance, then, it appears that there are both domain-independent and domain-dependent aspects to problem-solving in electronics design - and this is also likely to be the case within other real-world areas of problem solving skill.

Another theoretical notion which, like that of the schema, proved useful for understanding the nature of knowledge in engineering design was the "mental models" concept (see section 2.3.3 for a discussion of the application of mental models theory in research on problem solving, thinking and reasoning). As has already been outlined in section 6.2.1, three distinct types of mental modelling activity were implicated by the design data obtained in the present research including (1) the essentially static mental modelling (i.e. representation) of aspects of design problems and their solutions (2) the dynamic mental "simulation" of the behaviour of these latter types of models and (3) the rapid, depth-first mental exploration of a current design solution through levels of increasing design detail in order to evaluate its viability. Clearly, then, the mental models notion is a valuable one in problem-solving research as it can be applied in a multiplicity of useful ways to interpret the nature of the cognitive processes underlying activity and the representations that these processes operate upon. Continuing

with the theme of mental modelling in problem solving it is worth briefly emphasising the finding that the present designers were not only developing technical models of their design problems but also social models reflecting goals, requirements and constraints of a social origin. Whilst it is clear that such social models must be developed by problem solvers working in real world contexts, the importance of such modelling appears often to be overlooked by researchers.

Before concluding this section it is worth mentioning the fact that the interpretation of the present design data made use of both the mental models as well as the schemas notion - something that is relatively novel in problem-solving research. Moreover a serious attempt was made to apply these concepts in as clear and well-defined manner as possible in developing an integrated explanation of design behaviour such that they might capture conceptually distinct aspects of cognition (e.g. generic strategies, mental representations, simulation processes, forward reasoning processes or whatever). All of this arguably contrasts to a common trend within the problem-solving literature for researchers either (1) to develop theories based exclusively around the schemas or the mental models notions, or (2) to use both terms as if they were essentially interchangeable or (3) to use the terms in a highly ill-defined manner. The merit of drawing on both kinds of theoretical concept in the explanation of design behaviour is here considered to have been most worthwhile in the present research and is likely to be equally valuable in the context of other problem-solving domains.

6.3.2. IMPLICATIONS FOR TRAINING AND EDUCATING DESIGN SKILL

There appear to be two fundamental issues relating to the training and education of design skill which the present research has some bearing upon. The first issue concerns the possibility of teaching engineering design as a general domain skill, i.e. with only limited recourse to the content of any specific examples of domain problems. The second, and related issue, concerns the possibility of improving or optimising design skill in the sense of removing any tendencies toward biased, inconsistent or error-prone information processing.

Looking firstly, then, at the issue of teaching engineering design as a general procedural skill, it is initially important to stress that the present research seemed to provide a good deal of evidence for the view that there is a large strategic component to electronics design skill that is both (a) common across engineers and (b) generic to different problems and subproblems within the domain. This general procedural knowledge for electronics design (which is considered to reflect the existence of an abstract design schema) is interesting in as much as it suggests that electronics engineers, through experience and/or training within their domain have developed a high-level approach that can, in principle, produce at least satisfactory solutions to a multiplicity of design problems. More importantly, however, are the applied implications of finding evidence for a generic design schema - in particular the possibility of directly training novices by imparting <u>abstract</u> procedural knowledge.

In this latter regard, then, it is briefly worth mentioning some recent research that has been conducted on the training of both statistical

reasoning and logical reasoning by means of abstract rules and schemas (see Holland, Holyoak, Nisbett, & Thagard, 1986 for a review of this work). One main finding of interest derives from the study by Fong, Krantz and Nisbett (1986) in which they attempted to train experimentally the use of the statistical law of large numbers using either rule- or example-based tuition, both or neither. The results indicated that training of either a rule- or an example-based nature proved effective in facilitating the use of the law of large numbers on subsequent problem solving. In design contexts, then it might be possible to observe corresponding training effects of abstract rules to those observed within the statistical domain; clearly some research investigating this possibility would be worth pursuing.

In a similar vein to the latter results, another important finding, this time time relating to the training of logical - as opposed to statistical - reasoning, derives from an extensive study of instructional training on a logical task reported by Cheng, Holyoak, Nisbett and Oliver (1986). In their third experiment, these researchers showed that training subjects in the use of an abstract "obligation schema" (whose rules are isomorphic with those of formal logic) had a profound effect on their ability to solve problems that lent themselves to an interpretation within this schema. All in all, then the findings just reviewed are particularly encouraging in that they suggest that it may be possible to train abstract procedures (rules, schemas or whatever) for use in design contexts away from the content of specific design problems, Clearly, though, considerable research is needed in design domains to pursue the validity of these ideas. It also seems important to point out here that whilst the training of design skill by

means of instruction at an abstract, example-free level may well be possible, this does not mean that the traditional example-based training that dominates in this field (see, for example, Pahl & Beitz, 1985) is ineffective. Indeed the effectiveness of training with problem and solution examples is clearly borne out by the fact that the undergraduate designers studied in the present research (who are known to have been trained via such traditional methods) were generally very successful at their design projects. Bearing in mind this point, then, the suggestion is that there may be valuable roles for a multiplicity of convergent training approaches to design, including example-based methods as well as methods involving instruction in abstract principles.

The second issue relating to the training of design skill that was highlighted above - i.e. concerning the possibility of improving or optimising design skill by removing any tendencies toward bias and error in information processing - is clearly a difficult issue to address in light of the present findings. The point is, then, that whilst it seems clear that biased processing is going on during design (as reflected by the tendencies toward satisficing and notational inconsistency) it remains to pinpoint the causative basis of these observed deficiencies. Clearly whether satisficing reflects (1) a motivational factor (e.g. resulting from time pressure) or (2) defective rules within the design schema or (3) a more basic set of cognitive failures, is something which will dictate what improvements might be afforded by training regimes. For example, if satisficing has a large motivational component then instruction might be expected to have some value for encouraging subjects to generate and pursue alternative design solutions. Similarly, if subjects are using defective rules and procedures during their design

work, then it might be possible replace such rules with ones that would improve information search and evaluation during design performance. On the other hand, it would seem less likely that a basic cognitive component to satisficing (e.g. in terms of selective information processing caused by working memory limitations) would be readily open to correction via education.

In concluding this section it is worth briefly mentioning something about the potential that training software might have for educating design skill and improving its effectiveness. Whilst the author is not directly aware of the existence of any computer-based training aids for design it is clearly unlikely that the development of such systems has not been undertaken and it would clearly be interesting to look at the facilities that such systems offer. From the perspective of the present research, however, one important facility that such training software could provide is the capability for the engineer to be taken interactively through the design of a device in a range of possible ways (e.g. breadth-first, depth-first, via alternative solution concepts etc.) such that the relative value and effectiveness of such different design approaches can be readily seen. In this way, for example, the strength of a breadth-first design approach could be contrasted on-line with examples of difficulties (e.g. design conflicts across levels of the design hierarchy) that could ensue from pursuing a depth-first approach. An interesting suggestion in this latter regard might be to provide an on-line display of the "problem agenda" underlying different design strategies. Rendering the problem agenda explicit in this way might well give the designer a better intuitive grasp for the nature of different design approaches - especially if the problem agenda is a

genuine psychological construct unerlying design planning. Many other possibilities for computer-based training in design can be thought of including educating engineers in ways of effective problem definition and problem structuring and the pursual of design solutions that accurately meet given design specifications.

A further role for the computer in design domains relates to the aiding rather than the training desgn processes, and this theme is given some detailed consideration in the following section.

6.3.3. IMPLICATIONS FOR COMPUTER AIDED DESIGN

One important consequence of obtaining a theoretical understanding of cognitive processes in engineering design relates to the implications of such an understanding for the development of CAD (Computer Aided Design) systems. As was noted in the general introduction to this thesis, there are many approaches to aiding engineering design through the employment of computer-based work environments, and the resulting systems can be viewed as varying along a multiplicity of dimensions relating, for example, to their co-operativeness, their embedded intelligence, their generality, the level of design abstraction they support and so on. One aspect that these many different CAD systems have in common, however, is their general lack of any underlying ***psychological*** foundation. As a case in point consider one contemporary approach to design aiding commonly known as "design automation". This approach lies at the centre of most existing applications of intelligent systems to CAD and involves the development of "expert systems" by orthodox methods of knowledge engineering. The object of knowledge engineering is to acquire knowledge in a form that can be implemented in a computer program and hence the

focus of such methods is on eliciting the detailed *content* of expert knowledge in the form of domain-specific facts, rules and heuristics. In this way, then, orthodox knowledge engineering approaches to CAD (1) pay scant attention to the cognitive processes that are involved in design (2) overlook organisational differences between the representation of knowledge in humans and machines and (3) ignore considerable psychological evidence for tendencies toward bias and error in expert thought.

As an alternative to most existing CAD approaches, the Plymouth Engineers Design Assistant (PEDA) being developed her at Polytechnic South West was intended to be an "interactive" system that was firmly founded on a psychological understanding of the mental processes involved in electronics design. This psychological foundation was felt to be essential for the PEDA to be able to work with the engineer in a genuinely cooperative manner, facilitating his or her thought processes and helping to optimise the designs produced. Clearly, then, the PEDA was in no way aimed at replacing the engineer's design activity in the manner favoured by exponents of expert systems technology. Indeed, the expert system approach to CAD was felt by the PEDA team to be unrealistically ambitious in that even for narrow sub-domains of design the amount of technical knowledge that needs to be elicited tends to be both complex and vast. This latter factor perhaps goes some way toward explaining why most automated design systems have remained at a research stage rather than having been developed into commercial products.

Before proceeding to consider how the present research may have some some bearing upon the development of interactive CAD systems such as the PEDA, it is initially important to emphasise that the implications

stemming from the psychological studies are best viewed in a somewhat speculative vein because of the tentative nature of the findings upon which they are based. In accordance with this proviso, then, the prototype PEDA - whose development was informed by the psychological work - should also be viewed as a highly experimental system whose facilitative effect on engineering design processes remains to be evaluated by further psychological research.

With these foregoing caveats in mind, then, the major implication of the present findings for CAD would seem to centre on the issue of sub-optimal performance in electronics design - as typified, for example, by the satisficing approach which was consistently seen to be adopted by the engineers studied. In particular, the proposal being put forward here is that any interactive design system should be concerned with providing facilities that help counteract such tendencies toward sub-optimal problem solving. Clearly, however, there are various important trade-offs to consider as far as the optimisation of design performance is concerned. Paramount amongst these is perhaps the fact that in today's profit-oriented design climate the requirement for fast lead times on design solutions may well dictate that a satisficing approach to design <u>must</u> be adopted as part of a realistic design strategy. Clearly, however, any design system that (a) facilitates the efficient pursual and evaluation of multiple design ideas, which at the same time (b) maintains - or even reduces - the customary time-scale for attainment of a design, is a CAD system to be welcomed.

Going on a step further from the preceding general ideas about the need to optimise design performance it seems possible to formulate five specific recommendations for the development of design support systems.

That is, such systems should be developed so as to:

(1) encourage the designer to consider an increased number of initial high-level solution concepts and enable the efficient formulation of alternative versions of each solution concept through levels of increasing design detail,

(2) assist with the choice of competing design solutions, for example, enabling evaluations of solutions to be made on the basis of comparative functional simulations,

(3) superintend the designer's exploratory activity, for example, helping the designer to backtrack if a path proves unpromising (i.e. by providing a record of paths taken together with the current point of exploration) or suggesting worthwhile paths of investigation (i.e. by suggesting design alternatives),

(4) ensure the designer's awareness of design conflicts (e.g. if crucially important constraint requirements have been overlooked when the designer is focusing on a narrow aspect of the overall design solution) and

(5) ensure the designer's awareness of inconsistencies in the notation that is being used (e.g. if two different design parameters have been given the same symbolic label).

Clearly the latter set of recommendations for CAD are all geared toward assisting the engineer with the search for alternative and potentially optimal design solutions as well as toward reducing the time of the design process. As a backdrop to the set of recommendations stated above, however, it additionally seems eminently sensible to develop

design systems which not only improve the efficiency and quality of design performance but also enable engineers to work in as *natural* a manner as possible without placing "unnecessary" constraints on their preferred strategies or methods of working. In this latter regard, for example, it would seem reasonable to permit engineers to pursue both depth-first and breadth-first designs rather than imposing restrictions on the way that they traverse the design space. One important argument to support the maintenance of a natural design style relates to the idea that designer's skill has developed with experience so as generally to be very effective whilst still being prone to the weakening effects of biased or over-selective information processing. The point is, then, that whilst it is worthwhile countering bias and error in design it is still well worth preserving those features of engineer's design strategies which are largely successful. This principle of allowing for a natural style of working also has the advantage of being useful when it comes to the acceptance of a system by potential users for it is certain that any attempt to enforce adherence to rigidly formalised and inflexible methods of working would be met with apprehension by designers. Related to the issue of maintaining the naturalness of design style is also the issue of who has the control over design decision making when choices need to be made (i.e designer or system?). Again the proposal in this regard is that whilst the system may recommend design alternatives and fruitful paths of exploration to the designer, the ultimate control as to what is accepted or rejected should remain with the designer.

The realisation of the full set of recommendations outlined above within an engineering CAD system certainly has an air of being a difficult

design task in itself. Indeed, it is to the credit of the engineering and computing members of the PEDA research team that a prototype system has been implemented that embodies the majority of these recommendations. The PEDA itself has been implemented within an object-oriented framework provided by the Sun/ART development environment. Whilst a summary of the functionality offered by the PEDA system is presented below, the reader is directed to Baker et al. (to appear) and Scothern (1990) for rather more detailed reports on the current state of the system.

In brief, then, the designer using the PEDA is able to interact with the system via a mouse-driven, direct manipulation interface which offers a screen-based drawing board. Design solutions taking the form of block diagrams can be constructed at any level of design detail, with blocks being selected from a palette and representing both mathematical and electronic functions as well as specific components. The PEDA supports the hierarchical nature of design by permitting solutions to be pursued at several levels of design detail and allowing the user to zoom in on high level blocks so as to manipulate their constituent functions or components. Optimal rather than just satisfactory designs are encouraged by facilities which permit the modelling, evaluation (through simulation) and storage of a multiplicity of alternative design solutions in parallel. The PEDA also has the ability to monitor the development of design alternatives and to provide advice by, for example (1) giving reminders of incomplete activities (2) maintaining a record of all crucial design requirements and constraints and (3) advising when a particular line of solution development has already been explored previously. Knowledge-based functions have been included in the PEDA

which enable the validation and optimisation of designs, i.e. by means of a data-base of component attributes working in conjunction with a rule-base of electronic design heuristics.

From the brief account that has been presented of the current prototype PEDA it is clear that the system typifies a highly non-interventionist approach to improving design performance via a computer-based support environment. Whilst this approach has been a major feature of the general ethos of the PEDA research team, this is not to say that the possibility of incorporating some kind of strategic intervention within future versions of the PEDA has been totally ruled out. Indeed there are certain reasons to believe that the imposition of some form of structured design approach could have substantial benefits to engineers - though clearly acceptance issues are a major problem in this regard.

The major reason behind the belief that structuring designer's activities might enhance the design process is based on evidence deriving from research within the field of "decision analysis" - an activity concerned with helping decision makers systematically to structure their problems in a way which enables the formulation of a solution based on normative decision theory. The claim of particular relevance here (e.g. Humphreys & McFadden, 1980; Berkeley & Humphreys, 1982; Von Winterfeldt & Edwards, 1986) is that the actual "problem structuring" appears to provide the biggest benefit for improved decision making and that the computation of an optimal choice based on decision-theoretic principles is not of particularly major importance. Evans (1989) has suggested that problem structuring may have its beneficial effects since (a) the method enables people to develop effective mental models of their problems and (b) the externalisation of

these mental models as diagrams (e.g. decision trees) can compensate for difficulties arising through working memory limitations.

As far as future developments of the PEDA were concerned, then, it was decided that it would be fruitful to pursue some psychological investigation of the effects of strategic intervention on design and the following section reports on a pilot study that was undertaken in this regard. Clearly any types of intervention that could be shown to be effective for improving design performance might profitably be embodied within later versions of the PEDA system.

6.4. INTERVENTION APPROACHES TO IMPROVING DESIGN: A PILOT STUDY

6.4.1. AIMS OF THE STUDY

As outlined above the pilot study to be reported here was specifically concerned with investigating the viability of improving electronics design by means of some kind of direct intervention during the course of ongoing activity. Particularly appealing was the idea of trying to increase a subject's investigation and modelling of design alternatives - with the possible consequence of a larger number of optimal design concepts being generated and maintained. However, rather than investigating the benefits of a full-blown problem structuring approach to engineering design (e.g. as used in the field of decision analysis), it was felt to be more appropriate to look - at least initially - at the benefits of more subtle and unobtrusive forms of problem structuring. Would, for example, positive effects on design quality arise by periodically requesting engineers working at a design task simply to

provide structured, verbal reports about specific types of critical design-oriented information (e.g. recent design decisions that had been made or current goals that were being pursed)?

Certainly there are major theoretical reasons to believe that an imposed requirement to provide verbal reports of selected types of information during task-directed thinking might have *some* sort of influence on design processes (though clearly such influence need not necessarily be positive). In particular, the theoretical proposals of Ericcson and Simon (1980, 1984) concerning the nature of the cognitive processes underlying the production of verbal reports are especially relevant in this regard (see section 3.1.2 for a synopsis of these authors' ideas). It may be recalled that Ericcson and Simon present a "three-level" characterisation of verbalisation where level 1 involves direct articulation of verbally coded information, level 2 involves articulation of what was originally non-verbal information and level 3 involves verbalisation of either selected types of heeded information or information that would not usually be attended to during task performance. The main proposal of importance here is that level 3 verbalisations appear to be the most likely to change the actual nature of subjects' task-oriented processing. Clearly the verbalisation requirement being proposed within the present study (i.e. periodic verbal reports of selected design-related information) falls within the level 3 category and would thus be predicted to have some influence on subjects' design processes.

Methodologically, then, the present study was intended to take the form of experimental investigation involving quantitative research methods as opposed to the predominantly descriptive and qualitative analyses of

individual design behaviour undertaken in the previous studies. The experiment was also intended to look at design problem solving over shorter design scales than previously examined since it was felt that the use of a small-scale task (i.e. requiring about one hour of work) would be advantageous for maintaining the continued motivation of participants whilst also reducing the complexity of the data analyses phase of the research.

Three conditions were included in the experiment which manipulated the specific kinds of information that subjects had to externalise when prompted. In the "goals" condition subjects were required to write down the goals that they were currently striving toward whilst in the "rationales" condition subjects were required to write down the reasons underlying their selection of recent technical design options. The third condition required that subjects simply list any components that they had recently used in their design work. This "components" condition was incorporated in the experiment in an attempt to provide a baseline against which to assess the effectiveness of the other intervention conditions, i.e. it was predicted to be the least likely to have a reactive effect on normal design processes. On the other hand, it was predicted that the goals and rationales interventions would have not only a reactive effect on design processes, but also one that would enhance design quality since subjects' attention would be re-focussed on crucially important design information. For example, the articulation of goal-oriented information might be expected to bring subjects to an awareness of what they were actually striving to do in order to attain the given design specification, whilst the articulation of the rationales underlying design decisions might be expected to alert

subjects to alternative design possibilities (particularly if they failed to rationalise satisfactorily any design decision).

6.4.2. METHOD

Fifty-five male and female undergraduate electronic engineers from the third-year of their degree course served as subjects in the study. These subjects were all tested at a single sitting and were each awarded five pounds for their voluntary participation. Subjects were permitted one hour to try and solve the given design problem which had been devised so as to be within their range of ability and technical understanding whilst also being non-trivial (refer to appendix C for the problem specification). Design work was undertaken using pen/pencil and paper (booklets of blank sheets being provided by the experimenter), and whilst the use of erasers was prohibited subjects were permitted to use calculators.

All participants were required to stop their design work at ten minute intervals (a point that was signalled by the sounding of a bell) and spend the next two minutes writting down certain types of information on pre-formatted sheets of paper which were then collected up by the experimenter. The experiment involved three between-subjects conditions (refer to appendix C for the full set of written instructions) and depending on the condition to which subjects had been assigned, the instructions requested the following information to be externalised:

GOALS CONDITION (19 subjects) - "Write down the goals that you are currently aware of pursuing which relate to your design work".

RATIONALES CONDITION (18 subjects) - "Write down the rationales underlying your recent decisions to keep or reject technical design options".

COMPONENTS CONDITION (18 subjects) - "Write down the names of the components that you have recently included in your design and list two technical attributes of each".

6.4.3. RESULTS AND DISCUSSION

Before looking at the effects of the experimental manipulations on design quality it is first worth presenting the results of a preliminary analysis of subjects' design scripts that focussed on rather gross aspects of subjects' design work - in particular (a) the quantity of text (e.g. words and abbreviations) written (b) the quantity of numerical notation (e.g. mathematical operators and operands) used and (c) the quantity of diagrams (e.g. block diagrams and flow charts) produced. Carefully formulated *a priori* criteria were devised such that each item within a subject's design script could be consistently placed into one of these three categories. Table 6.1 presents the mean number of items per notation category as a function of experimental condition.

Further statistical analyses undertaken on the quantities of each type of notation present within design scripts only revealed a significant effect of experimental condition on the amount of text subjects produced ($p < 0.01$, $df = 2$, Kruskal-Wallis). Perusal of table 6.1 suggests that this latter result reflects an increased use of textual notation by subjects within the goals and rationales conditions.

Intervention condition	Category of notation		
	textual	numeric	diagrammatic
Goals	325.79 (150.12)	29.74 (32.32)	5.16 (2.39)
Rationales	275.33 (144.52)	29.11 (18.39)	4.22 (2.60)
Components	203.67 (135.53)	25.33 (18.50)	5.50 (2.90)

Table 6.1: Mean number of items per notation category as a function of experimental condition (with standard deviations in parentheses)

The result is clearly encouraging as it reveals that the goals and rationales interventions were having some kind of reactive effect on subjects design work. A plausible interpretation of this effect is that subjects were more inclined to present documentary accounts within their design of what they were trying to do (goals condition) or what they had done (rationales condition) as a result of the interventions. Clearly this effect has possible implications for improving designers' documentation of their design work (anecdotal evidence indicates that design documentation is something that engineers are notoriously bad at doing).

Moving on , however, to consider whether improvements in actual design quality were afforded by the manipulations in the present experiment it was clearly necessary first of all to devise some measurement instrument (i.e. a set of rating scales) that could be used to evaluate subjects' design work. At a general level, then, it was thought essential to assess design quality both in terms of (a) the underlying processes and strategies that subjects appeared to have used in their design work as

well as (b) the actual end products (i.e. the final design models) that resulted from the application of these processes and strategies. To these ends, then, a design assessment scheme was formulated which comprised fifteen questions (refer to appendix C). Eleven of these questions (labelled A to K) related to the quality of a subject's design procedures whilst the remaining four (labelled L to O) related to the completeness, accuracy and optimality of a subject's final design solution. Each question was associated with a five-point rating scale which permitted a range of quality assessments to be given within the range of very good (scored as a 5) to very poor (scored as a 1).

Two experienced electronic engineers (both members of the PEDA research team) who were highly familiar with evaluating undergraduate design work were recruited to assess subjects' design scripts. These assessors (henceforth referred to as assessor 1 and assessor 2) were provided with a detailed written briefing (see appendix C) on the meaning of the terms contained within the fifteen questions. This was done in an attempt to ensure a high level of consistency in the way that the assessors interpreted the questions when scoring subjects' design work. The two assessors carried out their design ratings independently and were blind as to each subject's experimental condition.

Tables 6.2 and 6.3 summarise the individual ratings that were made by the two assessors for each question as a function of intervention condition. To facilitate the perusal of these mean ratings, the highest score in each column has been underlined in order to provide a guide as to which intervention tended to produce the greatest improvement in design quality. It is clear that there are trends toward (a) the greatest improvements in terms of the quality of design processes

Intervention Condition	Question														
	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)	(I)	(J)	(K)	(L)	(M)	(N)	(O)
Goals	3.26	2.68	1.63	1.16	1.47	1.26	4.05	3.47	2.74	2.47	3.00	2.95	2.58	2.63	2.58
Rationales	3.44	2.67	1.50	0.89	1.39	1.06	3.94	3.33	2.66	2.72	2.83	3.11	2.61	2.94	2.67
Components	3.94	2.28	1.67	0.94	1.38	1.11	3.72	2.94	2.67	2.10	2.67	3.06	2.17	2.39	2.38

Table 6.2: Mean ratings of assessor 1 for questions A to O as a function of intervention condition

Intervention Condition	Question														
	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)	(I)	(J)	(K)	(L)	(M)	(N)	(O)
Goals	2.42	2.63	1.89	1.16	1.47	1.84	3.21	3.16	3.32	2.89	1.37	3.00	2.63	2.58	2.11
Rationales	2.61	2.67	1.83	1.00	1.56	1.50	3.11	2.67	3.17	2.89	1.39	3.33	2.89	2.89	2.11
Components	2.39	2.28	2.22	1.78	1.78	1.56	2.78	2.89	2.89	2.83	1.61	3.11	2.50	2.78	2.06

Table 6.3: Mean ratings of assessor 1 for questions A to O as a function of intervention condition

(reflected particularly by both assessors' ratings for questions F to I) being afforded by the goals intervention and (b) the greatest improvements in terms of the quality of design products (reflected by ratings for questions L to O) being afforded by the rationales intervention. It is also clear, however, that many of the differences between means ratings across conditions are slight.

In order to pinpoint whether any real differences in design quality were present between conditions, a variety of inferential statistical tests were applied independently to each assessor's ratings data. In a first phase of testing, then, for each assessor's ratings Kruskal-Wallis analyses were undertaken on the raw scores pertaining to each of the fifteen assessment questions. None of the fifteen analyses per assessor showed up any statistically significant differences between conditions, though some probabilities approached the 0.05 level of significance (something that would be expected by chance with so many tests being undertaken).

A second phase of statistical testing involved pursuing a principal components analysis on each assessors ratings data in order to extract the major factors accounting for the variance in that assessors scoring. Subsequent to the extraction of factors (note that factors were maintained for further analysis only if they had eigenvalues greater than 1) an attempt was made to look for any significant differences between experimental conditions in terms of the factor loadings on subjects. As it transpired, for both assessor 1 and assessor 2 no significant differences were found in factor loadings between conditions when either a Kruskal-Wallis test or an analysis of variance test was applied.

Certainly the lack of any evidence for significant differences in design quality between intervention conditions was disappointing. These null results, however, point out the difficulties inherent in devising intervention and problem structuring manipulations that are capable of effecting design processes positively whilst at the same time being subtle enough to not frustrate the engineer in his or her preferred mode

of working. Indeed it may well prove over-optimistic to assume that unobtrusive intervention approaches can be used to improve an engineer's design work - though certainly there is considerable scope for more research along these lines. Clearly the other tack to take for optimising design within CAD environments is to attempt the imposition of a greater degree of formal method on the engineer - an option that has little appeal from the perspective of user acceptance or in relation to the maintenance of creative and flexible design style.

6.5. CONCLUDING COMMENTS

6.5.1. FUTURE WORK AND FINAL COMMENTS

An important aim of the foregoing final discussion has been to draw out and clarify findings concerning the nature of engineering design processes which stemmed from the two major studies undertaken within the present research programme. Similar to the approach taken throughout the thesis, the focus has been on the theoretical interpretation of design processes in the context of general psychological theory on problem solving and thinking. Indeed it has been argued that much of the broader theoretical value of the present research resides in the contribution that it makes to existing knowledge concerning the the processes involved in tackling ill-defined, real-world problems. In this regard, one very general feature of the findings that is interesting is that many of the critical processes underlying problem solving in engineering design such as problem definition, problem reduction, simulation modelling and the like appear to be similar in type to those that underly activity in other problem-solving domains that revolve around

fairly well-defined problems. Whether, of course, the findings that derived from the studies undertaken here are generalisable to problem solving with ill-defined tasks in real-world domains apart from design (e.g. fault diagnosis or experimental research) or indeed to engineering design domains other than electronics, whilst likely, yet remains to be seen.

At a more specific level, the previous discussion of findings has dealt with evidence that electronics designers display many common strategies and methods of working. In this regard an important proposal has been that the high-level knowledge that engineers possess about how to design may usefully be described in terms of a general purpose "design schema" that hinges on a basic problem reduction strategy. Another important contemporary concept which has been repeatedly used in the interpretation of the present design data has been the "mental models" notion. Mental modelling has been argued to underly problem solving in enginering design in a number of conceptually distinct ways, sometimes with the emphasis being placed on mental models as representational constructs and other times as cognitive processes (e.g. in the simulation sense). Both the schema notion and the mental models notion have clearly proved invaluable in the context of the present research in that they have acted as heuristics to suggest both (a) useful questions to ask in exploring the nature of design activity as well as (b) helpful ways of understanding the data collected. In a similar manner, general theoretical evidence in the psychological literature on thinking and problem solving which implicates the existence of bias and error in cognition suggested that some kinds of similar tendencies are likely to be present in design settings. In this regard then, evidence for

suboptimal performance in design - as reflected by tendencies toward satisficing and notational inconsistency - has come as little surprise in the present research.

Whilst evidence for commonalities in engineers' design problem solving (i.e. in terms of the methods used as well as tendencies toward suboptimal performance) seems persuasive there is clearly a need to test the generality of such evidence - both within the electronics domain as well as within other fields of engineering design such as mechanical or civil engineering. In testing the generality of, for example, the design schema and satisficing notions, it would seem essential to investigate both (a) subjects having expertise levels different to those studied in the present context and (b) subjects tackling design problems having characteristics different to those undertaken in the present studies. Certainly there is an important role here for the controlled experimental comparison of design performance across expertise levels, problem types and the like, as opposed to the highly individualistic case studies that were undertaken in the present research programme. The experimental method would clearly be an exciting one to pursue and many interesting manipulations spring to mind that might be worthwhile investigating. For example, it would be interesting to see if the use of a satisficing rather than an optimising strategy is universally adopted in electronics design or if it is influenced by variables such as problem complexity or time pressure. It would also be valuable to pursue a controlled comparison of the effects of experience on the development of design schemas - particularly to assess whether the depth-first/breadth-first distinction maps on to differences in experience levels. Whilst, then, it would be appealing to see the use of

experimental methods in research on engineering design there is clearly an important role still to be played by the more qualitative types of research techniques that were employed in the present exploratory work. For example, one potentially valuable study in this regard would involve the longitudinal investigating of professional company engineers tackling design problems for real-world applications. In this latter respect it would also be useful to assess the applicability of the design behaviour graph technique for encoding and structuring the design data obtained in such a study.

In addition to work that would be worthwhile undertaking to assess the theoretical conclusions concerning engineering design processes that have emerged from the present studies, there is also considerable scope for research aimed at determining the effectiveness of intervention strategies on design performance. The assessment of whether various problem structuring approaches are helpful in engineering design situations would seem a particularly important avenue of research. For example, providing engineers with the facilities to produce overt representations of design options (e.g. as multi-attribute evaluation tables and decision trees) might have some positive effect on design quality - though it seems that a major problem which needs to be addressed here is how to get engineers to consider an increased number of design alternatives in the first place. As was seen in the small-scale intervention study that was undertaken in the present research programme, the use of relatively subtle interventions to optimise design processes are unlikely to afford much in the way of changes to subjects' normal processing. An alternative possibility, which involves the use of more powerful kinds of intervention, whilst much more likely to produce

alterations in design performance, is also likely to be met with apprehension by design engineers. Considerable research is clearly needed here to determine the best path for optimising design via intervention and the imposition of formal method.

This issue of intervention in design is certainly an important one as any computer-based design aid will inevitably impose some kind of structure and organisation on the human designer. The minimal interventionist approach that was adopted in the development of the Plymouth Engineer's Design Assistant, whilst attractive from a user acceptance point of view, still has to be evaluated as being of any genuine benefit to the electronics designer. Certainly there is an important role for the apllied psychologist in the actual evaluation of software tools - though this is something that was not undertaken within the present research programme. In addition it would be appealing to see applied cognitive psychologists involved in the development of computer-based training environments, and this is certainly a worthwhile avenue for further multi-disciplinary research of the kind that was undertaken in the present overriding research effort.

As a final point, it is worth stating that the investigation of the relatively unexplored domain of electronics design has proved to be both a challenging and rewarding endeavour for the present researcher and one that has been considerably facilitated by the interaction with colleagues from the PEDA research team who actually stem from the engineering discipline. It is hoped that the challenge of studying complex, real-world domains such as design will be taken up by other psychologists interested in applying their theoretical knowledge to the study of real-world problem solving.

References

Adelson, B. & Soloway, E. (1986). A model of software design. International Journal of Intelligent Systems, 1, 195-213.

Alba, J.W. & Hasher, L., (1983) Is memory schematic? Psychological Bulletin, 93, 203-231.

Anderson, J.R. (Ed.). (1981) Cognitive skills and their acquisition. Hillsdale, NJ.: Lawrence Erlbaum Associates.

Anderson, J.R. (1983). The architecture of cognition. Cambridge. Mass.: Harvard University Press.

Anderson, J.R. (1985). Cognitive psychology and its implications (Second edition). New York: Freeman.

Anderson, J.R, Farrell, R. & Sauers, R. (1984). Learning to program in LISP. Cognitive Science, 8, 87-129.

Asimow, M. (1962). Introduction to design. New York: Prentice-Hall.

Atwood, M.E. & Polson, P.G. (1976). A process model for water jug problems. Cognitive Psychology, 8, 191-216.

Baker, K.D., Ball, L.J., Culverhouse, P.F., Dennis, I., Evans, J.St.B.T., Jagodzinski, A.P., Pearce, P.D., Scothern, D.G.C. & Venner, G.M., (to appear). A psychologically based intelligent design aid. In P. Veerkamp (Ed.), Intelligent CAD systems 3: Practical experience and evaluation. Berlin: Springer-Verlag.

Bartlett, F.C. (1932). Remembering. Cambridge: Cambridge University Press.

Berkeley, D. & Humphreys, P. (1982). Structuring decision problems and the bias heuristic. Acta Psychologica, 50, 201-152.

Berry, D.C. & Broadbent, D.E. (1984). On the relationship between task performance and associated verbalisable knowledge. Quarterly Journal of Experimental Psychology, 36A, 209-231.

Best, J.B. (1986). Cognitive Psychology. West Publishing Company.

Bhaskar, R. & Simon, H.A. (1977). Problem solving in semantically rich

domains: An example from engineering thermodynamics. Cognitive Science, 1, 193-215.

Borgman, C.L. (1986). The user's mental model of an information retrieval system: An experiment on a prototype online catalog. International Journal of Man-Machine Studies, 24, 47-64.

Bowen, J.A. (1985). A search-oriented model of the design process. Technical Report, Department of Computer Science, Reading University.

Burton, A.M. & Schadbolt, N.R. (1988). Knowledge engineering. In N. Williams & P. Holt (Eds.), Expert systems for users. New York: McGraw-Hill.

Byrne, R. (1977). Planning meals: Problem-solving on a real data base. Cognition, 5, 287-332.

Byrne, R. (1981). Mental cookery: An illustration of fact retrieval from plans. Quarterly Journal of Experimental Psychology, 33A, 31-37.

Byrne, R. (1983). Protocol analysis in problem solving. In J.St.B.T. Evans (Ed.), Thinking and reasoning: Psychological approaches. London: Routledge & Kegan Paul.

Charness, N. (1979). Components of skill in bridge. Canadian Journal of Psychology, 33, 1-16.

Chase, W.G. & Simon, H.A. (1973a). Perception in chess. Cognitive Psychology, 4, 55-81.

Chase, W.G. & Simon, H.A. (1973b). The mind's eye in chess. In W.G. Chase (Ed.), Visual information processing. New York: Academic Press.

Cheng, P.W. & Holyoak K.J. (1985). Pragmatic reasoning schemas. Cognitive Psychology, 17, 391-416.

Cheng, P.W. & Holyoak K.J.,Nisbitt, R.E., & Oliver, L.M. (1986) Pragmatic versus syntactic approaches to training deductive reasoning Cognitive Psychology, 18, 293-328.

Chi, M.T.H., Feltovich P.J. & Glaser, R. (1981). CAtegorization and representation of physics problems by experts and novices. Cognitive Science, 5, 121-152.

Chi, M.T.H. & Glaser, R. (1985). Problem-solving ability. In R.J. Sternberg (Ed.), Human abilities: An information-processing approach. New York: Freeman.

Chi, M.T.H & Glaser, R. & Rees, E. (1982). Expertise in problem solving. In R.J. Sternberg (Ed.), Advances in the psychology of human intelligence, Volume 4, Hillsdale, NJ.: Lawrence Erlbaum Associates.

Cooke, N.M. & McDonald, J.E. (1986). A formal methodology for acquiring and representing expert knowledge. Proceedings of the IEEE, 74, 1422-1430.

Claxton, G. (1978). Special review feature: Memory research. British Journal of Psychology, 69, 513-520.

Craik, K.J.W. (1943). The nature of explanation Cambridge: Cambridge University Press.

Culverhouse, P.F. (1986). Design tools for engineers. Technical Report, Department of Electrical & Electronic Engineering, Plymouth Polytechnic.

David, B.T. (1987). Multi-expert systems for CAD. In P.J.W. ten Hagen & T. Tomiyama (Eds.), Intelligent CAD systems 1: Theoretical and methodological aspects. Berlin: Springer-Verlag.

de Groot, A.D. (1965). Thought and Choice in Chess. The Hague: Mouton.

de Groot, A.D. (1966). Perception and memory versus thought: Some old ideas and recent findings. In B. Kleinmuntz (Ed.), Problem solving: Research, method, and theory. New York: Wiley.

de Kleer, J. & Brown, J.S. (1983). Assumptions and ambiguities in mechanistic mental models. In D. Gentner & A.L. Stevens (Eds.), Mental models. Hillsdale, NJ.: Lawrence Erlbaum Associates.

de Kleer, J. (1984). How circuits work. Artificial Intelligence, 24, 205-280.

Director, S.W., Shen, J.P., Siewiorek, D.P. & Thomas, D.E. (1982). The CMU DA/CAD project.

Duncker, K. (1945). On problem solving. Psychological Monographs, 58, (270), 1-113.

Eastman, C.M. (1968). On the analysis of intuitive design processes. In G.T. Moore (Ed.), Emerging methods in environmental design and planning. Cambridge, Mass.: MIT Press.

Eastman, C.M. (1969a). Cognitive processes and ill-defined problems: A case study from design. Proceedings of the 1st Joint International Conference on Artificial Intelligence. 19-31.

Eastman, C.M. (1969b). Problem solving strategies in design. Proceedings of the 1st Annual Environmental Design Research Conference. 138-152.

Engle, R.W. & Bukstel, L. (1978). Memory processes among bridge players of differing expertise. American Journal of Psychology, 91, 673-689.

Ericsson, K.A. & Simon, H.A. (1980). Verbal reports as data. Psychological Review, 87, 215-251.

Ericsson, K.A. (1975). Instruction to verbalize as a means to study problem solving processes with the 8-puzzle: A preliminary study. Report No.458, Department of Psychology, The University of Stockholm.

Ericsson, K.A. & Simon, H.A. (1980). Verbal reports as data. Psychological Review, 87, 215-251

Ericsson, K.A. & Simon, H.A. (1984). Protocol analysis: Verbal reports as data. Cambridge, Mass.: MIT Press.

Evans, J.St.B.T. (1988). The knowledge elicitation problem: A psychological perspective. Behaviour and Information Technology, 7, 111-130.

Evans, J.St.B.T. (1989). Bias in human reasoning: Causes and consequences. Brighton: Lawrence Erlbaum Associates.

Fielden, G.B.R. (1963). Engineering design. London: Her Majesty's Stationery Office.

Fong, G.T., Krantz, D.H. & Nisbett, R.E. (1986). The effects of statistical training on thinking about everyday problems.

Cognitive Psychology, 18, 253-292.

Gammack, J.G. & Young, R.M. (1984). Psychological techniques for eliciting expert knowledge. In M.A. Bramer (Ed.), Research and development in expert systems. Cambridge: Cambridge University Press.

Gick, M.L. & Holyoak K.J. (1980). Analogical problem solving. Cognitive Psychology, 12, 306-355.

Gick, M.L. & Holyoak K.J. (1983). Schema induction and analogical transfer. Cognitive Psychology, 15, 1-38.

Gilhooly, K.J. (1987). Mental modelling: A framework for the study of thinking. In D.N. Perkins, J. Lochhead & J.C. Bishop (Eds), Thinking, Hillsdale, N.J.: Lawrence Erlbaum Associates.

Gilhooly, K.J. (1988). Thinking: Directed, undirected and creative (Second edition). London: Academic Press.

Hayes, J.R. (1978). Cognitive psychology: Thinking and creating. The Dorsey Press.

Holyoak, K.J. & Koh, K. (in press). Surface and structural similarity in analogical transfer. Memory and Cognition.

Holland, J.H., Holyoak, K.J., Nisbett, R.E. & Thagard, P.R. (1986). Induction: Processes of inference, learning and discovery. Cambridge, Mass.: MIT Press.

Hubka, V. (1982). Principles of engineering design. London: Butterworths.

Humphreys, P.C. & McFadden, W. (1980). Experiences with MAUD: Aiding decision structuring versus bootstrapping the decision maker. Acta Psychologica, 45, 51-69.

Jacob, V.S., Gaultney, L.D. & Salvendy, G. (1986). Strategies and biases in human decision-making and their implications for expert systems. Behaviour and Information Technology 5, 119-140.

Jeffries, R., Turner, A.A., Polson, P.G. & Atwood, M.E. (1981). The processes involved in designing software. In J.R. Anderson (Ed.),

Cognitive skills and their acquisition. Hillsdale, NJ.: Lawrence Erlbaum Associates.

Johnson-Laird, P.N. (1983). Mental models. Cambridge: Cambridge University Press.

Johnson-Laird, P.N. & Bara, B.G. (1984). Syllogistic inference. Cognition,16, 1-62.

Jones, J.C. (1980). Design methods: Seeds of human futures. (Second edition). New York: Wiley.

Kahneman, D. & Tversky, A. (1982). The simulation heuristic. In D. Kahneman, P. Slovic & A. Tversky (Eds.), Judgement under uncertainty: Heuristics and biases. Cambridge: Cambridge University Press.

Kahney, H. (1987). Problem solving: A cognitive approach. Milton Keynes: Open University Press.

Kaplan, B. (1968). Comment: On the rational reconstruction of intuition in the design process. In G.T. Moore (Ed.), Emerging methods in environmental design and planning. Cambridge, Mass.: MIT Press.

Keane, M. (1988). Analogical problem solving. Chichester: Ellis Horwood.

Kidd, A. & Wellbank, M. (1984). Knowledge acquisition. In J. Fox (Ed.), Infotech state of the art report on expert systems. Maidenhead: Pergammon Press.

Kosslyn, S.M. (1980). Image and mind. Cambridge, Mass.: Harvard University Press.

Kowalski, T.J. & Thomas, D.E. (1983). The VLSI design automation assistant: First steps. Twentysixth IEEE Computer Society International Conference, 126-130.

Larkin, J.H. (1978). Problem solving in physics: Structure, process and learning. In J.M. Scandura & C.J. Brainerd (Eds.), Structural/process models of complex human behaviour. The Netherlands: Sijthoff and Noordhoff.

Larkin, J.H. (1979). Information processing models and science instructions. In J. Lochhead & J. Clement (Eds.), Cognitive process instructions. Philadelphia, Pa.: Franklin Institute Press.

Larkin, J.H. (1981). Enriching formal knowledge: A model for learning to solve problems in physics. In J.R. Anderson (Ed.), Cognitive skills and their acquisition. Hillsdale, NJ: Lawrence Erlbaum Associates.

Larkin, J.H., McDermott, J., Simon, D.P. & Simon, H.A. (1980). Models of competence in solving physics problems. Cognitive Science, 4, 317-345.

Luger, G.F. (1976). The use of state-space to record the behavioural effects of sub-problems and symmetries on the Tower of Hanoi problem. International Journal of Man-Machine Studies, 8 411-421.

Malhotra, A., Thomas, J.C., Carroll, J.M. & Miller, L.A. (1980). Cognitive processes in design. International Journal of Man-Machine Studies, 12, 119-140.

McKeithen, K.B., Reitman, J.S., Reuter H.H. & Hirtle, S.C. (1981). Knowledge organisation and skill differences in computer programmers, Cognitive Psychology, 13, 307-325.

Montgomery, H. & Svenson, O. (1976). On decision rules and information processing strategies for choices among multiattribute alternatives. Scandinavian Journal of Psychology, 17, 285-292.

Moran, T.P. (1968). A model of a multilingual designer. In G.T. Moore (Ed.), Emerging methods in environmental design and planning. Cambridge, Mass.: MIT Press.

Morris, P.E. (1981). Why Evans is wrong in criticising introspective reports of subject strategies. British Journal of Psychology, 72 465-468.

Mostow, J. (1985). Toward better models of the design process. AI magazine, 6, 44-57.

Newell, A. (1969). Heuristic programming: Ill structured problems. In J. Aronofsky (Ed.), Progress in Operations Research, Volume 3, New York: Wiley.

Newell, A. (1980). Reasoning, problem solving, and decision processes: The problem space as a fundamental category. In R. Nickerson (Ed.), Attention and performance, Volume 8, Hillsdale, NJ: Lawrence Erlbaum Associates.

Newell, A., Shaw, J.C., & Simon, H.A. (1958). Elements of a theory of human problem solving, Psychological Review, 65, 151-166.

Newell, A. & Simon, H.A. (1972). Human problem solving. Englewood Cliffs, NJ.: Prentice-Hall.

Nisbett, R.E. & Wilson, T.D. (1977). Telling more than we can know: Verbal reports on mental processes, Psychological Review, 84, 231-259.

Pahl, G. & Beitz, W. (1985). Engineering design. Berlin: Springer-Verlag.

Pitt, R.B. (1983). Development of a general problem-solving schema in adolescence and early adulthood. Journal of Experimental Psychology; General, 112. 547-584.

Popplestone, R., Smithers, T., Corney, J., Koutsou, A., Millington, K. & Sahar, G. (1986). Engineering design support systems. Proceedings of the 1st International Conference on Applications of Artificial Intelligence to Engineering Problems.

Pylyshyn, Z.W. (1973). What the mind's eye tells the mind's brain: A critique of mental imagery. Psychological Bulletin, 80, 1-24.

Pylyshyn, Z.W. (1981). The imagery debate: Analogue media versus tacit knowledge. Psychological Review, 88, 16-45.

Rasmussen, J. & Jensen, A. (1974). Mental procedures in real-life tasks: A case study of electronic trouble shooting. Ergonomics, 17, 293-307.

Reed, S.K., Ernst, G.W. & Banerji, R. (1974). The role of analogy in transfer between similar problem states. Cognitive Psychology, 6, 436-450

Reed, S.K. & Simon, H.A. (1976). Modelling strategy shifts in a problem solving task. Cognitive Psychology, 8, 86-97.

Reitman, W.R. (1964). Heuristic decision procedures, open constraints,

and the structure of ill-defined problems. In M.W. Shelleey & G.L. Bryan (Eds.), Human judgements and optimality. New York: Wiley.

Reitman, W.R. (1965). Cognition and thought. New York: Wiley.

Reitman, J. (1976).
Skilled perception in Go: Deducing memory structures from inter-response times. Cognitive Psychology, 8, 336-356.

Rips, L.J. (1986). Mental muddles. In M. Brand & R.M. Harnish (Eds.), Problems in the representation of knowledge and belief. Tuscon: University of Arizona Press.

Rips, L.J. (1989). The psychology of knights and knaves. Cognition, 31 85-116.

Rowe, H.A.H. (1985). Problem solving and intelligence. Hillsdale, NJ.: Lawrence Erlbaum Associates.

Rumelhart, D.E. (1980). Schemata: The building blocks of cognition. In R.J. Spiro, B.C. Bruce & W.F. Brewer (Eds.), Theoretical issues in reading comprehension. Hillsdale, NJ.: Lawrence Erlbaum Associates.

Schadbolt, N. & Burton, A.M. (1989). Knowledge elicitation. In J. Wilson & N. Corrlett (Eds.), Evaluation of human work: Practical ergonomics methodology Taylor & Francis.

Schneiderman, B. (1976). Exploratory experiments in programmer behavior. International Journal of Computer and Information Sciences, 5, 123-354.

Scothern, D.G.C. (1990). A knowledge-based support tool for the early stages of engineering design. Draft PhD thesis, Department of Electrical & Electronic Engineering, Polytechnic South West.

Simon, H.A. (1981). The sciences of the artificial (1st Edition). Cambridge, Mass.: MIT Press.

Simon, H.A. (1972). The heuristic compiler. In H.A. Simon & L. Siklossy (Eds.), Representation and meaning. Englewood Cliffs, NJ.: Prentice-Hall.

Simon, H.A. (1973). The structure of ill structured problems. Artificial Intelligence, 4, 181-201.

Simon, H.A. (1978a). Information processing theories of human problem solving. In W.K. Estes (Ed.), Handbook of learning and cognitive processes, Hillsdale, NJ.: Lawrence Erlbaum Associates.

Simon, H.A. (1981). The sciences of the artificial (2nd Edition). Cambridge, Mass.: MIT Press.

Simon, H. & Gilmartin, K. (1973). A simulation of memory for chess positions. Cognitive Psychology, 5, 29-46.

Smithers, T. (1985). The ALVEY large scale demonstrator project "design to product". Proceedings of the 3rd International Conference on Advanced Information Technology: Artificial Intelligence in Manufacturing - Key to Integration?

Smyth, M. (1988). Articulating the designer's mental codes. Technical Report, LUTCHI Research Centre.

Soloway, E.L. & Woolf, B. (1980).
From problems to problems via plans: The content and structure of knowledge for introductory LISP programming. Technical Report No. 80-19, University of Massachusetts, Department of Computer and Information Science.

Steinberg, L. (1987). Design as refinement plus constraint propagation: The VEXED experience. Proceedings of the American Association for Artificial Intelligence.

Sussman, G.J. & Steele, G.L. (1980). Constraints - A language for expressing almost-hierarchical descriptions. AI Journal, 14, 1-39

Thorndyke, P.W. (1984), Applications of schema theory in cognitive research. In R.C. Anderson & S.M. Kosslyn (Eds.), Tutorials in learning and memory. Hillsdale, NJ.: Lawrence Erlbaum Associates.

Ullman, D.G., Stauffer, L.A, & Dietterich, T.G. (1986). Preliminary results of an experimental study on the mechanical design process. Technical Report 86-30-9, Oregon State University, Corvallis, Oregon.

Ullman, D.G., Dietterich, T.G. & Stauffer, L.A. (1988). A model of the mechanical design process based on empirical data. Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AI EDAM), 2, 33-52.

Von Winterfeldt, D. & Edwards, W. (1986). Decision analysis and behavioural research. Cambridge.: Cambridge University Press.

Voss, J.F., Greene, T.R., Post, T.A. & Penner, B.C. (1983a). Problem solving skill in the social sciences. In G. Bower (Ed.), The psychology of learning and motivation: Advances in research and theory, Volume 17. London and San Diego: Academic Press.

Voss, J.F., Tyler, S.W. & Yengo, L.A. (1983b). Individual differences in the solving of social science problems. In R.F. Dillon & R.R. Schmeck (Eds.). Individual differences in cognition. New York: Academic Press.

Wason, P.C. & Evans, J.St.B.T. (1975). Dual processes in reasoning? Cognition, 3, 141-154.

Waterman, D.A. & Newell, A. (1971). Protocol analysis as a task for artificial intelligence. Artificial Intelligence, 2, 285-318.

Waterman, D.A. & Newell, A. (1973). PAS-II: An interactive task-free version of an automatic protocol analysis system. Proceedings of the 3rd International Joint Conference on Artificial Intelligence.

White, P.A. (1988). Knowing more about what we can tell: "Introspective access" and causal report accuracy 10 years later. British Journal of Psychology, 79, 13-45.

Wright, G. (1984). Behavioural decision theory. Harmondsworth: Penguin.