Human-Computer Interaction: From Classifying Users to Classifying Users' Misunderstandings

Paul Andrew Booth

Ph.D. Thesis submitted to the CNAA

(in partial fulfillment of the requirements of this degree)

Sponsoring Establishment: Huddersfield Polytechnic

Collaborating Establishment: International Computers Limited

Submitted: April, 1989, Resubmitted November 1989

Booth

Contents

Acknowledgements

I am indebted to my supervisors, Ken Eason, Chris Fowler, and Linda Macaulay for the way in which they have guided me during this research project. I also owe them a special thanks for their invaluable help with the most difficult phase of the project: coping with the unbelievable bureaucracy encountered within Huddersfield Polytechnic and the CNNA. In this last respect, I am also grateful to Professor Frank Sumner at Manchester University.

I would like to thank Phil Marsden, Adrian Castell, Deborah Twigger, and all of the people at the HCI Unit at Huddersfield Polytechnic, for their help and the very positive influence they have had both upon me and upon my work. I am grateful to Peggy Newton, Thomas Green, Erik Hollnagel and David Woods for their comments on some of the arguments and ideas within the thesis. Chris Marshall and Amanda Prail at Hewlett Packard were extremely helpful in allowing and contributing towards one of the studies within the project, and their help was greatly appreciated. Finally, I must thank my wife, Gill Brown, who has not only contributed to many of the ideas within this thesis, but has also been a constant source of support and inspiration.

Human-Computer Interaction: From Classifying Users to Classifying Users' Misunderstandings

Paul Andrew Booth

Abstract

The overall objective of the research has been to address the question of how best to understand user behaviour at the interface. The use of cognitive grammars to analyse tasks and predict behaviour was rejected for seven theoretical and practical reasons. Following this, cognitive style measures were rejected as a result the first study, where the visualizer-verbalizer and conceptual tempo cognitive style measures were not found to be accurate predictors of behaviour at a task. The results of this experiment indicated that interaction between a system and its user has certain dynamic qualities that make prediction of a fixed set of activities in a set order difficult. Furthermore, it seemed likely that behaviour is determined by a potentially complex interaction of variables rather than any single over-riding factor, such as a user's cognitive style.

Consequently, attention was focused upon the errors that occur during human-computer interaction. An approach where errors are classified was adopted, and a classification scheme was developed (ECM: an Evaluative Classification of Mismatch) as a vehicle for further research.

An initial pilot study showed that user-system errors could be classified using the scheme. This suggested that the concepts it employed did have some validity in both cognitive and computing domains. The second study of ECM involved a design team at Hewlett Packard's Office Products division in Wokingham. This study demonstrated that the classification scheme was usable by a design and development team that consisted of software engineers, human factors engineers, and technical authors. The third and final study of ECM demonstrated that it could be used to improve a design. A system, that had been changed using ECM, was shown to be significantly better, in terms of time, errors and user attitude ratings, than either its original or an iteration where ECM had not been employed.

This research has provided strong indications that evaluative classifications can be of use within the design and development process. Furthermore, this work emphasizes the importance of providing structures for thinking about the user's problems that are divorced from the structure and terminology of design.

Executive Summary

The overall objective of the research was to address the question of how best to consider user behaviour. As an initial step in the research, cognitive grammars were considered. These, however, were rejected as tools that might be of use in the design and development process. The seven reasons for this decision were as follows: These are that: it is difficult when using these methods to consistently employ the same grain of analysis across analyses, they assume consistent user behaviour, they may not expose actual mismatches between the user's and the designer's model of the task, they are not compatible with the design and development process, they require a clear view of the task to be modelled and this is not always available, they are too complex to use on an everyday basis, and many notations address only one aspect of interaction. Consequently, it is suggested that present cognitive grammars are best considered as research tools, and as the forerunners of more practical tools for the early stages of design, rather than as techniques that are likely to have an immediate impact upon design and development.

Following this, attention was focused upon the possible use of cognitive style measures to predict behaviour at a task. The visualizer-verbalizer and conceptual tempo cognitive style dimensions were the subject of an experiment. The results indicated that cognitive style, as measured along these two dimensions, did not significantly affect behaviour at the complex experimental task. In other words, cognitive style does not appear to be an over-riding factor that could be used to predict user behaviour. It was suggested that predicting detailed aspects of the user's behaviour was likely to prove problematic for two reasons. Firstly, interaction between a system and its user has certain dynamic qualities that make prediction of a fixed set of activities in a set order difficult. Secondly, it seems likely that behaviour is determined by a potentially complex interaction of variables rather than any single over-riding factor, such as a user's cognitive style.

Finally, returning to the question of how best to consider user behaviour, it was suggested that, as behaviour appears to be the result of a potentially complex interaction between a number of individual and contextual factors, it might be more fruitful to concentrate upon past user behaviour in order to predict future events. In other words, although the origins of user behaviour may be complex, there is little reason to believe a user in one situation one day will behave differently in the same situation at a later date, unless one or more of the factors affecting behaviour have changed in some significant way.

Nevertheless, this still leaves the question of which aspects of user behaviour to concentrate upon. It was proposed that a consideration of the errors (or misunderstandings) that occur during human-computer interaction might prove most fruitful, for two reasons: Firstly, errors may reveal underlying and fundamental mismatches between the model of the task and system held by the user and the model held within the system. Secondly, a consideration of errors within design and development is likely to highlight those areas that most require attention.

Within cognitive psychology there has been considerable interest in human error, as a basis for studying cognition. Many approaches to this area rely upon classifying human errors into different types. Such classifications allow us to view errors from particular perspectives. In essence, they are used to increase both the amount and the quality of the information about an error.

However, after some consideration it was decided that, although the overall approach of classifying errors might prove useful, human error classifications themselves were unlikely to be suited to analyzing user-system errors. Their main drawbacks were, firstly, that the concepts used in these classifications were primarily aimed at analyzing human cognition and these concepts do not easily map onto computing concepts and constructs. Secondly, user-system errors are not the same as human errors, in that the latter is a failure in cognition while the former is a failure in communication between the human and a system. In short, it was concluded that any classification of user-system errors would need to emphasize the failure in communication rather than failure in any of the participants to the communicative process, and that it would need to use concepts and terms that might have validity in both cognitive and computing domains.

With these constraints in mind, a classification scheme was developed (ECM: an Evaluative Classification of Mismatch) as a vehicle for further research, and an initial pilot study showed that user-system errors could be classified using the scheme. This suggested that the concepts it employed did have some validity in both cognitive and computing domains.

The second study of ECM involved a design team at Hewlett Packard's Office Products division in Wokingham. This study demonstrated that the classification scheme was usable by a design and development team that consisted of software engineers, human factors engineers, and technical authors. Furthermore, all of the team reported their belief that the use of the scheme was likely to improve system design.

The third and final study of ECM was concerned with the question of whether ECM can be described as useful, in the sense that it might be used to improve a design. A small system was implemented on a BBC micro-computer. Two iterations of the system were then produced. One iteration was produced from a set of recommendations made by individuals who did not use ECM in their analysis. The second iteration was produced from a set of recommendations made by individuals who did use ECM. The three systems (the original and the two iterations) were then evaluated in an experimental comparison. The system that had been changed using ECM was shown to be significantly better, in terms of time, errors and user attitude ratings, than either the original system or the iteration where ECM was not employed. This result indicated that the use of an evaluative classification scheme can significantly improve a design.

The first study lent support for the notion that behaviour at complex tasks is not determined by any single over-riding factor, such as cognitive style, while the development and examination of the ECM classification scheme has demonstrated that concentrating upon past user behaviour, particularly those points where breakdown occurs, can provide a useful way of matching the system to the user. In essence, it has been argued that consistently predicting detailed user behaviour might not be possible, and that a more useful way of considering user behaviour, and of exposing mismatches between the designer's and the user's model of the task, is to concentrate upon those points in a dialogue exchange where communication fails.

This research has provided strong indications that evaluative classifications can be of use within the design and development process. Furthermore, this work emphasizes the importance of providing structures for thinking about the user's problems that are divorced from the structure and terminology of design.

An Introduction to the Research and the Thesis

1 The aims of the research

The overall aim of the research, as stated in the abstract, was to address the question of how best to consider user behaviour. It has been suggested that predicting user behaviour in detail, during an interactive session, may not be practical. Underlying this assumption is the belief that complex formal methods, (whether they are founded upon design principles or an underlying cognitive model of the user) that are intended to guide design choice by predicting user behaviour, might be of use as research tools, but cannot be considered as practical tools for the design and development process.

The aim of the research was to investigate and, if possible, to demonstrate the worth of an alternative approach; that of considering user behaviour in terms of the dialogue failures that occur at the human-computer interface. The notion of developing, in the course of this research, a practical tool was considered only as a secondary (though desirable) aim. In essence, the objective of the research was to investigate the possibility of using evaluative classifications of user-system errors (or dialogue failures, misunderstandings, model mismatches, or whichever term we might choose to use) as a means for predicting future user behaviour at the interface. In particular, to predict, and to understand, those points in a design that might confuse users or cause them difficulty.

2 The aims of the thesis

As we might expect, the overall aim of the thesis is to set out both the research programme and the major research decisions and questions. The structure and aims of the thesis are reflected in the chapters. The chapters are as follows:

Chapter 1: To provide a short and general introduction and overview of the field of human-computeinteraction.

Chapter 2: To review some of the human-computer interaction literature relevant to the area of matching models within human-computer interaction.

Chapter 3: The critically consider the role of cognitive grammars within human-computer interaction, and to outline the case for using cognitive style measures to predict user behaviour.

Chapter 4:	To describe the first study, its results, conclusions and implications for the research.
Chapter 5:	To outline an approach to understanding the user at the interface; that of considering user-system errors, and to describe a scheme for classifying user-system errors, an Evaluative Classification of human-computer Mismatch (ECM).
Chapter 6:	To describe the pilot study of ECM, which tested whether the scheme was usable.
Chapter 7:	To describe the second study of ECM, involving designers at Hewlett Packard, which tested the scheme against several of the criteria for an evaluative classification set out in chapter 4.
Chapter 8:	To describe the third and final study of ECM, where a system was re-designed using ECM and tested in an experimental comparison against the original system and a system re-designed without using ECM.
Chapter 9:	To summarize the arguments for an approach that concentrates upon user-system errors. Then to consider how such schemes might be developed in the future, with particular respect to the way in which the user represents the task and system.

Chapter 1

A Brief Introduction to Human-Computer Interaction

1. 1 Overview

The purpose of this chapter is to briefly introduce and outline a view of human-computerinteraction that provides an overall framework for the research. Some of the reasons why we need research into human-computer interaction are discussed, and an explanation as to why human-computer interaction has become such an important issue in recent years is offered. Following this, the question *what is human-computer interaction* is addressed and a series of definitions, that forms the overall framework, is outlined. Human-computer interaction is characterized as consisting of five areas of inter-related interest; research into interactional hardware and software, research into matching models, research at the task level, research into design, and research into organizational impact. The research reported as part of this thesis is concerned with the second level in this taxonomy; that of matching models at the interface.

1. 2 Introduction

Throughout academia, industry and government there is an increasing awareness of theimportance of human-computer interaction. Evidence of this development can be seen not only in the interest shown by the general computing press, but also in the growing number of papers and books devoted to human-computer interaction (HCI).

The computing industry has been encouraged to increase its expenditure on HCI by large research programmes. These programmes have often funded joint collaborative projects between computer companies and academic institutions. Consequently, the strengthened links between academia and industry, together with the expansion in the numbers of researchers concerned with HCI, has led to a considerable growth in the numbers of HCI conferences, seminars and workshops.

The past ten years has witnessed large-scale development and progress in human-computer interaction and today HCI continues to expand, both in financial terms and in terms of the areas and disciplines it embraces.

1. 2. 1 Why do we need research into human-computer interaction?

In the opinion of many researchers in the HCI field, although computer technology has made great advances over the past thirty years, the designer's knowledge and understanding of the user has not significantly changed. It is now the communication with the user that is seen as the greatest obstacle to the efficient functioning of many systems.

Unfortunately, angry and frustrated users are the norm rather than the exception, as many researchers in the field have noted:

> "Users of advanced hardware machines are often disappointed by the cumbersome data entry procedures, obscure error messages, intolerant error handling and confusing sequences of cluttered screens. In particular, novice users feel frustrated, insecure and even frightened when they have to deal with a system whose behaviour is incomprehensible, mysterious and intimidating." (Bertino, 1985).

The importance of this problem has been highlighted by Baker (1977), who estimated that *people costs* exceed *machine costs* in human-computer interaction for ninety-five percent of the time. Many systems have been developed that are considered to be functionally excellent, but perform badly in the real world. The poor performance of these systems has been linked to the human-computer interface (Eason *et al*, 1975) and it is now generally accepted that poor interfaces can lead to stressed users, lower work rates, decreased job satisfaction and even higher absenteeism. These undesirable effects can be produced in a number of ways. Some of the following examples provide a flavour of what sometimes creates difficulties:

- Computer systems require users to remember too much information.
- Computer systems are intolerant of minor errors.
- Computer systems can seem confusing to new users.
- Interactional techniques are sometimes used for inappropriate tasks (eg command language may be unsuited for use in a task requiring the production of graphics/pictures).
- Computer systems often do not provide the information that is needed or produce information in a form which is undesirable as far as the user is concerned. Alternatively, systems may provide information that is not required.
- Computer systems sometimes do not provide all of the functions the user requires, and more often provide functions that the user does not need.
- Computer systems force users to perform tasks in undesirable ways.
- Computer systems can cause unacceptable changes in the structure and practices of organizations, creating dissatisfaction and conflict.

While these illustrations may underline the need for research into human-computer interaction, a question that they raise is; if computer systems have been in serious commercial use for the past twenty-five years why has human-computer interaction only become an important issue in the last ten to fifteen years?

1. 2. 2 The growth of human-computer interaction

In previous decades the majority of computer users were themselves programmers and designers of computer systems. Consequently, a person using a computer system was likely to have been immersed in the same conventions and culture as the individual who designed it. In recent years however, there has been a substantial growth in the number of users who are not computer experts. This change has focused attention upon the needs of what Eason (1976) has termed the *naive user* and the lack of understanding of the naive user on the part of many designers. Shackel (1985) summarizes the situation in the following way:

"The users [of computer systems] are no longer mainly computer professionals, but are mostly *discretionary* users. As a result, the designers are no longer typical of or equivalent to users; but the designers may ~~not~~ realize just how unique and therefore how unrepresentative they are." (Shackel, 1985).

As designers are no longer typical of most users, we appear to need tools, techniques, design practices and methodologies that will inform design teams of how users behave at an interface, and what users require from a system. It is a widely held view that this will require a multi-disciplinary approach. Branscomb (1983, cited in Shackel, 1984), the Vice President and Chief Scientist of IBM, clearly adopts this view:

"No longer the exclusive tool of specialists, computers have become both commonplace and indispensable. Yet they remain harder to use than they should be. It should be no more necessary to read a 300 page book of instructions before using a computer than before driving an unfamiliar automobile. But much research in both cognitive and computer science will be required to learn how to build computers that are easy to use". (Branscomb, 1983).

1. 3 What is human-computer interaction?

We have examined the question of the need for research into HCI. We now need to consider the question, *what is human-computer interaction?*

The term "human-computer interaction" is commonly used interchangeably with terms such as "man-machine interaction" (MMI), "computer and human interaction" (CHI) and "human-machine interaction" (HMI). In other words; these terms are usually taken to mean roughly the same thing, although researchers may argue over which is the most appropriate term. Here and throughout following chapters the term "human-computer interaction" (HCI) will be used, not because it is in any way *better* than other terms, but because it is the most common in the literature.

Having decided that "human-computer interaction" is the term we prefer to use, we now need to define it. An oversimplified definition of human-computer interaction might say that *it is the study and theory of the interaction between humans and complex technology (usually computers).* This may be acceptable as a general definition of HCI, but alone it does not do justice to the true complexity and multidisciplinary nature of human-computer interaction. Therefore, rather than suggest an extended alternative, five further definitions will be offered, each of which covers one aspect of HCI. These definitions reflect the different areas of study within human-computer interaction and their purpose is to supplement rather than replace the general definition just given. These definitions form the basis for the overall framework in which the research is discussed, although they will be assumed rather than explicitly discussed in later chapters.

1. 3. 1 Research into interactional hardware and software

First, and possibly most obviously, HCI is concerned with both the software and hardware of interactional techniques and technologies. While in the past the most common form of input has been in a command language dialogue style via a keyboard, many more methods are becoming available (the term *command language* simply means that the user types in commands in a particular format to the computer). For example, many systems are being developed that present the user with menus of alternative commands. These new techniques however, present not only new possibilities, but sometimes new problems. The problems of interactional techniques and how they affect communication between the user and the system is one of the central issues of HCI. Subsequently, *human-computer interaction is concerned not* *only with how present input and output technologies affect interaction, but also with the consequences of new techniques such as speech recognition and generation (input and output). The aim of human-computer interaction is to both develop interactional techniques and to suggest where and in what situations these technologies and techniques might be put to best use.*

1. 3. 2 Research into matching models

Second, HCI is the study of *how* users interact with computer systems. The central issue here, and throughout the whole of HCI, is how to match the computer system's model of the task to that of the user. If models are to match, then both parties to the communicative process must have a shared understanding of the task at hand; that is to say that communication requires *mutualknowledge* (Habermas, 1981). Consequently, the knowledge users need to operate a system, and *how* users apply their knowledge, has been the focus of much research aimed at providing tools to predict and describe user behaviour at the human-computer interface. In short; *human-computeinteraction is concerned with providing theories and tools for modelling the knowledge a user possesses and brings to bear on a task. Its purpose is to enable designers to build more usable systems by making explicit the user's model of the task and system.*

1. 3. 3 Research at the task level

Third, successful human-computer interaction depends upon systems fulfilling users' information needs and allowing them the freedom to perform tasks in the way that they wish. The *task fit;* the extent to which a system provides the information a user needs, is a major determinant in the success or failure of most computer systems (Eason, 1976). Therefore, good design requires the elicitation of the users' information needs and this is often more complicated than it first seems. For example, the users needs may not be fixed and constant, but may vary. This in turn may determine whether a task needs to be structured or unstructured.

This discussion of the task may appear similar to the discussion associated with the last definition. The crucial difference, however, is that the last definition concentrated upon *how* users perform tasks whereas here we are concerned with the overall nature of tasks and the users' information needs, although these areas clearly overlap to some extent. Consequently, *at a task level our concerns are with the means by which the user's information needs and a system's information provision might be matched. The* *purpose of human-computer interaction is to develop methods for determining users' needs, thus ensuring that systems provide users with the information they require, in the form they desire without excessive effort on their part.*

Figure 1.1: A representation of the different areas of study within HCI.

1. 3. 4 Research into design

Fourth, human-computer interaction is the study of both the individuals involved in design, and the design and development process itself. If any of the methods and findings produced by research into human-computer interaction are to influence the design of systems then they must be compatible with the process of design. Therefore, a consideration of the design process is, by necessity, an integral part of HCI.

The design stage in the life-cycle of many software products suffers from both inadequate *specification*, and a lack of communication between members of the design team (Macaulay *et al*, 1986). *Specification* is the process where the requirements and function of the system are agreed, or more often passed to the designers in the form of a document. A further compounding problem in design is that many designers pay more attention to the technical elegance of a system than they do to the more practical consideration of *is it usable?* This technological determinism (Bjørn-Andersen *et al*, 1986) is one of the major obstacles to good systems design. Consequently, it can be concluded that *human-computer interaction is the study of the design process. The aim of human-computer interaction is to suggest how design might be improved by taking* *more account of the user. In short; this means engineering a shift from system-centred to user-centred design.*

1. 3. 5 Research into organizational impact

Finally, human-computer interaction is the study of the impact that new systems have on organizations, as well as the impact they have on individuals and groups within these companies or corporations. When a system is installed within an organization the duties and responsibilities of many employees change. Groups of workers may find their status and influence within the new system has either increased or decreased. This includes secondary users as well as primary users (where primary users are those who interact directly with the new technology and secondary users are those who do not interact directly, but either provide input to the system, receive output or are affected by the system in some other way).

Understandably, change within any organization can generate difficulties, firstly for those who are introducing the new system, and secondly for the organization as a whole. Research in human-computer interaction is directed towards *the examination of the impact a new system has upon the roles of individual users and user groups within an organization. The objective of such research is to suggest both design and implementation techniques that might prevent problems such as job deskilling and conflict between groups.*

None of the definitions given above accurately characterizes human-computer interaction when taken alone. HCI has been artificially divided into these five supplementary definitions (see figure 1.1), and while this partitioning might prove to be a useful way of considering some of the different problem areas in human-computer interaction, in reality these areas are not easily distinguished. The true picture is of a multi-disciplinary approach to a whole series of different but related problems.

Chapter 2

The Background to the Research

2.1 Overview

The purpose of this chapter is to describe the background to the research. In the last chapter a view of human-computer interaction was proposed where it was characterized as consisting of five inter-related areas of research. As mentioned in chapter one, the research reported as part of this thesis is concerned with the *matching models* area of interest. Both the cognitive and the ergonomic or usability approaches to human-computer interaction address this area. Consequently, both are reviewed in this chapter, as well as the nature of the design process and the roles of the cognitive and usability approaches within design.

2. 2 Cognitive models in human-computer interaction

The communication that occurs at the human-computer interface is often considered to be a form of dialogue. Dialogue might be thought of as the exchange of symbols between two or more parties, as well as being the meanings that the participants in the communicative process assign to these symbols.

Within human-computer interaction a distinction is sometimes made between the *style, structure* and *content* of human-computer dialogue (Barnard & Hammond, 1983). The *style* of the dialogue refers to the, "..character and control of the information exchange". Command languages, menu selection, question-answering are all examples of different types of dialogue style.

The *structure* of the dialogue refers to the, "..formal description of dialogue elements in terms of their constituent structure together with their ordering within and between dialogue exchanges". For example, figure 2.1 shows a simple difference in dialogue structure for a command language statement. In the first case (1) the object is first and the operation to be performed upon it is second. In the second case (2) this ordering has been reversed. Most dialogue structure differences are likely to be more complicated than this simple example.

1	<Fredfile> (object)	delete (operation)
2	delete (operation)	<Fredfile> (object)

Figure 2.1: An example of a simple difference in command language dialogue structure.

The *content* of the dialogue refers to "..the semantics of the information exchanged - in terms of the user's general knowledge of the meanings of words and specific knowledge of the nature and consequences of computer representations and actions." (Barnard & Hammond, 1983).

It is generally accepted that human-computer dialogue is more restricted and less flexible than normal human dialogue (cf Sheehy, 1987). Given that this is the case, we might expect the development of a theory of human-computer dialogue, using the concepts of structure, style and content, to be relatively easy. However, this has not proved to be the case.

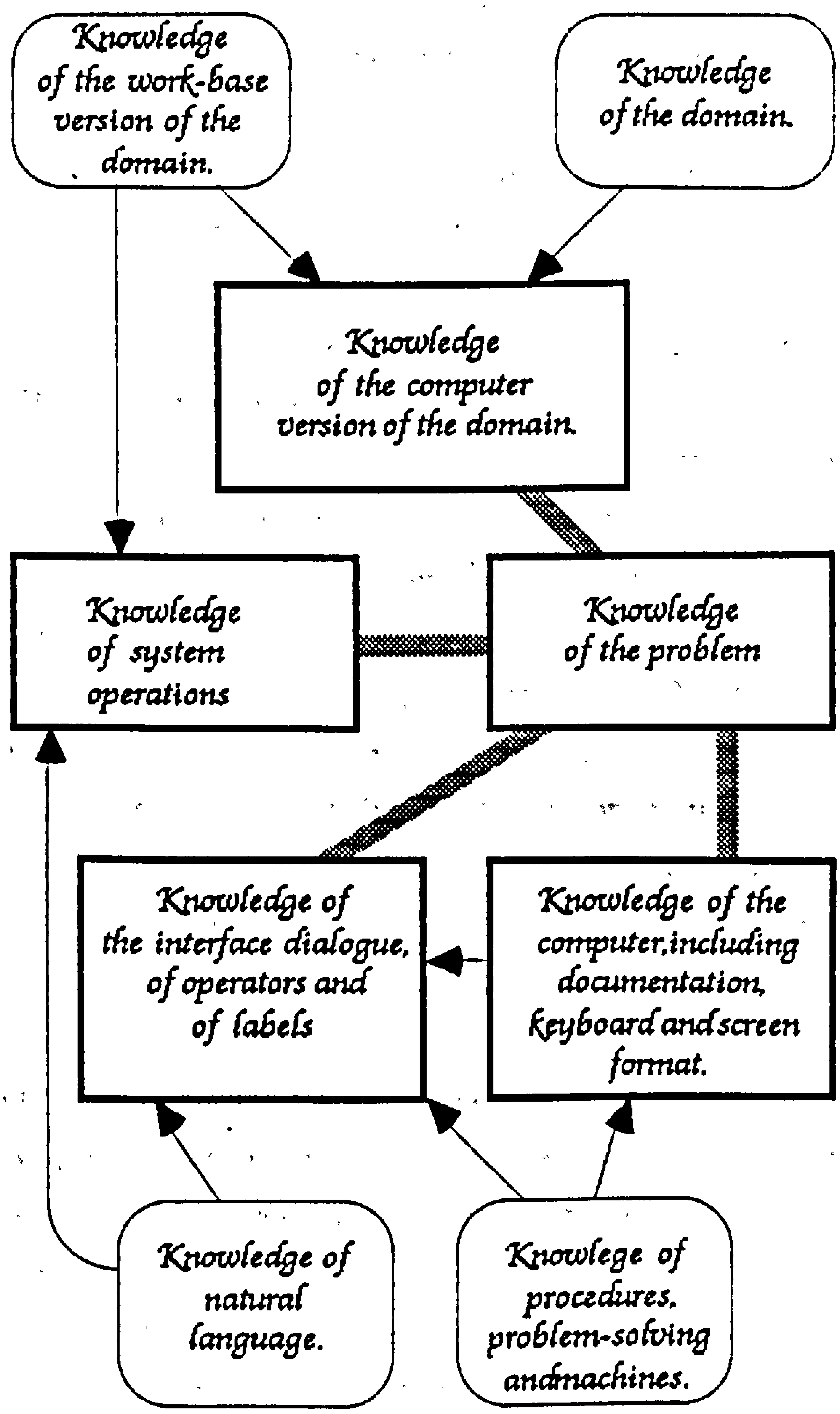


Figure 2.2: Morton *et al's* (1979) Block Interaction Model. The blocks with thick boundaries connected by thick lines indicate blocks of knowledge used by the ideal user; other rounded blocks and arrows indicate possible sources of interference or facilitation (adapted from Hammond & Barnard, 1985).

It has been found that the context of the interaction between the user and the system is a significant factor. The structure and content of dialogue, although important, does not affect the interaction between the user and the system in any simple and straightforward way (Barnard *et al,* 1981; Hammond *et al,* 1980; 1983b; 1987; Barnard & Hammond, 1982; 1983).

Users tend to recruit information and knowledge from related domains depending upon the demands of the task. A user's knowledge of the task, knowledge of natural language, knowledge of the machine can all potentially interfere with the interaction in subtle and complex ways. This interference can by represented by Morton *et al's* (1979) Block Interaction Model shown in figure 2.2.

It has been the complexity of the effects upon the interaction between the system and the user, often termed the *multi-determination of usability* (Barnard & Hammond, 1982), that led Barnard *et al* (1981) to argue that linguistic principles alone were not enough to account for human-computer dialogue. Furthermore, Hammond *et al* (1987) suggest that any general design principle is likely to miss the "..detailed context-dependent aspects of user cognition", although they do agree that such principles might, nevertheless, be useful in design.

These views have led to a shift in emphasis with HCI. Researchers have begun to consider the effects of the user's knowledge of the task, the system and related domains. Moreover, researchers have begun to look for the means to model the way users employ their knowledge. This shift, from a consideration of straightforward dialogue towards a consideration of users' models, is best illustrated by Barnard & Hammond (1983) who state that;

> "..the cognitive context of a dialogue exchange includes mental representations and cognitive processes relating not only to the explicit structure and content of the dialogue, but also those representations and processes relating to the general cognitive demands imposed by the system, information extracted from the wider task environment, the specific question or problem motivating an exchange, and the cognitive strategies mobilized in the course of learning, use and remembering." (Barnard & Hammond, 1983).

The overall finding of Hammond & Barnard's (1985) research was that the context of communication strongly affected dialogue at the interface. Consequently, drawing straightforward and valid rules about dialogue design from this research, without considering how context affects user behaviour, was not considered to be a realistic possibility.

This realization caused a shift of emphasis within HCI, as research began to concentrate on how users process information about a task and system. An example of this shift can be seen in Kidd's (1982) statement:

"If interactive computer systems are to be easy and efficient to communicate with then their dialogue design must be compatible with the information processing characteristics of the human mind." Kidd (1982).

This emphasis has led to one of the major themes in HCI; that of understanding human-computer interaction in cognitive terms. Storrs *etal* (1984) epitomize this view:

"The 'man' in MMI [HCI] is not primarily interacting with a machine but is interacting with information, program logic, knowledge, another intelligence. That this interaction takes place through computers and their peripheral devices should not be allowed to obscure that fact that it is essentially cognitive and that the most important issues are cognitive." Storrs *etal* (1984).

The character of this cognitive perspective has been summed-up by Green (1986):

"The special quality of a cognitive approach is, of course, that it starts from considering the mental life of the user, often using modelling techniques drawn from artificial intelligence and cognitive science to reveal what he or she knows about the interface and how this knowledge is put to use." Green (1986).

2. 2. 1 A cognitive perspective on human-computer interaction

The cognitive perspective on human-computer interaction suggests a particular way of considering both the user and the interaction between the user and the computer system. Several researchers have considered what users do at an interface from this cognitive perspective, most notably Donald Norman.

Norman (1986) points out that many of the problems that we experience, in operating machinery of any sort, can be related to the difficulties of linking our psychological goals to the physical variables and controls of the task. To illustrate this point Norman gives the simple example of filling a bath with water.

We have two psychological variables; the temperature of the water, and the rate at which the water flows into the bath. However, the physical variables that we can control are just two valves, one for the hot water and one for the cold water. Adjusting one of these valves changes, not just the temperature, but also the flow rate. In other words, our psychological variables do not directly *map* onto the physical variables of the task. If we had a bath where the flow rate control and the temperature control worked independently, then these controls might *map* onto our psychological variables (or goals) more accurately.

The term *mapping* is used widely in the psychological literature, but comes from the discipline of mathematics, where it is used to mean associating one element from one set to another element from another set. So we say that we *map* our psychological variables to appropriate physical variables, which means almost the same as saying that we *relate* our psychological variables to the physical variables of the task.

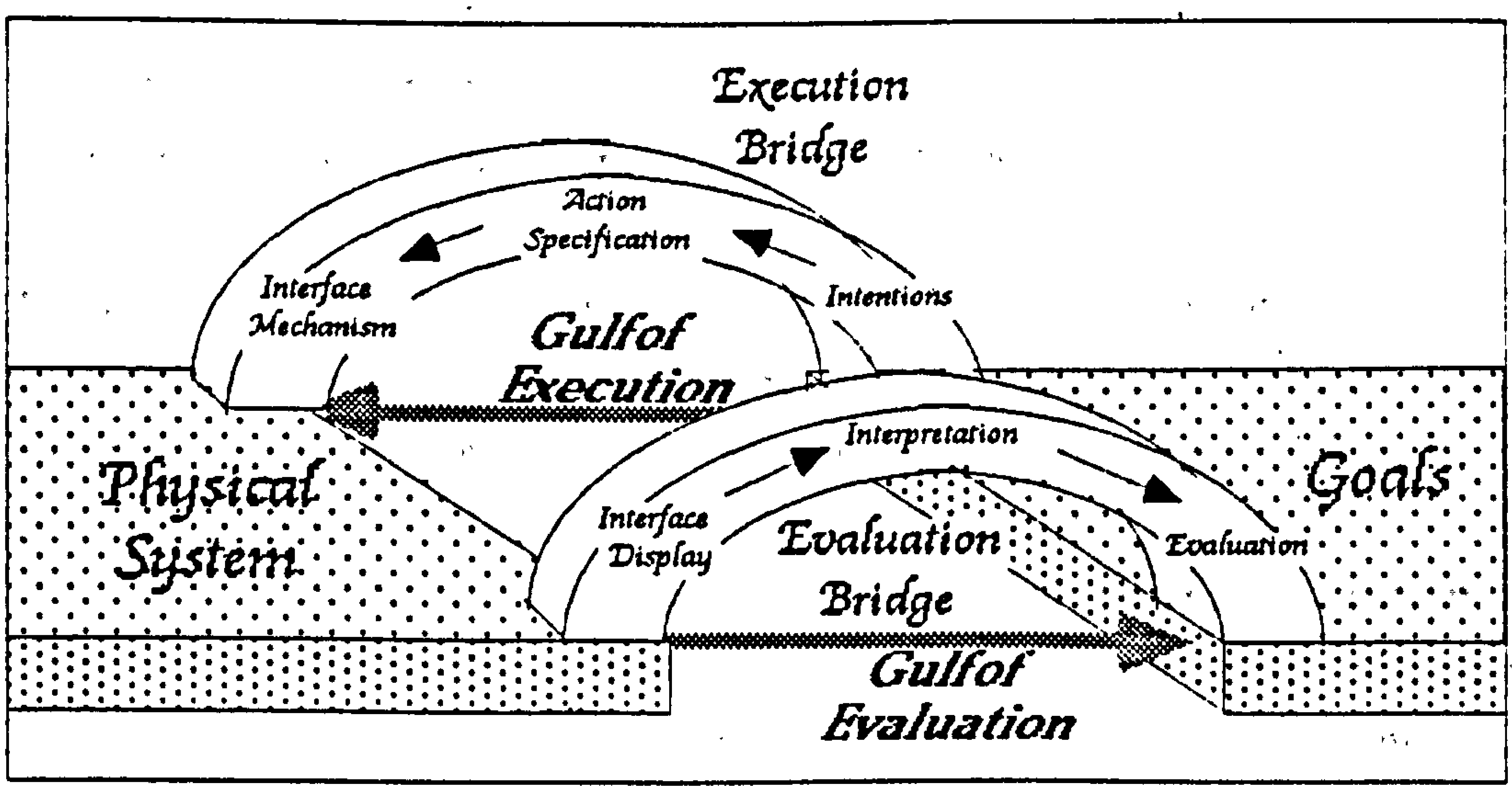


Figure 2.3: The gulfs of execution and evaluation, together with the bridges that span these gulfs. Adapted from Norman (1986).

The problems of mapping our psychological goals onto the physical variables of the task might be expected to become greater as the complexity of the task increases. This problem has also been identified by other researchers, such as Young (1981) and Moran (1983). Mapping difficulties have been described by Norman (1986) as being the gulfs that prevent users from dealing easily and efficiently with computer based tasks.

The *gulf of execution,* that Norman (1986) describes, is where the user knows what needs to be achieved, the user has a goal, but does not know which physical variables to adjust, or in what way to adjust them. The *gulf of evaluation* is where the system has altered, usually as a result of the user's actions, but the user cannot easily understand the change in the system's state. In other words, it is difficult for the user to work out what has happened to the system, and whether the change fits in with the initial goals and intentions. These gulfs are represented in figure 2.3.

Norman (1986, 1987) suggests that users engage in seven stages of activity. These stages span the gulfs of evaluation and execution shown in figure 2.3, and are as follows:

- Establishing the goal.
- Forming the intention.
- Specifying the action sequence.
- Executing the action.
- Perceiving the system state.
- Interpreting the system state.
- Evaluating the system state with respect to the goals and intentions.

Norman (1983c) previously suggested four stages of activity, but has since revised this to seven, and these stages form an approximate theory of action. The theory is only approximate because the seven stages are not necessarily discrete, and because the stages might not necessarily follow one another in a strict order.

What Norman has suggested is an account of how people interact with computers. This is not necessarily intended as an empirical theory, but is a useful way of thinking about the activities involved in performing a task. Norman's (1986) approximate theory provides a way of breaking down what user's do at the interface (for other models see Clarke, 1986; Moran, 1981; Nielsen, 1986). In essence, these different models or ideas provide a means by which we can structure how we think about the cognitive aspects of human-computeinteraction.

However, while approaches such as Norman's approximate theory of action allow us to break down the different components of human-computer interaction, if we wish to understand and predict the user we may need to understand how the user thinks about the task and system. In other words, we may need to consider the user's *mental model* of a system.

2. 2. 2 Mental models

It is commonly accepted within the psychological literature that people form mental models of tasks and systems, and that these models are used to guide behaviour at the interface. Norman (1983c) explains this in the following way:

"In interacting with the environment, with others, and with the artifacts of technology, people form internal, mental models of themselves and of the things with which they are interacting. These models provide predictive and explanatory power for understanding the interaction." (Norman, 1983c).

Although there is no general agreement as to the specific details of mental models, much of the research that has been conducted allows us to outline some of the general characteristics of mental models.

The first thing we can say about a mental models is that they are simpler than the entities they represent and as a consequence are incomplete (Johnson-Laird, 1983). When people encounter new machines, devices or computers they begin to construct mental models to represent their behaviour and operation. These internal models provide a means by which people can understand and predict the world around them, but we construct these models as we go along and as a consequence our models tend to be incomplete, unstable, do not have firm boundaries, are unscientific and parsimonious (Norman, 1983c).

Mental models are incomplete as it is rare that all of the subject matter concerning a task or system is known to the user. The instability of mental models can be seen by the way that people forget details, and confuse one system with another because of the lack of firm boundaries between models. What people do not know, they will work out and assume. One of the efficiencies of the human mind is the ability to use a small amount of knowledge to infer a great deal. Sometimes however, we assume incorrectly, and the mistakes and misunderstandings that we make are characteristic of these incomplete and confusable models of the world.

Mental models are parsimonious, according to Norman, because they are no more complicated than they need to be, and possibly as a result of this people are apt to maintainsuperstitious beliefs in the way that they act towards a system. For example, once we feel we understand a system at least well enough to cope with it, we do not usually expend extra effort trying to understand it further. It is, after all, easier to maintain a superstitious belief that apparently explains a system than to find out how it really works.

Norman's (1983c) description of mental models fits the views held by many other researchers in the field. As mentioned earlier, Barnard & Hammond (1983) found that users recruited knowledge from related domains depending upon the perceived demands of the task. In other words; users appear to have blocks of knowledge relating to different domains (eg natural language understanding) and use parts of these knowledge blocks when they believe that it is appropriate (Morton *et al,* 1979). The idea that users have coherent and accurate models is unsupported by much of the evidence. It appears as though users proceed through a task picking and choosing from both appropriate and inappropriate areas of their knowledge, only after much experience do users form a more precise and representative model of the system with which they aredealing.

2. 2. 3 User models

Although the term *user model* may have originally been intended to mean the user's mental model of a task and system, the term has come have a number of different meanings. Here, we will attempt to clarify the different uses of this term. Hammond *et al* (1983b) distinguish three main uses of this term:

First, the term *user model* can be used to mean a representation of the user embedded within a system. For example; a system for advising accountants on some aspect of the tax system might have a model of the user (the accountant) which dictates what the accountant can be assumed to know. This type of model is held within the system.

The second usage that Hammond *et al* identify is that where a user model is something closer to what Norman (1983a) calls a conceptual model. It is an ideal model which the ideal user might hold. It acts as a goal for the designer of a product during the design process. Hammond *et al* refer to this as the *design interface image* (this is similar to what Norman, 1986, calls the *system image*).

Third, and most common, the term *user model* can be taken to mean a model of the user's knowledge of the system and task. In this sense *user modelling* is taken to refer to the representation of the user's model of a system and task.

This distinction, although perfectly valid, is not the only view of the function and place of user modelling within HCI. Young (1985) makes a similar, but slightly different distinction between the different senses in which the term *user model* can be used:

First, it is the designer's model of the user. This is a predictive model which helps guide design and aids predictions about the overall performance of the human-computer system.

Second, it is the user's conceptual model of the system. In other words; it is the internal model the user has of the system.

Third, the term *user model* can be used to mean an embedded user model. This is a representation of certain aspects of the user which is implicit within the software of the machine and is used to adapt the system to suit the user. It is possible to argue that all systems have an implicit model of the user embedded by the designers within the software. Here however, Young is referring to *explicit* models within computer systems.

The view which appears to have been most commonly accepted within HCI as to the possible uses of the term *user model* has been expressed by Clowes (1987), who suggests that there are the following types of user models:

- The designer's model of the user.
- The user's model of the task.
- The user's model of the system.
- The system's embedded model of the user.

Although this view appears to be generally accepted, the degree to which the user's model of the task and the user's model of the system can be distinguished practically is in some doubt. Furthermore, it is important to recognize that views on user modelling which are alternative to these are not necessarily wrong; they represent different ways of considering the problems of HCI and as such are still potentially valuable contributions.

User modelling, as a process, is usually taken to entail the use of cognitive grammars to analyse the task in question. However, as this issue is of considerable importance to the thesis as a whole, it will be considered in the next chapter.

2. 3 The Usability approach

An approach to improving human-computer interaction that does consider the user is the *ergonomic* or *usability* approach. In contrast to this heavily theoretical cognitive approach to HCI, the usability perspective might be characterized as an approach that first addresses the practical issues and second theoretical issues, although some might dispute this and argue that the two go hand-in-hand.

Usability problems appear to afflict all manner of complicated products, from complex IT systems to everyday household items, such as video recorders and washing machines. The issue of concern is how to mitigate the effects of these usability difficulties, or better still, how to ensure that usability problems never arise. But how might this be achieved, and what is the *usability* approach to the problems of human-computer interaction?

2. 3. 1 Concepts for usability

A useful starting point is provided by Gould (in press) who sets out a list of the components of usability (see figure 2.4). Gould's list implies the notion of usability to be a broad concept that involves many of the stages and aspects of design and implementation.

System performance
- Reliability
- Responsiveness

System functions

User interface
- Organization
- Input/output hardware
- For end users
- For other groups

Reading materials
- End-user groups
- Support groups

Language translation
- Reading materials
- User interface

Outreach program
- End-user training
- On-line help system
- Hot-lines

Ability for customers to modify and extend

Installation
    Packaging and unpacking
    Installing

Field Maintenance and serviceability

Advertising
    Motivating customers to buy
    Motivating user to use

Support-group users
    Marketing people
    Trainers
    Operators
    Maintenance workers

Figure 2.4: Gould's (in press) list of usability components.

In essence, Gould is suggesting the areas that a study of usability should cover. He is not only considering the end-user or operator (the person at the computer terminal), but also all of the other types of *user*. This includes programmers, systems engineers, installation engineers, the people who support the users, etc. These are the aspects of system development and implementation that directly affect the total usability of a system.

Nevertheless, although Gould's list is undoubtedly helpful, his list does not suggest concepts that might contribute to a definition or better understanding of usability. Such concepts may be needed if we are to understand what makes a usable system easy to understand and operate. Eason (1984) however, has suggested a series of concepts for just this purpose (see figure 2.5).

Several researchers have noted that usability is not determined by just one or two constituents, but is influenced by a number of factors (eg Barnard & Hammond, 1982). These factors do not simply and directly affect usability, but interact with one another in sometimes complex ways. Eason (1984) has suggested what these variables might be (see figure 2.5).

TASK CHARACTERISTICS: First we will consider the two task variables that Eason identifies; *frequency* and *openness*. The term *frequency* simply refers to the number of times any particular task is performed by a user. Eason (1976) points out that if users perform a task infrequently then they will expect a dialogue that guides them through the task. On the other hand, such a dialogue may not be appropriate for a task that is performed routinely. For a frequent task users may well expect an economic dialogue. This is because they can easily remember the steps that are required for the task and do not usually require help or prompting.

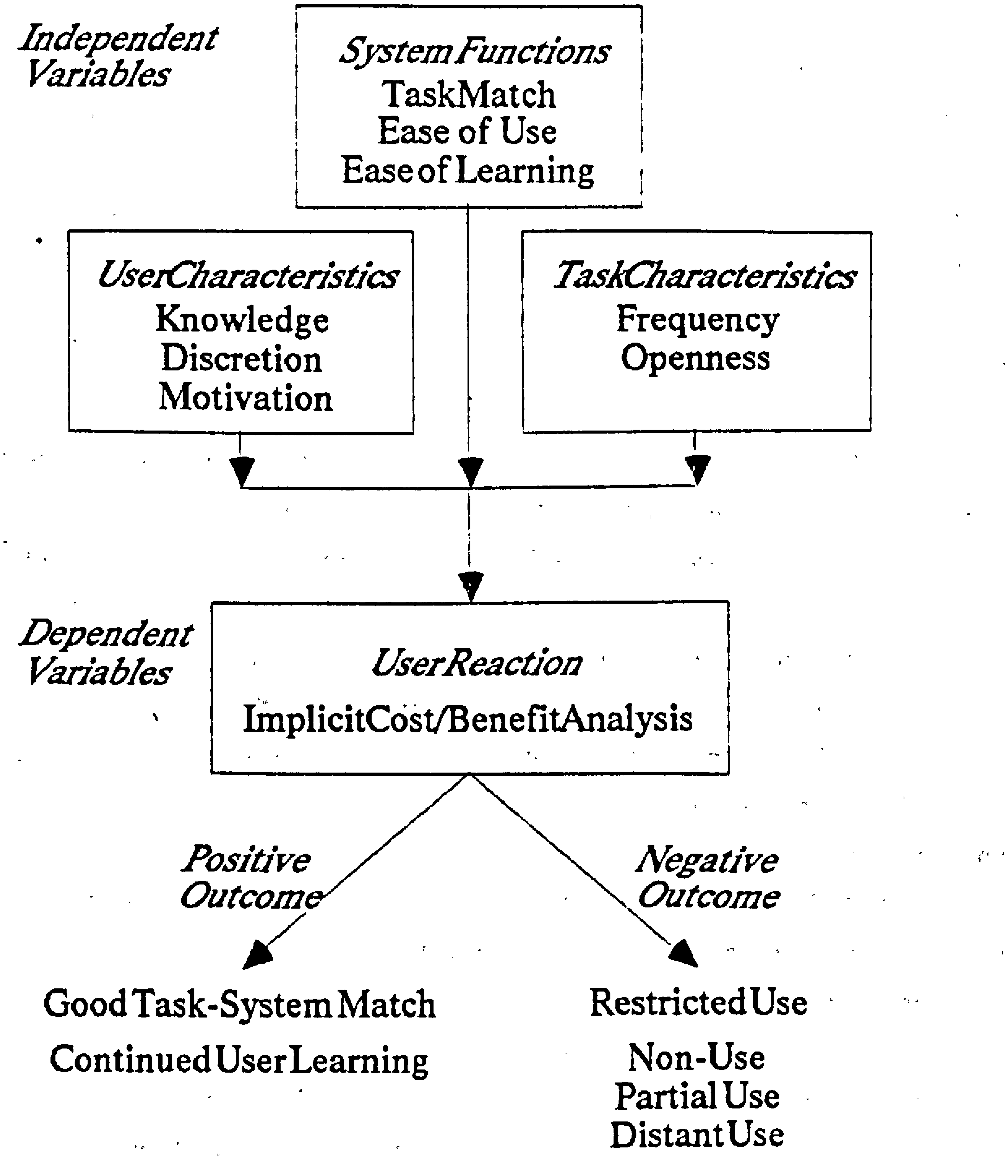


Figure 2.5: A causal framework of usability, taken from Eason (1984).

The second term *openness* refers to the extent to which a task is modifiable. An open task is one where the information needs of the user are variable. Consequently, the task must be structured to allow the user to acquire a wide range of information (Eason, 1976). Alternatively, the user's information needs may be fixed. If this is the case then the task need not be open and flexible, as the same information is required each time the task is performed.

SYSTEM FUNCTIONS: The three major system variables that Eason (1984) identifies are *ease of learning, ease of use,* and *task match.* The term *ease of learning* refers to the effort required to understand and operate an unfamiliar system. Clearly, this will depend upon the knowledge the user possesses and the ease with which this knowledge can be mapped onto the unfamiliar system. The second term, *ease of use,* refers to the effort that is required to operate a system once it has been understood and mastered by the user.

At first, *ease of use* and *ease of learning* appear to indistinguishable concepts. However, it is quite possible to have a system that is easy to learn but difficult to use, or a system that is difficult to learn but easy to use. For example, consider a system that is easy to learn. This system may be thoughtfully explained and its dialogue may guide the user easily through the various tasks. However, once the user has come to know the system well, the dialogue that was initially so helpful could become obstructive and time consuming. In other words, by not allowing the user short-cuts through tasks, and by insisting that everything is repeatedly explained to the user, the system becomes difficult and frustrating to use, although it was initially easy to learn.

On the other hand, a system that has many abbreviated commands and little explanation may be difficult to learn, but relatively easy to use. For example, the UNIX operating system requires a user to type "cd" to change from one file directory to another. This sort of system is often frustrating and difficult for users to learn, but easy and straightforward for users once they have mastered the system. The dialogue of such a system provides little explanation of how the system works for first-time users. Nevertheless, such a dialogue does not obstruct the experienced user. In short, the concepts *easy to learn* and *easy to use* can be distinct and relatively independent.

The third system concept that Eason (1984) suggests is that of *task match.* This term refers to the extent to which the information and functions that a system provides matches the needs of the user. In short, a system may be easy to learn and easy to use, but does it do the job? This is a question of whether the system provides the necessary functions that are required, as well as the information that the user needs.

USER CHARACTERISTICS: The final set of variables that Eason (1984) identifies are those that belong to the user. These are *knowledge, motivation* and *discretion.* The user's knowledge of the task and system have already been discussed. However, to reiterate a little of what was stated earlier; the knowledge that the user chooses to apply to a task, whether that knowledge is appropriate or not, can be considered as a variable that contributes towards the ultimate usability of a system.

The second of Eason's variables is *motivation.* This term is used with respect to the user's motivation to perform a task. If the user has a high degree of motivation then more effort will be expended in overcoming problems and misunderstandings. Alternatively, if the user is not strongly motivated to complete a task then the user's commitment to the system may wane, and there may be a reluctance to learn or use complicated parts of a system.

The third variable, *discretion,* refers to the user's ability to choose not to use some part, or even the whole of a system. A user has discretion every time a choice is presented. However, in some circumstances this choice may be limited. For example, many shop assistants in supermarkets use systems that recognize bar codes on the foods they sell. They appear to have little discretion as to whether and how they use the system. On the other hand, a statistician in a business may have more discretion. Besides the large number of statistical techniques that a statistical software package will offer this user, the statistician will also have the ultimate choice of not using the system at all, as long as the information that is required is produced.

The essence of what Eason (1984) suggests is that the usability of a system will depend, not only upon the variables of the system, but also upon the characteristics of the task and system. That is to say, that the variables of task, system and user all combine to determine the usability of a system. Eason's stance has been to consider usability from the perspective of how systems are used in a work environment. While such a view is undoubtedly useful from a global standpoint, it is not the only perspective on usability, although it may be one of the more comprehensive.

2. 3. 2 Definitions of usability

The global definition of usability suggested by Eason (1984), and implied in the last section, is that the,

> "..major indicator of usability is *whether* a system or facility is used.." Eason goes on to say, "It is the [user's] act of choice which is the essence of usability which suggests that the crucial measure [of usability] is the pattern of [the user's] responses to options and the way ..[these responses] build into a learning or non-learning strategy." (Eason, 1984).

The argument in favour of this view, and against trying to measure usability in the laboratory, goes as follows: If we force an individual to use a system in order that we might assess its usability, then we may be destroying the best measure of the usability that we have; *whether or not a system is used.*

However, definitions of usability have often been tied to the question of how to evaluate the usability of a system. For example,

> "Everyone knows what usability means until its recognition as a criterion implies evaluation, which requires measurement and operational definition." (Shackel, 1981, cited in Twigger, 1986).

Shackel's operational definition of usability.

Usability can be specified and measured via the operational criteria defined below. The terms should be given numerical values when the usability goals are set during the design stage of 'requirements specification'.

For a system to be usable the following must be achieved -
The range of required tasks must be accomplished -

Effectiveness

- at better than some required level of performance (eg in terms of speed and errors),
- by some required percentage of the specified target range of users,
- within some required proportion of the range of usage environments.

Learnability

- within some specified time from installation and start of user training,
- based upon some specified amount of training and user support,
- and within some specified re-learning time each time for intermittent users.

Flexibility

- with flexibility allowing adaptation to some specified percentage variation in tasks and/or environments beyond those first specified.

Attitude

- and within acceptable levels of human cost in terms of tiredness, discomfort, frustration and personal effort.

Figure 2.6: An operational definition of usability. This has been taken directly from figure 4 in Shackel's (1986) paper.

When Eason's definition is viewed in this light certain problems emerge. The difficulty with this ideal approach, when taken to its extreme, is that it suggests that we must build and implement a system, and then wait to see what happens if we wish to evaluateits usability. Although data from systems that have already been implemented may prove to be the most rich and possibly the most useful, this is not an approach that could be recommended to cost-conscious design teams. We appear to need definitions of usability that will allow us to evaluate systems early in the design process, as well as during and after implementation.

Shackel (1986) has suggested just such an operational definition of usability. This definition can be seen in figure 2.6. The essence of this definition is that all its constituent elements are measurable. What Shackel is suggesting is that any system should have to pass the usability criteria of *effectiveness, learnability, flexibility,* and user *attitude.*

A system must be *effective,* in that a certain proportion of target users must be able to use the system in a number of environments, within a certain time and without too many errors. A system must be *learnable,* in that users must be able to learn the system after a certain amount of training. Furthermore, users who do not frequently use the system must be able to relearn the system within a certain time. A system must be *flexible,* in that user performance must not deteriorate by more than a certain percentage across tasks and environments. Finally, a system must provoke user *attitude* ratings where a certain percentage are positive towards the system.

To date, Shackel's (1986) definition is one of the best operational statements of usability that we have, although it is certainly not the last word on the subject. Nevertheless, it has been this sort of operational definition of usability, suggested by Shackel (1986) and others (eg Bennett, 1984; Bury, 1984; Gould, in press; Gould & Lewis, 1985) that is beginning receive considerable attention within industry.

2. 3. 3 Evaluating the usability of a system

The utility of usability definitions such as Shackel's is that they provide a means for structuring usability goals. For example, taking Shackel's scheme, we require effectiveness, learnability, flexibility, and attitude goals for any product. But how can we evaluate a product against these goals?

Hewett (1986) distinguishes two forms of evaluation; *formative* and *summative.* The difference between these two types is in their purpose. Formative evaluation helps the designer to refine and *form* the design. Because of this objective certain types of evaluative measure might not be appropriate for this type of evaluation. For example, an overall score, however it is derived, is unlikely to tell a designer what should be done to improve the design. Formative evaluation is more likely to require qualitative information that can be used to help the designer pinpoint those parts of the systems that require alteration.

Alternatively, summative evaluation is more likely to require quantitative information rather than qualitative data. As Hewett (1986) explains:

"Summative evaluation involves assessing the impact, usability and effectiveness of the system - the overall performance of user and system." (Hewett, 1986).

Hewett goes on to suggest that different types of evaluation may be suited to different stages in the design process. For example, qualitative data may be required when a design is being refined. Designers will need to know more about why errors or misunderstandingsoccurred rather than just their absolute numbers. In the later stages of design quantitative information may be required. This will allow the designers to assess the usefulness of changes to the design. If problems are shown to exist within a design at this stage then more qualitative data may be required to inform more design changes.

This distinction between formative and summative forms of evaluation also serves to highlight one of the problems with Shackel's (1986) definition. It is not clear where the evaluation proposed by Shackel is supposed to take place. If it is in the earlier stages of design then simply acquiring user attitude scores and error counts may not properly inform the designers of the type of changes that are required. In essence, Shackel's operational statement of usability appears to be aimed only at the summative evaluation of a system.

2. 3. 4 Measures for evaluating systems

Having considered the different forms of evaluation we might employ, we will now consider some of the measures that might be used to evaluate a system. Furthermore, the ways in which these measures might be used are also considered. As the latter part of the thesis is concerned primarily with evaluation of systems this subject will be considered in some detail.

Time: One of the most commonly used measures of usability is the time it takes a user to perform a task. Measures of time have the advantage that they are easy to measure and suitable for statistical analysis. The problem with using simple measures of time is that it is not always clear what these measures should be considered against. Often the time it takes a naive user to perform a task is compared to the time it took an expert user to perform the task. A more sophisticated means of calculating user performance has been suggested by Whiteside *et al* (1985).

Whiteside *et al* (1985) have proposed a work rate metric for assessing the usability of systems. This metric is as follows:

$$S = \frac{1}{T} PC$$

where:

	S	=	the user's performance score,
	T	=	time spent on task,
	P	=	percentage of task completed,
	C	=	arbitrary constant based on the fastest possible task solution for a practiced system expert.

This metric gives an overall indication of work rate performance by taking account of what can be achieved (the expert's performance, C), the time the user takes (T), and the percentage of the task the user manages to complete (P). Consequently, this can be thought of as a more sophisticated metric than a simple measure of time.

However, there is no need to restrict measurement to the time it takes a user to perform a task. Bennett (cited in Shackel, 1981) has suggested that we might also measure the time it takes to adequately train users to use a system, the time that is required before the user can perform actions automatically, the time it takes users to *warm-up* after a period of non-use of a system, and the time the user requires to recover from errors.

Errors: The errors that occur at user interfaces are potentially one of the most useful sources of information. Measures such as time only provide gross indications of where the user experienced difficulties. Errors on the other hand, have the potential to show where problems exist within a system. Moreover, a study of user-system errors may also suggest the cause of a difficulty.

However, as Norman (1983b; Lewis & Norman, 1986) points out, the term *error* may not be the most appropriate for describing misunderstandings at the human-computer interface. The word *error* seems to apportion blame, where the terms *misunderstanding* or *dialogue failure* seem to more accurately characterize what happens when a user and a system fail to properly communicate with one another. For the moment however, we will continue to use the term *error* because it is so commonly used within the literature.

Errors are unusual as a form of data as they provide both quantitative and qualitative information about usability (see figure 2.7). For any task simple counts of errors, or counts of particular error types can be made. Error counts for novices can be compared with error counts for experts or intermediate users. The problem with using errors in this way is that it is sometimes difficult to decide what is and what is not an error. Used qualitatively, errors can be a rich source of information. This is possibly because errors can often be related to points in the design that do not match either the user's needs or the user's understanding of the task.

In order that we might better understand user-system errors Norman (1983b) has suggested a classification scheme. Norman (1983b) suggests that an error in a user's intention is a *mistake*, while an error in carrying out an intention is a *slip*. These types of classification arise from the psychological study of human error (see also Norman, 1981a; Reason & Mycielska, 1982). Norman (1983b) also suggests the categories of *mode* errors and *description* errors. Mode errors are the type that occur when users believe that they are in one mode, when they are really in a different mode (for a discussion of mode errors see Monk, 1986). Description errors occur where there is insufficient specification of an action and the confusion leads to the wrong action. For an example of the use of Norman's classification scheme see Riley & O'Malley (1984) who use it in their approach to planning nets, mentioned earlier.

Verbalprotocols: These are written statements of what a user has said either during or after a task. There are two types of protocol; *concurrent* and *retrospective*. A concurrent protocol is one that is taken while the user performs that task. A retrospective protocol is one that is taken once a task has been completed.

A concurrent verbal protocol has the advantage that it is likely to produce more data about a task and system. Distortions may arise in this type of protocol, however, as the act of providing the protocol may influence the user's view of the task and consequently the user's behaviour. A retrospective protocol has the advantage that it is unlikely to affect the user's behaviour at the task. Unfortunately, distortions in the protocol may occur as the user will only remember parts of the information that might have been provided by a concurrent protocol (for a discussion of verbal protocols and their relative advantages and disadvantages see Ericsson & Simon, 1980). One alternative to using single user protocols, described by Gould (in press) is that of using two users simultaneously. The users speak to one another about the system they are using and their dialogue provides the data for analysis.

Visual protocols: These can be taken using a video camera. As with verbal protocols the information that this sort of measure can produce is qualitative and requires interpretation. Distortions in behaviour can arise as users may behave differently when they know that they are being filmed. The advantage of this technique within industry is that design teams can be shown where users experienced difficulties with a system.

Visualscanningpatterns: Saccadic eye movement studies can show where a user's eyes fixated on the screen and for what duration of time. In other words, this sort of data can show us what information the user was looking at, at what time, and for how long. However, such data requires careful interpretation. For example, if we know that a user was looking at a certain piece of information shortly before an error was made what does this tell us? It may be that this information was crucial to the error. On the other hand, the user may have been thinking about the problem and just have happened to be looking at that particular part of the screen. Nevertheless, this sort of data can show scanning strategies and demonstrate when some of the information on the screen is viewed for greater periods of time. For an example of this sort of study see Graf *et al* (1987).

Patterns of system use: Earlier in this chapter Eason's (1984) definition of usability was explained. The essence of this definition was that the ultimate measure of usability was whether a system was used. By placing prototype systems in work environments and studying their patterns of use it is possible to assess which parts of a system are more frequently used. Such an approach may indicate that certain sections of a system are easier to use, or that certain tasks are more amenable to automation. Moreover, it may be possible to see which user groups prefer a system and which user groups avoid it.

Attitude measures: These sorts of measures are usually elicited using questionnaires or interviews (for an example see Brooke, 1986). These measures often cover a wide range of areas. For example, questions may be asked about the user's opinion of the learnability of the system, the ease with which the system could be used, and whether the system adequately performed the task. Other types of questions that are often asked are whether the user felt in control of the task and situation, or whether the user felt frustrated and not in of control on the interaction.

Althoughattitudemeasuresrequire careful construction and validation, and may be open to bias, they nevertheless provide a valuable source of information. This type of user feedback can serve as an indication as to whether a system is likely to be used and appreciated in the work environment.

Cognitive complexity measures: These type of measures are produced by using some of the modelling techniques described earlier. They can be used to assign cognitive complexity scores to various parts of a design. This is done by counting the number of rules that were required in cognitive grammar to describe a particular part of a system. Some of the advantages and disadvantages of this sort of approach have already been discussed earlier. It is difficult to argue that the cognitive complexity scores that are produced using the modelling techniques can be used in any other way than quantitatively. However, the proponents to the cognitive modelling approach argue that the process of producing the grammar describing part of a system produces qualitative information in the form of a better understanding of the task and system.

Measure	Can be used qualitatively	Can be used quantitatively
Time	No	Yes
Errors	Yes	Yes
Verbal protocols	Yes	Sometimes
Visual protocols	Yes	Sometimes
Visual scanning patterns	No	Yes
System use patterns	No	Yes
Attitude	Yes	Sometimes
Cognitive complexity scores	No	Yes

Figure 2.7: A table of evaluation measures together with the type of information that they can be used to produce.

2. 3. 5 Methods for evaluating systems

Now that we have considered some of the measures that can be used to assess a system's usability, we shall address the question of what methods to use. In other words, how can we use the measures we have just discussed?

Friendly users: It is commonly accepted that most evaluation programmes should involve users at some point, but what sort of users? Evaluation should certainly involve users from the different user groups that have interests in the system. However, other alternatives to using naive users include using friendly users who have some technical knowledge to make suggestions as to possible alterations that might be made to a system. The advantage of this is that people can be used who have a reasonably good knowledge of how systems work, whereas naive users are often unable to suggest how the design might be changed from an informed point of view. Unfortunately, as Hewett (1986) points out, these users tend to miss aspects of a system that often cause difficulties for naive users.

Hostile users: A possible alternative to using either naive or friendly users is to use hostile users. This is the sort of user that has no investment in the system. In other words, the sort of user who does not care if the system fails. When hostile users attempt to destructively use and criticize a system they often expose many inconsistencies and flaws that would have created difficulties for naive users (Hewett, 1986).

Simulating users: Another possibility suggested by Hewett (1986), is that of simulating users. If the progress of several naive users is charted, then later their routes through the system can be replicated by the designers. The advantage of this approach is that the designer is taken through the system in a way that may not have been previously envisaged. This way problems within the system can be identified and rectified.

Expert review: A review such as this is where a system is examined by either a human factors expert or a designer who has not been directly involved in the project. The advantage of such a review is that comments and criticisms are made from a position of knowledge. The disadvantage of this sort of approach is that the expert is often immersed in the same technical culture as the designer, even if not to the same extent. Consequently, although such examinations can provide useful information about system inconsistencies, there is also the likelihood that difficulties that will hinder a naive user will not be detected.

Simulation trials: A simulation trial is where a system is tested using rough prototypes or mock-ups of the intended system. When a system is built a great deal of effort, time and expense is put into creating and implementing a design. Understandably, designers are sometimes reluctant to change their designs, which often include elegant solutions to difficult technical problems. If a system can be tested before the majority of this effort and time is expended then the first iteration of the design is likely to be closer to the eventual *usable* design. The advantage of this approach is that time and money is less likely to be wasted in designing a system that is substantially unusable, as mock-ups (eg using pencil and paper type tasks) and rough prototypes do not require the technical elegance of the eventual design.

Iterativeinformallaboratoryexperiments: This approach is based around iteratively refining mock-ups or prototypes of a system (Twigger, 1986). The iterations of the system are refined using predominantly qualitative information gained through user testing, expert review and alike.

Formallaboratoryexperiments: The advantage of formal experiments is that they control the environment. As a result, experimenter effects (eg the biases of the design team) are less likely to affect the results. This sort of approach tends to be based around collecting quantitative data to confirm or refute hypotheses. The disadvantage of formal laboratory experiments is that they are often far removed from the work environment in which systems are used. Consequently, there may be doubts as to whether the user's behaviour in an experiment can be taken as reliable indicator of behaviour outside the laboratory.

Field trials: A field trial is where a system is placed within an organization prior to its formal release so that its use can be studied. Measures such as patterns of system use and user attitude questionnaires can be used to assess the potential success and usability of the product. This information can then be used to refine design. The disadvantage of this sort of approach is that it tends to be time consuming, particularly if data is required on long-term use of a system.

Follow-up studies: Once a system has been designed and implemented within an organization there would appear to be little that might be done to improve its design. However, many systems are refined over time and updated versions of systems are often produced and released by manufacturers. By following a product into an organization and using attitude scales and interviews with users it is possible to provide qualitative data that can be used to refine the next release of a system. Several of the large IT companies (eg Hewlett Packard) have adopted this approach (Twigger, 1986).

Field studies: These are studies of systems in organizations. They tend to be more rigorous than simple follow-up studies or field trails, although as formal experiments they are difficult to control. The advantage of this approach is that it can be seen as an ultimate measure of the usability of a system; *whether or not the system is used in the work environment* (Eason, 1984). Moreover, this type of study also provides informationthat can help us to understand why some systems are used and others are not.

2.3.6 The difficulties with the usability approach

The idea of setting usability goals for products has been well accepted within both academia and industry. Unfortunately, the question of who sets usability goals and how are they set, has received less attention. One argument is that a system might only be as usable as its usability goals. In other words, if we choose inappropriate goals then no matter how well we meet these goals the system will still fall short of being usable. Furthermore, the degree to which a system fails to meet usability demands may be proportionate to the gulf between the goals we set and the needs of the user.

By setting usability goals we are supposedly making usability a specific design objective. However, it is possible that when we set usability goals, without some process to inform how these goals are formulated, we are placing the usability problem one step back, rather than making a genuine contribution to the usability of the proposed system. One approach to this, that is commonly accepted (eg Damodaran, 1983; Eason, 1985), but not always made explicit, is to incorporate the user into the process of how usability goals are set, and not just when and how a system is measured against these goals.

A further problem with the usability approach is with the definitions of usability. As mentioned earlier in this chapter, Shackel's (1986) operational definition of usability did not take account of how we might use qualitative data to inform and refine the design of systems. Although quantitative data is required to accurately assess the usability of a system, it is the qualitative information that informs designers how to change an unusable system. In other words, operational definitions such as Shackel's suggest how we might practically measure usability, but do not give any indication how we might improve the usability of a design. While it may seem a nonsense to suggest that we might quantitatively specify how designers might use qualitative information, we nevertheless need to acknowledge the important role of qualitative information within iterative design.

These criticisms, that usability definitions are only concerned with quantitative information, also serve to highlight another related problem. The usability approach suggests how we can measure the usability of a system, but does not say how a system might be changed in order to improve its usability. One reply to this sort of criticism is that a formal consideration of usability can only suggest where problems exist, it cannot suggest how to design and refine a system, as this is a creative process. However, an approach that provided a means for understanding what happens when humans interact with computers could suggest how systems might be improved, although design solutions might always require a degree creativity from the designer. In essence, the criticism is this; *the usability approach may show us how to measure usability, it may suggest what evaluation methods we should use, but it does not provide a framework* *forunderstandinghuman-computerinteraction.*

2. 4 The Design Process

Having discussed both the cognitive and the ergonomic or usability approaches to human-computer interaction, we will now consider the design process. Moreover, we will consider the cognitive and usability approach within the context of design.

A simplified view of the stages of design is presented in figure 2.8. In the first stage information is collected about the tasks a system will have to perform, and who its users will be. In the next stage the system is designed and specified. That is to say, that the questions of how the system is going to work and what it is going to do are set out in detail. Part of this stage is to partition different parts of the design to different programmers and to state how these different programs are going to link together to build into the system as a whole. The third stage is to build the system, and to test both the parts, and the system as a whole to see if it works. The final stage is to deliver the system to the user. Once the system has been implemented within an organization it may require adjustments, and minor changes are often made.

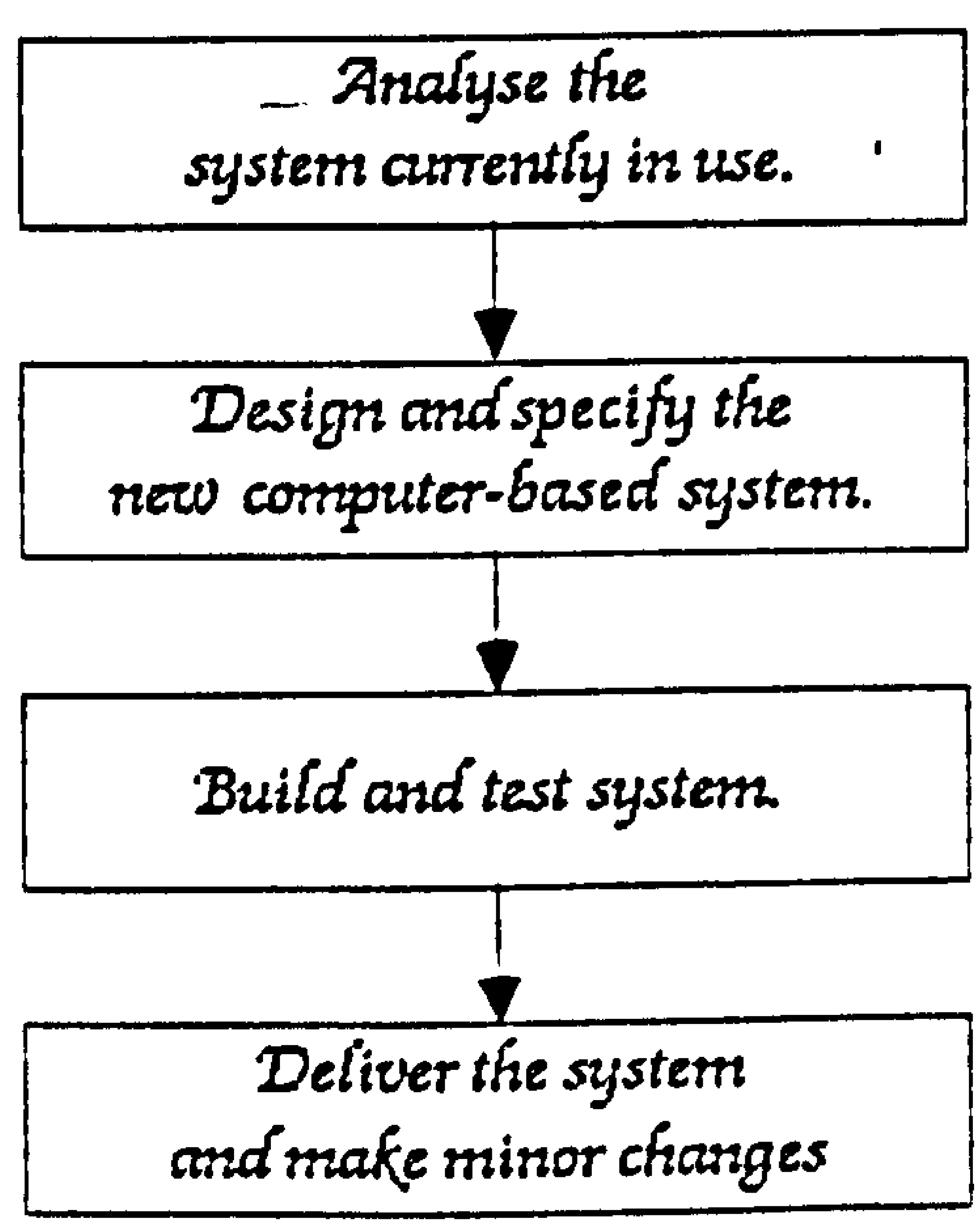


Figure 2.8: A simplified view of the design process.

The major problems with this traditional design process are; a) the information that is received about an organization does not provide adequate information about users, and b) designers are unable to properly understand the naive user, as was mentioned in the last chapter and in chapter one. The inadequacies of present design practice lead to the kind of problems mentioned in chapter one; users are required to remember too much information, systems are intolerant of minor errors, systems seem confusing to new users, interactional techniques are used inappropriately, systems do not provide the functions users require, systems force users to perform tasks in undesirable ways, and new systems cause unacceptable changes to the structure and practices of organizations.

The question that comes to mind when considering these problems is, why are present systems analysis and design methodologies failing to properly account for the user? One answer is that systems analysis techniques such as Structured Systems Analysis and Design Methodology (SSADM) are primarily concerned with information flow within a system. Although users are considered to some extent, the central focus is upon the system and the information it must deal with. Consequently, these types of methodologies have been labeled *system-centred.*

In essence, it is suggested within HCI research that present systems analysis and design methodologies are inadequate to meet the needs of users. Furthermore, it is argued that many system usability problems are a consequence of these inadequacies. But if this is the case, how can we engineer a shift from *system-centred* design to *user-centred* design? Furthermore, what is meant by the term *user-centred* design?

The answer to this last question is that the term *user-centred* design is a statement of intention; the desire to make the user the central focus of the design process. This aim is commonly shared within HCI research, and the real question is; how can we best achieve this? If we are to begin to move towards an answer to this question, then we may need to understand the major features of the design process a little better.

2. 4. 1 The characteristics of design

Carroll & Rosson (1985), using the earlier work of Carroll *et al* (1980), have suggested four aspects of design that essentially characterize it:

1 Design is a *process,* it is not a state and cannot be adequately represented statically.
2 The design process is *non-hierarchical,* neither strictly bottom-up nor top-down.
3 The process is *radicallytransformational,* involving the development of partial and interim solutions which may ultimately play no role in the final design.
4 Design intrinsically involves the *discovery of new goals.*

The essence of what Carroll & Rosson are trying to convey is that the design of a system evolves throughout the design process, a system is not simply specified and built. We might like to think of design beginning at a high level, where overall goals are chosen for the system. The rest of the design process might be thought of as a straightforward process of meeting these goals in the detail of design. However, in reality this does not occur.

At the beginning of the design process some of the low-level goals are known as well as some of the high-level goals. Throughout the design process, through compromises and tradeoffs, these goals build into a more complete and coherent picture, as goals are added, changed or discarded. Solutions to design problems often require creativity, and consequently the process of design, both in the emergence of goals and the development of solutions, cannot be described as completely rational or logical.

Other research into design has tended to produce results that generally agree with the characteristics outlined by Carroll & Rosson (1985). Rubinstein & Hersh (1984) for example, state that:

> "Design is seldom an orderly or linear process. System building occurs in a real world with constraints, interruptions, distractions, emotions, personalities, and politics. Any guidelines used for systems development must operate within these complex conditions." They go on to say, "Because design is an art as well as a science, it is never a completely rational process." (Rubinstein & Hersh, 1984).

This view of design has been further supported by the opinions designers express about their role in creating systems. Figure 2.9 shows a perspective on the design process that emerged from interviews with designers (Hammond *et al,* 1983a). Notice how the needs of the user are not represented explicitly in figure 2.9; the major focus is upon how to build the system, rather than how to match the system to the user's needs.

Hammond *et al* (1983a) found that the task analysis the designers were likely to perform was based around considering the logical structure of the task, and was not concerned with the user's view of the task. Furthermore, designers were particularly concerned with "clean" internal architectures and "clean" interfaces that were consistent across different parts of the system. The difficulty with this approach is that this desire to be "clean" may cause the designer to ignore the user's needs with respect to performing the tasks that are required.

Moreover, designers' *theories* of users tended to consist of broad generalizations about user behaviour, there did not appear to be any recognition that user behaviour might vary across tasks. However, it became clear to Hammond *et al* that designers made many of their design decisions based upon these inexact *theories*. Nevertheless, designers did recognize that their knowledge of users was inadequate, and complained that they had no appropriate means for acquiring information about a particular user group or task.

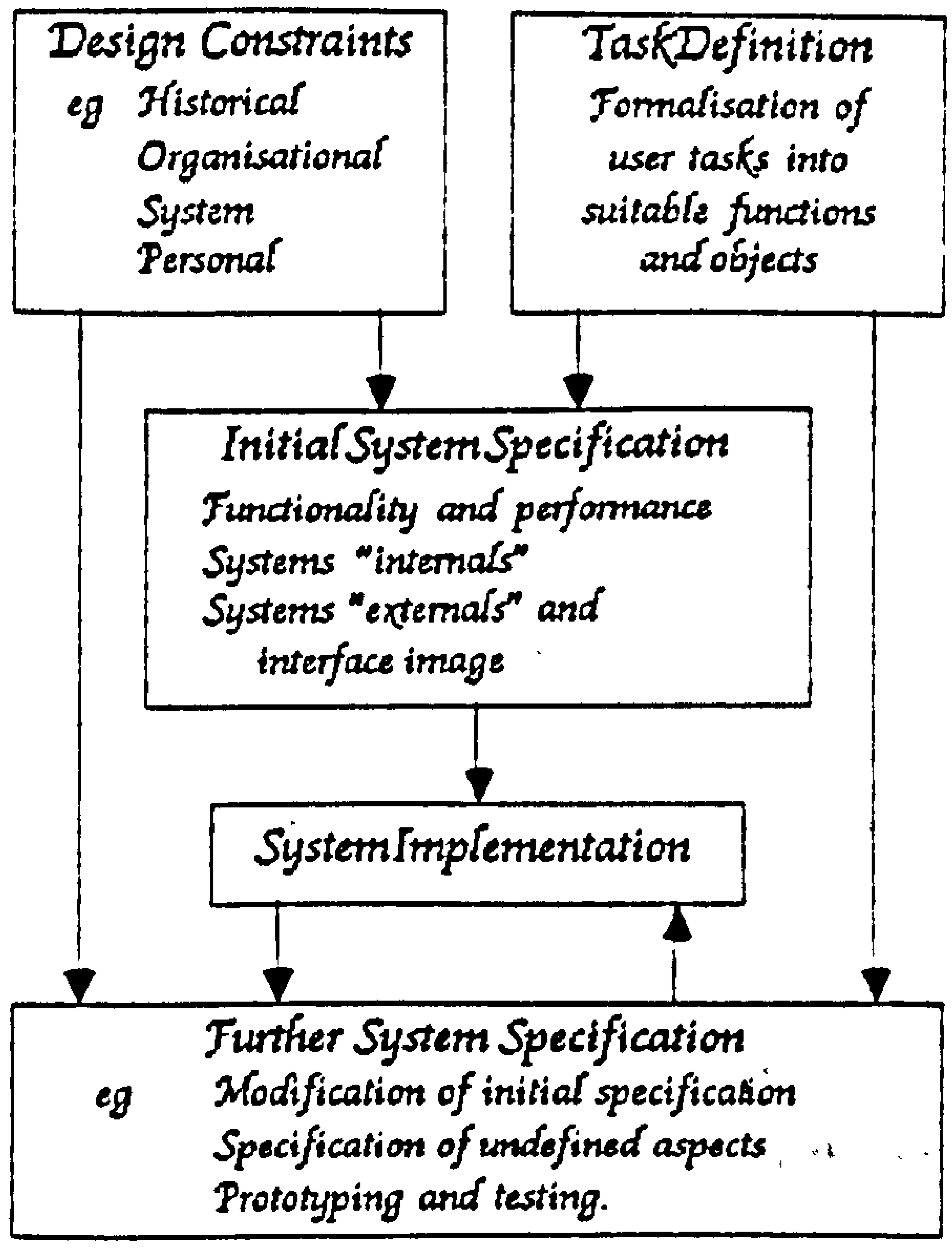


Figure 2.9: A simplified framework for the sequence of design steps based upon evidence obtained from interviews with designers. Taken directly from Hammond *et al* (1983a).

Although the work of Hammond *et al* (1983a) provides an interesting insight into how designers think about users and design, the designers that they interviewed appeared to be already committed to testing the product with users, and to the process of iterative design generally (iterative design is where a system is built, tested, and built again according to the results of the user tests). Some might argue that many designers do not acknowledge the role of the user within design to the same extent as Hammond *et al's* designers. In other words, the situation regarding the acceptance of human factors within design might be even worse than Hammond *et al's* interviews imply.

Nevertheless, Hammond *et al's* (1983a) work generally supports the view of design presented by Carroll & Rosson (1985). That is to say, that design is based upon the creativity and intuition of the designer. However, while creativity may be required within the design process, we might improve design if designers were better informed and did not have to rely upon their intuitions and guesses about tasks and users.

The question that emerges from this discussion of design is; how do we inform the designer so that:

1 the correct interactional device is chosen for any particular task;
2 the most useful dialogue style is chosen for any particular task;
3 anappropriateconceptual model is chosen so that a system can be easily learnt and understood;
4 users are allowed to perform a task in the way that they choose;
5 the information that users require is provided in the form that is acceptable;
6 the systems fits easily into the working practices of an organization?

There are three approaches to this problem, all of which stem from different disciplines; a cognitive approach, a mathematical approach, and an ergonomic or usability approach. The mathematical approach and parts of the cognitive approach employanalytical methods, while the usability approach is based around the notion of iterative design and informing the designer's intuition rather than providing formalisms. Mathematical formalisms for proving the logic and consistency of interfaces and systems will not be considered here. However, cognitive grammars that are intended to model the user's knowledge will be briefly discussed with respect to their role in design.

2. 4. 2 The role of cognitive grammars in the design process

Cognitive grammars, whether aimed at describing the logical or cognitive aspects of human-computeinteraction, appear to offer a means by which alternative designs can be analyzed. Within the literature, many of the discussions of these techniques have focused upon questions of whether a notation sufficiently describes a system, or accuratelycharacterizes human cognitive processes and representation. However, a question that is less frequently addressed is; can these methods be used in design?

In short, the answer to this question is not entirely clear, although many researchers are confident that usable grammars that accurately describe human-computer interaction will be developed sometime in the future. Nevertheless, it may be useful to consider some of the criticisms of cognitive grammars that have been advanced by various researchers.

The area where cognitive grammars have appeared to be most strongly criticized has been with respect to their usefulness in design. For example, it was noted in earlier in this chapter that none of the proposed modelling techniques or grammars directly involve the user. Therefore, there is a risk that these methods cannot detect when a user's conceptualization of the task is significantly different from that held by the designer or engineer who applies the grammar.

Moreover, Carroll & Rosson (1985) have argued that current cognitive grammars or modelling techniques do not fit easily into present design practice. They argue that;

> "..the formal evaluation of a given set of design specifications does not provide the kind of detailed qualitative information about learnability, usability, or acceptance that designers need in order to iteratively refine specifications, rather it assigns a figure...of merit to the design [i.e. the number of rules needed to describe part of the system]. This could be used to order a set of alternative designs, but contrasting alternative designs is an extremely inefficient means of converging on the best solution." Carroll & Rosson go on to say; "In a word, analytic approaches [modelling techniques/formal methods] do not support the process of design...The initial definition of design specifications can often rely on analytic methods. But this initial definition is only the beginning of the design process. Unfortunately, this is where the analytic approaches stall." (Carroll & Rosson, 1985)

In essence, Carroll & Rosson (1985) are arguing that current modelling techniques (oranalytic approaches) are almost irrelevant to the process of design. Of course, this does not necessarily mean that all future cognitive grammars are unlikely to be of use, only that present methods may have serious shortcomings.

2. 4. 3 The usability approach to the design process

The ultimate objective of the usability approach is the incorporation of human factors issues into the design process. In other words, this means making the user the centre of the design process rather than the system. The emphasis is upon creating computer systems that support the user within an organization, rather than the user supporting the system. Yet we would expect this to be the major objective of all of the approaches to human-computer interaction. The answer is that this goal is also shared by both cognitive and mathematical approaches to HCI, but that this objective has been most strongly stated by those pursuing a usability approach.

This usability approach begins with the aim of creating *socio-technical systems.* Quite simply, this means creating a technical system and an organizational system

(Eason, 1983b). A socio-technical design process is one where it is recognized that design is not just about creating a computer system, but also about engineering a new organizational system. In order to do this, present design practice needs to be re-directed towards a consideration of the user. That is to say, that we need to engineer a shift from system-centred to user-centred design. Damodaran (1983) has considered these two different types of design process, as well as the different stages between these two extremes. Figure 2.10 shows the five types of user involvement in design that Damodaran identifies.

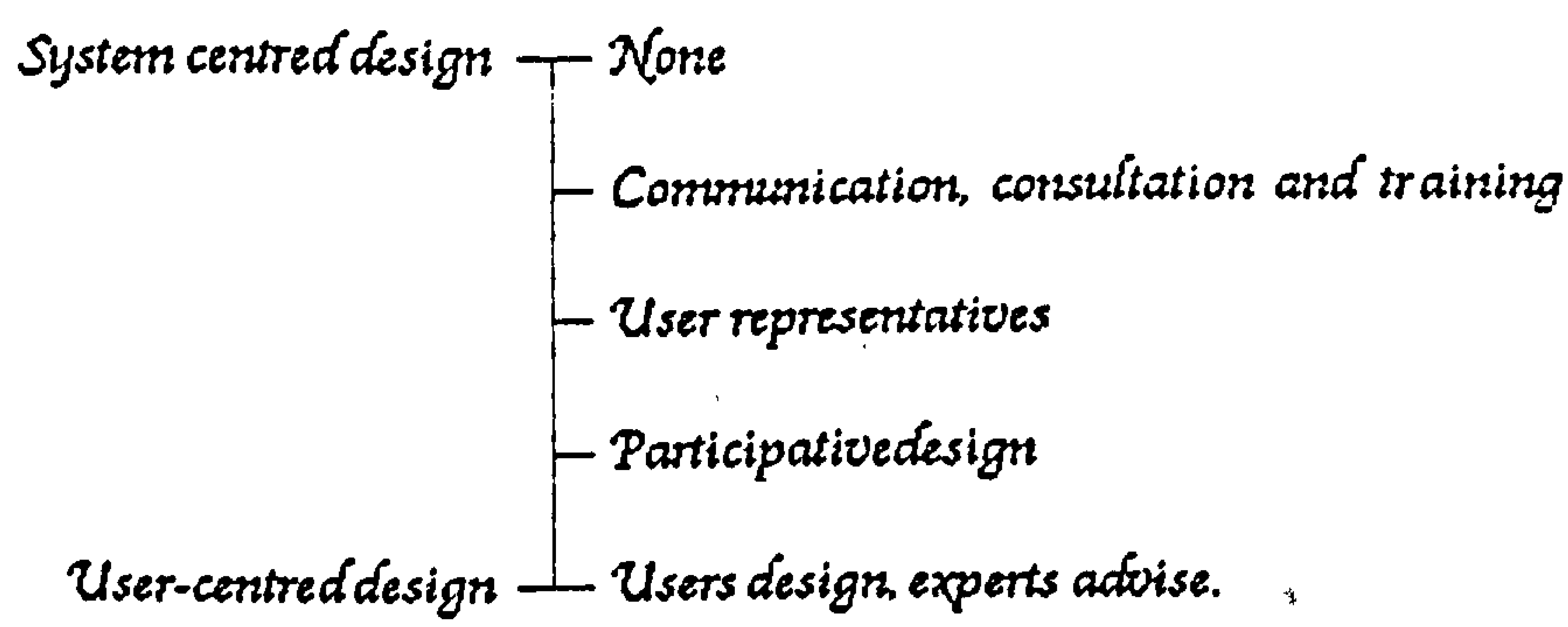


Figure 2.10: The five types of user involvement in design identified by Damodaran (1983).

As can be seen from figure 2.10, the form of design that is perceived as being closest to *user-centred* is where the users design the system themselves and experts only advise. These experts cannot be called "designers" as they are not longer designing a system. Whether designers might ever accept this radical approach is not an argument that will be fully entered into here. Although some might argue that user-centred design is a process of partnership between users and designers, rather than a simple handing over of responsibility and control.

Work such as Damodaran's (1983), where present design practice is considered as well as its alternatives, is central to the usability approach. These alternatives usually involve some form of *iterative design* and prototyping of systems. Iterative design is where a system is built as a prototype and then evaluated, using some of the methods discussed in the last chapter. The results of this evaluation are then fed back into design to produce the next *iteration* of the system. This design philosophy can be seen in the view of the design process that Rubinstein & Hersh (1984) present (see figure 2.11).

The problem with adopting this prototyping approach is that it requires time and effort to produce the prototypes. Many design teams argue that they do not have the time to build a prototype, and that building a system that is ultimately not going to be the final design is a waste of time. In other words, the argument against prototyping is largely an economic one. That is to say, that design teams cannot afford the time to develop and test their system, and then redesign it in the light of what users do and say.

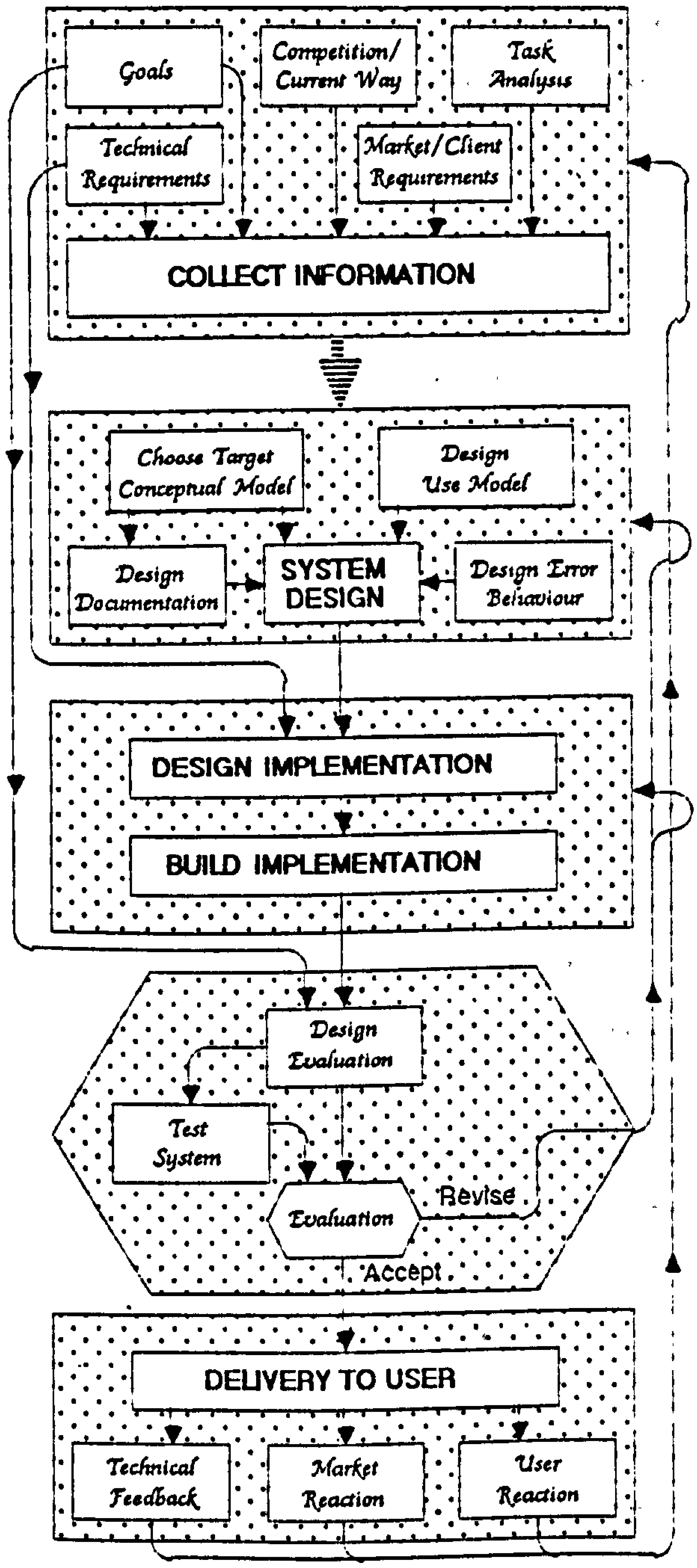


Figure 2.11: The elaborated view of the design process presented by Rubinstein & Hersh (1984).

However, usability difficulties often force designers to change their system anyway, albeit late in the design process, after the system has been trailed or released. These changes usually involve *patching* the software. That is to say, that carefully designed software, with its economic and elegant technical solutions, has to have chunks of programming code crudely inserted. The argument for an iterative approach that involves prototyping is that this sort of software patching could be avoided if systems were tested with users before their design was fully implemented.

In response to the criticism that prototyping is too time consuming, there has been a great deal of effort directed towards the development of prototypng tools and languages (e.g. MacLean *et al,* 1986; Richards *et al,* 1986). The idea behind these prototyping applications (systems that enable prototypes to be built) is that the dialogue for a system can be specified quickly and with the minimum of effort on the part of the designer. The central goal for these prototyping tools is not that they should be economical with memory space, nor that they should produce programs that are technically elegant, but that they should be quick and easy to use.

Iterativeprototyping can provide the design team with information on the usability of their proposed design, and the design of the system can be changed accordingly. Clark *et al* (1984) make just this point, and argue that developing prototypes for iterative design allows users to make a contribution to the design process. They state that prototyping is, "A means for end users to define, refine and re-define their own systems under their own control, within a systematic framework." This view has been widely supported within HCI (e.g. Bury, 1984; Wixon *et al,* 1983), and it appears as if the arguments for prototyping are gradually being accepted, at least at a corporate level within the large computer manufacturers such as IBM, DEC, Hewlett Packard etc.

A general iterative design process was outlined by Rubinstein & Hersh (1984) and can be seen in figure 2.11. Nevertheless, this is only a generalized diagram, and while it proscribes a process, it does not bring out the essential characteristics socio-technical design (the idea of designing both a technical and an organizational system). Shackel (1986) however, outlines five fundamental features that he argues the design process should possess to produce usable systems (see figure 2.12).

Shackel's Five Features of Design.

User centred design - focussed from the start on users and tasks.
Participative design - with users as members of the design team.
Experimental design - with formal user tests of usability in pilot trials, simulations and full prototype evaluations.
Iterative design - design, test & measure, and re-design as a regular cycle until results satisfy the usability specification.
User supportive design - training, selection (when appropriate) manuals, quick

reference cards, aid to 'local experts', 'help' systems, e.g. on-line: context specific, off-line: 'hot-line' phone service.

Figure 2.12: Shackel's five fundamental features of design for usability. This has been taken directly from figure 7 in Shackel's (1986) paper.

Shackel (1986) then suggests how these features are incorporated within the process of design and what should be considered during design (see figure 2.13).

Shackel's Usability Actions in the Stages of System Design.

System Design Stage	Usability Actions
Feasibility:	Define the range of users, tasks and environments to be covered. Do the proposals match the needs? Preliminary functions and operations analyses. Preliminary allocation of functions. Participative design - panel of users in the design team. Create and formalize the usability specification by defining user requirements and setting usability goals.
Research:	Studies, often experimental, of human capabilities re system operation concepts. Use pilot studies in the field to explore users' operational needs and to study possible effects upon organizational and social structure.
Development:	Detailed analyses of all functions, tasks and operations involving or affecting humans. Design all human factors aspects of equipment and workplaces. Specify all environmental issues. Use guidelines to assist as design ideas are developing. Check design ideas against available human dimension, behaviour and performance data. Test subsystem sections in initial evaluation trials with samples of likely users. Iterative design - use test results as basis for re-design, and test again. Propose selection criteria (if relevant); develop training scheme; provide for other forms of user support needed.
Prototype:	Extensive laboratory evaluation with samples of likely users. Full field trials with representative actual users in proper working environment. Iterative design.
Regular operation:	Provide for user support - provide training, encourage and aid 'local experts', arrange 'hot-line' for help, etc. Gather extensive evaluation data (both objective performance data and subjective attitude data); feed back the evaluation data as check on

decisions and predictions made during design; learn from the data - modify the design databases, models and methods for future use.

Figure 2.13: Shackel's list of activities in the stages of system design to improve usability. This has been taken directly from figure 8 in Shackel's (1986) paper.

2. 4. 4 The two approaches to the design process

The cognitive approach to the design process has concentrated upon cognitive grammars that analyse the cognitive complexity of a task. These methods also appear to offer the designer a potentially useful tool, although it is questionable whether these grammars can ever adequately provide the same benefits as allowing the user to participate in design.

At a more general level within HCI, the cognitive approach has provided interesting descriptions of some of the central problems that users face at the human-computer interface. For example, earlier, Norman's (1986, 1987) view of the gulfs of execution and evaluation in HCI was outlined. This sort of description has undoubtedly enabled a greater appreciation of the cognitive activities that are involved in communicating with a computer.

However, practitioners from industry might argue that, although such accounts are useful, the problems of human-computer interaction require more than just elegant descriptions. Yet at present, besides guidelines such as those suggested by Marshall *et al* (1987b), the cognitive approach does not appear to have a great deal to directly offer to the designer. Formal cognitive methods have been criticized for being unusable and for not fitting into the true process of design, while design methodologies that directly address the cognitive issues are only beginning to be developed.

The ergonomic or usability approach, on the other hand, is generally better established within industry. Some might argue that the usability approach to the design process has been the *practical* approach. However, the difficulty that this creates is that the usability approach does not offer a means by which we can understand human-computerinteraction.

Chapter 3

Cognitive Grammars and Cognitive Style

3. 1 Overview

The central issue addressed in this chapter is the question of whether predictive grammars offer the most promising means for modelling the user within the design and development process, particularly with respect to the development of IT systems for use within complex systems. Payne & Green's (1986; Payne, 1984) Task-Action Grammar is briefly described to provide a flavour of how such analytic methods are used, and seven criticisms of predictive grammars are offered which relate to their use within design.

These are that: it is difficult when using these methods to consistently employ the same grain of analysis across analyses, they assume consistent user behaviour, they may not expose actual mismatches between the user's and the designer's model of the task, they are not compatible with the design and development process, they require a clear view of the task to be modelled and this is not always available, they are too complex to use on an everyday basis, and many notations address only one aspect of interaction. Consequently, it is suggested that present predictive grammars are best considered as research tools, and as the forerunners of more practical tools for the early stages of design, rather than as techniques that are likely to have an immediate impact upon design and development.

An alternative approach to using predictive grammars for modelling the user is proposed, where aspects of user preferences and behaviour are predicted according to their position of cognitive style dimensions. Some of the literature regarding cognitive style is reviewed.

3. 2 Introduction

It is now generally accepted that present systems analysis and design methodologies do not adequately account for the user's (or operators) needs, and that many system usability problems are a consequence of these inadequacies. This problem is compounded within complex systems, where users or operators may have considerable problems understanding the system as a whole. One proposed solution to this problem has been to provide predictive grammars (also called analytic techniques) for analyzing the user's tasks. Consequently, the development of such methods has been the focus of much research within the HCI community. But what do we mean by the term *grammar?*

In its most straightforward sense a grammar is a notation for describing an aspect of a task, a system or user behaviour. The purpose of a grammar is to break-down a task or system into manageable units, as well as showing the relationships between these units. In this way inconsistencies and complexities can be exposed. These methods are predictive in the sense that they are used to analyse a task and predict some aspect of user behaviour at a task.

This approach has been characterized by the development of grammars (notations) that rely less upon design principles, but upon particular cognitive models of one or other aspect of cognition. That is to say, that a model of the user and of cognition is implicit within the constructs and conventions of the notation. Moran's (1981) Command Language Grammar (CLG) and Payne & Green's (1986) Task-Action Grammar (TAG) are both examples of notations that fall within this category. In essence, this sort of grammar is a form of user modelling; these methods address the question of how the user represents the task and system.

The aim of CLG is to describe the user's mental model of a system, while TAG is concerned with the cognitive complexity of performing a task. In other words, when a task or interface is described using the symbols and conventions of a grammar such as TAG the output (the number of rules describing the task) is considered to be representative of the cognitive complexity of performing the task.

At this point it may be useful to understand how these grammars work in a little more detail. Consequently, we will consider Payne & Green's (1986) Task-Action Grammar, as it is one of the more interesting grammars, and addresses the area of task-action mapping, which is viewed as being one of theoretical importance.

3. 3 The Use of a Cognitive Grammar

Payne & Green's (1986) TAG is concerned with the question of how we map our conceptual model of a system onto our actions towards that system. In short, it is a model of how we relate what we understand onto what we can do. Payne & Green describe the purposes of the model in the following way:

> "The central aim of TAG is to formalize [the mapping from the task level to the action level] in such a way that simple metrics over the grammar, such as the number of rules, will predict aspects of the psychological complexity of the mapping."

They go on to say:

> "A secondary aim of TAG is to help the analyst appreciate the structure of a task language." (Payne & Green, 1986).

3. 3. 1 The model underlying TAG

Within TAG the task language is described in terms of *simple tasks* and *rule schemata*. *Simple tasks,* according to Payne & Green, are tasks that are cognitively automated. That is to say, they are tasks that do not require conscious control. An example might be the actions that are required to drive a car. Changing gear, where the clutch is depressed, the gear lever moved, the clutch released and the accelerator (throttle) applied, is clearly complex, yet it is possible for the driver to engage in other activities simultaneously, such as conversation with a passenger. Within Payne & Green's model it is as if we have a great library of these automated tasks to draw upon in any given context.

The *simple tasks* are selected from this library according to their features. This selection occurs according to the *rule schemata*. The *rule schemata* are memory structures where the *simple tasks* are represented according to their features. Hence, Payne & Green's term *feature-tagged rule schemata*. These structures enable task descriptions to be mapped onto action specifications. In other words, *simple tasks* (represented in the rule schemata) are selected and ordered according to their features to provide an action specification. It may be helpful if we illustrate this point with a simple example.

Imagine that we have decided to make a cup of coffee. There is a vast array of actions or *simple tasks* that we can choose from. However, only one set of actions is likely to succeed. There are many related actions we might inadvertently choose, such as taking a tea-bag from the cupboard. Other simple tasks, such as selecting fourth gear in the car are clearly a long way from the making-a-cup-of-coffee set of actions. Indeed, one of the arguments that Payne & Green advance for this model is that they can show that making a cup of coffee is similar to making a cup of tea, and far-removed from driving a car. Choosing the correct set of actions to make a cup of coffee, in an appropriate order, is the responsibility of the *ruleschemata*, and Payne & Green (1986) suggest that the actions of these *rule schemata* can be modelled according to certain rules which relate to the grammar of TAG.

3. 3. 2 The grammar of TAG

A description of one aspect of the MacDraw™ system on the Apple Macintosh can be seen in figure 3.1. However, if we are to understand this description then we need to know a little about the MacDraw system.

To draw an ellipse on this system we must select the "ellipse" tool from the side of the window (see figure 3.1). We must then use the mouse to position the cursor on the screen, press the mouse button, and move the mouse to another position to create the ellipse. The size and shape of the ellipse is determined by the position to which the mouse is moved.

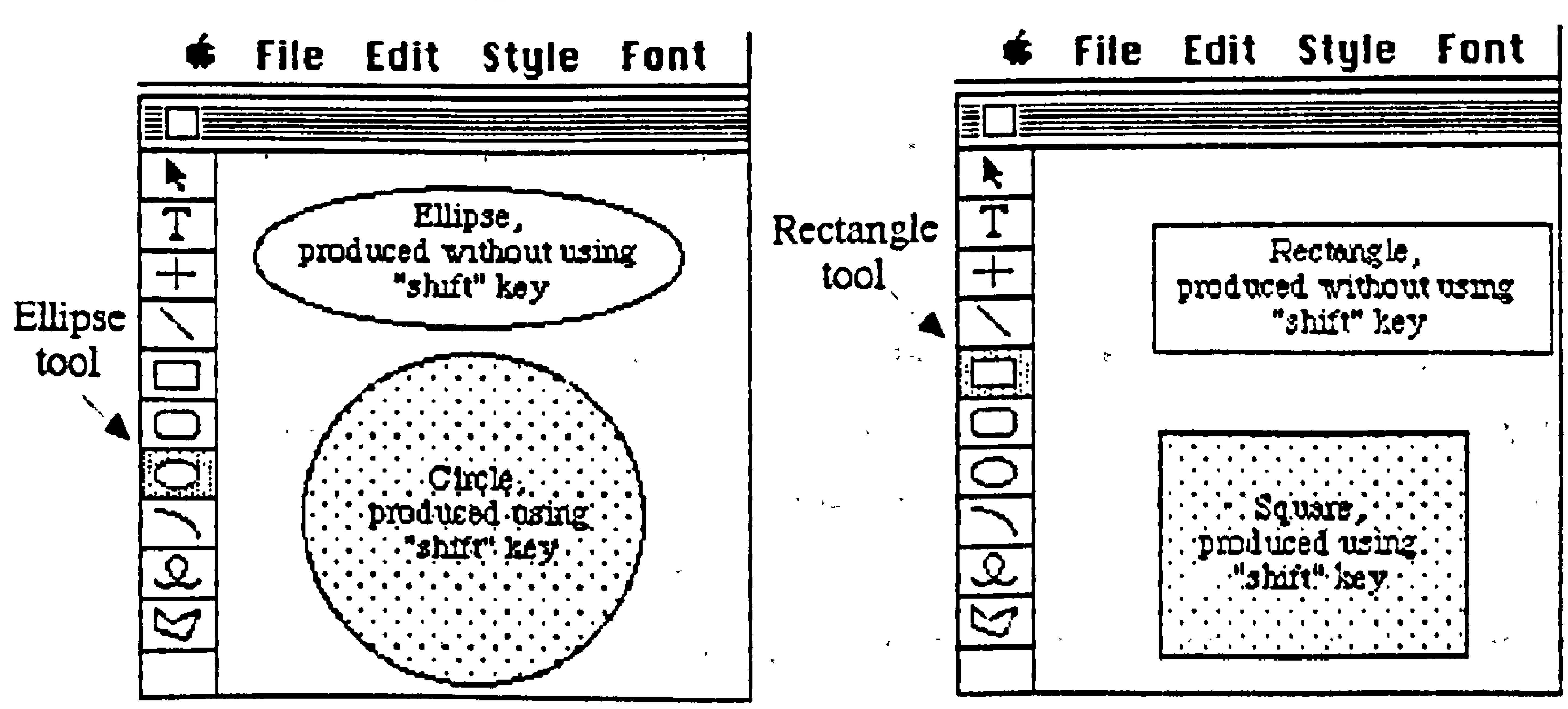


Figure 3.1: Two partial screens from the MacDraw™ system for the Apple Macintosh.

We might, however, decide to draw a circle. The MacDraw system treats a circle as a special-case ellipse. To draw a circle we must follow the same series of actions as before, but the "shift" key must be held down while the operation is executed. The series of actions for drawing a rectangle are the same as those for drawing an ellipse, except that the "rectangle" tool is selected instead of the "ellipse" tool. Again, to draw a square, which is treated as a special-case rectangle, the shift key must be held down as the operation is performed. Intuitively, it appears as though there is a consistency between these two operations, as drawing a square or a circle requires the use of the "shift" key. These actions are represented in the single rule of the TAG grammar shown in figure 3.2.

---

task [effect = insert, type of entity = ANY, special case = ANY]

—» special action [special case] + draw-object [type of entity]

draw object [type of entity] —» select tool [type of entity] + move mouse...

The "effect = insert" means that entities (rectangles or ellipses) are being added to the picture, not moved or deleted. The "type of entity = ANY" means that the rule stands for any entity, whether it is a rectangle, ellipse, or whatever else. The "special-action [special case]" means that if it is a special case (i.e. a square or circle) the special action is performed, which means holding down the "shift" key.

Figure 3.2: A rule from a TAG description of the MacDraw™ package for the Apple Macintosh. Original Source, Green (1987).

As Green (1987) points out, the compact representation, shown in figure 3.2, is only possible when the interface language is structured consistently and this structure fits the user's own representation, according to Payne & Green's (1986) model. In short, the number of rules that are required to describe a system is a guide to the cognitive complexity of that system, and the extent to which the system is likely to match the user's mental representation of the system. This, of course, assumes that Payne & Green's (1986) model is correct.

Overall, Payne & Green's TAG has a number of particularly attractive features, in that it appears to take account of a number of factors when assessing cognitive complexity, rather than just considering goals and subgoals (cf. Kieras & Polson, 1985). Furthermore, the model addresses an area that is seen as being of theoretical importance; that of task-action mapping (see Young, 1981). The strong theoretical base of TAG is undoubtedly one of its greatest assets, and the uses and advantages of user modelling techniques in general have been presented positively within the literature.

3. 3. 3 What do user cognitive grammars offer?

A number of uses of user modelling have been identified. These include:

- Matching the facilities that a system provides to the needs of the user.
- Suggesting metaphors to improve user learning.
- Guiding design decisions and making design choices and assumptions explicit.
- Guiding the design of experiments and helping in the interpretation of the results.
- As a predictive evaluation tool for proposed designs.
- As a means by which variations in the user population can be identified.

Some of the above claims might be better viewed as goals rather than achievements. For example, it is not clear how any of the existing grammars might be used to expose differences and variations in the user population. Nor do present techniques show us how to structure a system so that it appears simple and straightforward to some users, and progressively more complex for those who wish to exploit the system to its fullest capabilities.

Nevertheless, the advantage of using a formal notation is that a prototype does not have to be developed in order to evaluate the proposed system. Perhaps the most helpful aspect of any grammar is that it can show a design in a new and revealing light. Sufrin, although writing about Z notation, which is not based upon any cognitive model, makes just this point:

> "..it is only by understanding the essence of the purpose of an information system - abstracting from the details of any proposed implementation - that one can begin to judge the validity of design choices concerning the user-system interface." (Sufrin, 1986).

In other words, grammars can reduce a system to its basic elements, as well as exposing the logical relationships between these different elements. In essence, the advantage of using grammars is that they increase the amount, type and quality of information that is available when making a design decision. A grammar that has, as its basis, a cognitive model of some aspect of the user's cognition (a user modelling technique), supposedly has the further advantage that its results should reflect more accurately the cognitive complexity of performing a particular task in a particular way on a particular system, because the way in which its measurements (the number of rules) are constructed should more accurately mirror the user's representation.

3. 3. 4 The disadvantages of predictive grammars

Despite the advantages of grammars, particularly those that attempt to model the user, there have been criticisms of their proposed use within the design and development process. For the purposes of the argument here seven drawbacks to employing predictive grammars have been identified.

The grain of analysis. Firstly, there is the ever-present problem of the grain-of-analysis of a technique. For example, how, in practical terms, do we decide what is and what is not a *simple task* when using TAG? Is a *simple task* a single action such as unscrewing the lid to the coffee jar, or is it the whole sequence of actions from opening the cupboard door through to replacing the jar after the coffee has been placed in a cup? Clearly the answer is that it depends upon the extent to which sequences of actions have become automated, but how can we know what is and what is not automated?

This problem does not just have theoretical implications for the use of TAG, but also practical repercussions. For example, it appears quite possible that we might analyse one task assuming simple tasks of a particular size and then analyse another task and divide the simple tasks up in a different way. In other words, we cannot be consistent without knowing exactly what is and what is not cognitively automated, and there appears to be no practical way in which this information might be obtained.

Inconsistencies in the grain-of-analysis is an acute problem for grammars such as GOMS (Goals, Operators, Methods, and Selection rules; Card *et al.* 1980; 1983) and CCT (Cognitive Complexity Theory, Kieras & Polson, 1985). Kieras & Polson's CCT is a production system model based on the GOMS model. One response to the *grain-of-analysis* problem might be to provide guidelines as to how grain size should be determined during analysis. However, one of the arguments in favour of using cognitive grammars is that guidelines are difficult to interpret and apply, and are often disregardedduring design. Consequently, using guidelines to steer the analysis of a task using a grammar might be viewed as inefficient, as well as being something of an *own goal.*

User consistency. A second problem with predictive grammars is that they assume that the user is consistent from one situation, one task and one action to another. Yet much of the evidence from research into reasoning has demonstrated that our behaviour does not always conform to the rules of formal logic, but is based rather upon our knowledge of the domain in which our problem lies. Johnson-Laird (1981, 1983) for example, has demonstrated that if logically identical problems are presented in different knowledge domains then people produce different answers, according to the domain. There is no reason to believe that inconsistencies will not also occur within domains, as well as between them. Behaviour then, is based upon the knowledge the user possesses and brings to bear upon the task, it is not necessarily consistent from one task to the next. Consequently, techniques such as CCT (Kieras & Polson, 1985) appear to be based upon a false assumption. For example, Polson states that:

> "It is assumed that rules common to two tasks represent common elements and that these common rules can be incorporated in the representation of the second task in a training sequence at little or no extra cost in training time." Polson (1987).

This, then, is one of the problems of using a predictive grammar. Such methods rely upon formal consistency in behaviour, yet we are not always consistent from one task to the next, but base our behaviour upon knowledge we already possess about a particular problem area. As a consequence, models such as CCT can only be considered, at best, as approximate, and at worst as inaccurate. We can, however, only make rough guesses about the knowledge that any one individual or user group might possess. Even if we could chart all of the knowledge a user possesses, we cannot always know which areas of knowledge an individual will use in dealing with a new situation or task.

For example, if we return to the MacDraw system that was discussed earlier then we might remember that both a circle and a square were treated as special-case ellipses or rectangles respectively. To obtain these special cases (either a circle or a square) then the user had to hold down the "shift" key as the drawing operation was performed. As the "shift" key is held down for both special-case operations then this appears to be intuitively consistent. However, imagine that we have a user group who have previously used a different drawing system. They can draw squares without difficulty, by holding down the "shift" key. When these individuals try to draw a circle then they complain that the system is not consistent. They expect to have to hold down the "control" key to draw a circle, not the "shift" key. This, they claim, is quite consistent; "S" stands for "square" and "C" (or control) stands for "circle".

Consistency, then, depends upon our perspective and our knowledge of previous tasks and situations. With respect to formal logic, we are not consistent from one situation to the next. However, a system that requires the user to depress "shift" for a square and "control" for a circle might be viewed as consistent by some users, yet would not be considered so using a predictive grammar.

User involvement. A third problem with predictive grammars is that these techniques might show the designer what the user should need, working from a logical basis, but they do not inform the designer of what the user *actually* wants or needs from a system. Because the user is not involved they do not show where the user has a conceptualization of the task which is markedly different to that held by the designer.

The example given in the previous section might also be taken as an illustration of this related point. However, these are separate points, as the former is concerned with the assumption of consistency in behaviour, while this point is that predictive grammars do not necessarily expose the mismatches between the user's and the designer's models of the task. In other words, a grammar might demonstrate that an interface is consistent, but will not show where the user has a view of the task that is different to that held by the designer.

Compatibility with the design process. The extent to which predictive grammars might be of use within the design and development process has been the focus for the majority of the criticism of these techniques. The character of the design and development process has been summed up by Carroll & Rosson (1985):

1 Design is a *process,* it is not a state and cannot be adequately represented statically.
2 The design process is *non-hierarchical,* neither strictly bottom-up nor top-down.
3 The process is *radically transformational,* involving the development of partial and interim solutions which may ultimately play no role in the final design.
4 Design intrinsically involves the *discovery of new goals.*

In essence, design is not simply a simple process where a specification is agreed and then implemented. A system evolves throughout the design and development process via compromises and tradeoffs, and only gradually does a more complete and coherent picture emerge. Carroll & Rosson (1985) use this picture of design, as a semi-organized race against time and budget, to argue that current grammars do not fit easily into present design and development practice. They argue that:

> "..the formal evaluation of a given set of design specifications does not provide the kind of detailed qualitative information about learnability, usability, or acceptance that designers need in order to iteratively refine specifications, rather it assigns a figure...of merit to the design [i.e. the number of rules needed to describe part of the system]. This could be used to order a set of alternative designs, but contrasting alternative designs is an extremely inefficient means of converging on the best solution."

Carroll & Rosson go on to say:

> "In a word, analytic approaches [modelling techniques/grammars] do not support the process of design...The initial definition of design specifications can often rely on analytic methods. But this initial definition is only the beginning of the design process. Unfortunately, this is where the analytic approaches stall." (Carroll & Rosson, 1985).

The doubts expressed by Carroll & Rosson as to the usefulness of predictive grammars (or analytic techniques) appears to be supported by some recent research. Bellotti (1988) reports a study into the use of such techniques within commercial environments and concludes that some constraints upon design teams, such as their lack of autonomy, their poor access to user and task information and market pressures, are unavoidable. Furthermore, she suggests that:

> "..HCI DETs [Design and Evaluation Techniques, such as TAG and CLG], although potentially valuable to commercial design, are not applied in practice. The design environment conditions required for the successful application of current HCI DETs do not appear to be satisfied by commercial design projects. The reason for this is the existence of unavoidable constraints in commercial design which future HCI DETs should try to cater for." (Bellotti, 1988).

Understanding the task. Not only is it apparent that predictive grammars might not easily fit into design, but furthermore, the information that is required to use such a technique might not be available in the early stages of design, when their use might be of most benefit. Using a predictive grammar requires a clear view of the task that is to be analyzed. However, because of the way in which a design evolves, a clear view of how a task might be performed may not emerge until the system is beginning to be implemented. In an ideal world this should not be the case. However, in reality, with the pressures that are placed upon many design and development teams, this form of development is viewed by many as inevitable (see Bellotti, 1988).

Practical complexity. Although the conventions and meanings of many notations are abundantly clear to those who invent them, almost all predictive grammars have been criticized for being too complex and time-consuming to use. Moran (1986) has answered the criticism regarding the time and effort required to use a grammar by suggesting that such models should only be applied to small aspects of the design. However, this leaves the question of how to select those small parts of a design that require attention. Furthermore, the complexity of these methods remains an unanswered criticism, with Bellotti (1988) reporting that designers found such methods: "..intimidating in their complexity."

It appears as though present predictive grammars are too complex to understand and use on an everyday basis. Using these methods to describe even the interface to a cash dispenser at a bank or building society has often proved too difficult for those new to such grammars. Given these problems, there might be some doubts as to whether even the authors of these grammars might be able to describe a complex system using some of the notations that are currently offered as design tools.

Theareas addressed by a method. One argument in defense of any one predictive grammar might be that, despite its evident complexity, the information it provides justifies the time and effort required to learn and use it. This might imply that only one method will be needed during design. However, most grammars address only one area of concern. For example, the GOMS model and CCT are essentially performance models, while Payne & Green's (1986) TAG deals with only one aspect of cognition; that of task-action mapping. Although TAG may deal with an important aspect of human-computerinteraction, it addresses, nevertheless, only one of the gulfs of interaction described by Norman (1986). Moreover, it addresses only *one aspect* of this gulf; it does not address the identification of goals, planning issues, etc. In essence, the argument is this; if even one grammar is intimidating in its complexity how can we expect members of a design team to learn and use several complex formalisms? Furthermore, a sound theoretical knowledge of these techniques might also be required in order that the design team might be able to choose the formalism that best suit their needs at any particular time.

3. 3. 5 The role of predictive grammars

The seven drawbacks to using predictive grammars within the design and development process might imply that predictive grammars have no place within human-computer interaction. However, these are criticisms of present methods, and need to be considered against their suggested uses and advantages. As the design and development process changes in character over the coming years, and more sophisticated, but usable, predictive grammars are developed with accompanying software support, it is possible that such methods may have a permanent place in the design itinerary. Certainly, approximate predictive techniques have a place if used judiciously. Although such techniques might not be able to predict all aspects of user behaviour, they might, nevertheless, make a contribution in steering the early stages of design in the correct direction. Their shortcomings might be compensated for in the later stages of an iterative design process. Nevertheless, it is difficult to accept that such grammars can replace testing systems with users, as some have claimed (i.e. Polson, 1987). For the present, however, these techniques do not appear to have a practical place within design and development. But what is their current role within HCI?

Firstly, they are the possible forerunners of methods that will be of practical use within design (as has just been suggested). Secondly, they can be considered as research tools (see Booth, 1989); they are a means by which we can test cognitive models of the various aspects of interaction, such as task-action mapping. In short, the empirical validation of the predictions of a formalism can be used as an indicator of the advantages and disadvantages of the model of interaction embedded within the notation. In this second respect, grammars such as TAG are clearly advances on techniques such as CCT, in that TAG provides a model (simple tasks and rule schemata) around which we can base our thinking about interaction and the performance of a task. CCT, for example, cannot tell us anything about errors, and assumes error-free performance. Although some might claim that errors cannot be modelled, this is clearly not the case. TAG provides some idea of how errors occur (i.e. inserting an incorrect simple task into a sequence of actions, or omitting a correct simple task), while Marsden (1989) has developed a computational model of human decision-making, based upon Reason's (1989) Underspecification Theory. This model exhibits error forms that correlate highly with those exhibited by human subjects.

Models such as GOMS, Command Language Grammar (CLG, Moran, 1981) and CCT have been used as stepping stones en route to grammars that more accurately reflect cognition. TAG might be viewed as one of these developments. Nevertheless, TAG has its faults, as Payne & Green (1986) will admit. It would be wrong, therefore, to view any of the present grammars as end-points, or as new paradigms, as some have claimed. At best, they can only be considered as vehicles for the development of more useful and usable grammars.

3. 4 Cognitive Style

Having discussed the role of cognitive grammars, it might be useful to consider the potential role of an alternative; cognitive style. Moreover, the two cognitive style measures that are the subject of the first experiment will be outlined. In essence, the argument is this: if we cannot use cognitive grammars to predict user behaviour and preferences, what can we use? Cognitive style measures have been of interest within psychology for many years, and they might provide a means by which we might understand some aspects of user behaviour and preferences.

It follows from what has already discussed in chapter 2, that users who have differing task perceptions and expectancies might reasonably be expected to behave differently. If these users could be differentiated to any significant degree then different interfaces, which would incorporate different models of the task, so matching those of the different user groups, might prove to be a realistic possibility. First, however, we need to consider the question of the nature of the differences between users and groups of users. Following this, we need to address the question of how, in a practical sense, users can be differentiated.

3. 4. 1 Occupational groups and individual differences

Within any one business system there will be members from different occupational groups using the same computer system. However, all will have different occupational experience and training and the;

> "..way the person expects to perform his job (his role expectations) plays a large part in determining the kinds of behaviour he will accept when using a computer system. [Furthermore]...people who occupy similar jobs, e.g. bank managers, will have certain characteristics in common as computer users." (Eason, 1976).

Users from different occupational groups will not only have different perceptions of a task and different task models, they will also have different perceptions and different models of the overall computer and business system. This claim is partially supported by Hiltz (1980) who found that the perceived effectiveness of a system varied from group to group and that the way a person viewed and acted towards a computer system was a group related phenomenon. One conclusion that may be drawn from this is that, as a result of the different task models adopted by each occupational group, different interfaces may be required, with each interface reflecting one particular task model.

However, occupational experience and training might not be the only factor that affects the user's model of the task and system and interface requirements. Witkin *et al* (1977) have shown that what people choose as their academic subjects and possibly their careers can be related to their cognitive style. Furthermore, according to Witkin *et al*, those individuals who opt for academic subjects that do not appear to suit their cognitive style either perform poorly, give up before finishing their course, or fail. Cognitive style will be explained in more detail in the next section, but for the moment cognitive style can be taken to be a robust measure of cognitive individual difference. Theimportance of Witkin *et al's* finding is not that academic preference and career choice are correlated with one measure of cognitive style, but that it demonstrates the likelihood of consistent differences in cognitive functioning between the members of differentoccupational groups.

It is already accepted that the members of different occupational groups will differ in their experience and their training. However, in addition to this, they may show differences in the way that they perceive and process information. From what is already known about individual differences in cognition it may be possible, with some experimentation, to expand upon previous findings and move towards a state where the interface preferences and requirements of a particular occupational group can be predicted, or at least understood to some useful extent, in terms of their characteristic mode of cognitive functioning.

To summarize, individual differences in cognition may be important to HCI because users may differ in their interface preferences and requirements. Users appear to differ, not only in their experience and training, but also in the way that they perceive and process information. In addition to this, individual differences in cognition can be associated with particular generic professional groups and so it could be possible to describe some occupational differences in terms of individual differences in cognitive functioning. The initial objective of this research then, is to assess the extent to which it is possible to use what is known about individual differences in cognition to predict and recommend both interface dialogues and display format preferences for different occupationalgroups.

3. 4. 2 The psychological study of individual differences

It was German empirical psychology (e.g. Wundt, 1907) that prompted Galton

(1887) to investigate intelligence and mental deficiency. Although Galton is considered as the 'father' of the mental ability test, it was Binet who proved most successful in this field. Frustration with the inability of the empirical approach to analyse complex mental processes led Binet to develop tests and measures of ability in order to predict academic performance. Binet's success in turn, led to an emphasis within certain parts of psychology upon the study of individual differences and the complex mental processes which give rise to them. Of the many thousands of cognitive tests that have been developed and applied, it has been the measures of cognitive style that have aroused most interest in recent years amongst psychologists, but what is cognitive style and how does it differ from cognitive ability?

Cognitive styles research began with Adorno's consideration of authoritarianism (1950) and Witkin's investigation into the 'perception of the upright' (1950). Witkin's work culminated in the development of the field-dependent/independent cognitive style dimension and its measures (the Embedded Figures Test (EFT) and the Group Embedded Figures Test (GEFT)).

According to Messick (1976) cognitive styles are "characteristic modes of mental functioning" and "consistent individual differences in organizing and processing information". Ragan *et al* (1976) take a similar view defining cognitive styles as "psychological dimensions which represent consistencies in an individual's manner of acquiring and processing information". Robertson (1982, 1985) considers cognitive style as consistent individual differences in the strategies people use for selecting and processing information. Cognitive styles might be viewed as high-level heuristics which control lower-level strategies that deal with complex processes such as problem-solving or learning. As Messick (1976) explains:

> "..it is important to distinguish cognitive styles, which are high level heuristics that organize and control behaviour across a wide variety of situations, from cognitive strategies, which are decision-making regularities in information processing that at least in part are a function of the conditions of particular situations." (Messick, 1976).

According to this view, unlike cognitive styles, cognitive strategies are selected by the individual and are amenable to change. Furthermore cognitive strategies are situation specific while cognitive style can be viewed as the tendency to choose certain strategies in particular situations. Cognitive style might be considered as a consistent series of individual biases in selecting information processing strategies. In addition to this, there is a view held by some that cognitive style cannot be directly measured and that tests of cognitive style measure only the resultant manifestations of cognitive style in cognitive strategy.

Support for the view that an individual's cognitive style is expressed through his or her choice of cognitive strategies is provided by Bruner *et al* (1956) who showed individual differences in problem-solving strategies. Cognitive style then. might be thought of as consistent individual differences in selecting cognitive strategies, but what difference is there between this and cognitive abilities? What distinguishes cognitive styles from cognitive abilities?

According to Robertson (1982, 1985) and Messick (1976) this is an important question. The answer that appears to be generally accepted is that measures of ability consider content and are quantitative, whereas cognitive style deals with processing and its measures are qualitative. Moreover, abilities are considered to be unipolar whereas styles are bipolar. Messick (1976) explains cognitive abilities as dealing with questions of *what* (e.g. what individuals understand, what a person knows etcetra) while cognitive style is concerned with questions of *how* (e.g. how people process information).

A further question that might be asked about cognitive styles is; where do they come from, are they genetic or do they develop? A comprehensive answer has never been provided. However, Messick (1976) suggests that cognitive styles are 'habitual modes of information processing' which develop 'slowly and experientially'. Furthermore, Witkin (1976) states that socialization is of 'overwhelming importance in the development of individual differences'. In addition, Witkin suggests that genetic factors may be implicated. Although the suggestion refers particularly to the field-dependence cognitive style dimension, his comments have implications for all cognitive style dimensions.

To summarize, cognitive style is viewed as a set of dimensions which describe consistent individual differences in the selection of information processing strategies. Cognitive strategies differ from cognitive styles in that they are situation specific, can be changed and are selected by the individual. Cognitive ability tests differ from cognitive style measures in that they deal with questions of content, are quantitative and unipolar, in contrast to cognitive styles which are bipolar, qualitative and deal with questions which relate to how individuals process information. Where cognitive style originates is not completely clear and not always considered as important, except to say that it is almost certainly the product of the socialization of an individual and may also, to some extent, be influenced by genetic factors.

Although cognitive styles are conceptualised in terms of information processing, this is not the way that measures of cognitive style have always been used. Many studies consider correlations between various activities, professions or skills without considering these results in information processing terms. As a result there is a catalogue of unexplained relationships and correlations (McKenna, 1984; Robertson, 1982, 1985). This situation has prompted Lewis (1976) to call for research to focus upon individual differences which are basic and fundamental. In other words; there is currently a need for a unified explanatory theory of cognitive style. This problem and the difficulties in applying cognitive style to 'real world environments' will be addressed in more detail in a section towards the end of this chapter. For the moment this problem need only be borne in mind when considering the literature relevant to the two cognitive style dimensions reviewed here.

3. 4. 3 Two cognitive style measures

Only the two cognitive style dimensions considered relevant to the first experimental study will be reviewed here. For a in-depth discussion of cognitive style and its different dimensions see Messick (1976) and Ragan *et al* (1979). How these measures might be of use in the investigation will be dealt with in the next chapter, this section will be restricted to a description of previous research and the measures themselves.

The Conceptual Tempo cognitive style dimension: An aspect of individual differences with important implications for decision-making first noted by Kagan *et al* (1964) is the *conceptual tempo* cognitive style. Kagan (1966 & Kagan *et al* 1966) found that children, when dealing with a problem which involved a degree of response uncertainty, could be classified as either *impulsive* or *reflective.* An *impulsive* individual typically has short response times and makes a greater number of errors while a *reflective* individual takes longer to reach a decision and makes fewer errors. Kagan (1966) defines conceptual tempo in the following way:

> "The reflection-impulsivity dimension describes the degree to which a subject reflects upon the differential validity of alternative solution hypotheses in a situation where many response possibilities are available simultaneously. In these problem situations the subjects with fast tempo impulsively report the first hypotheses that occurs to them, and this response is typically incorrect. The reflective subject on the other hand, delays a long time before reporting a solution hypotheses and is usually correct." (Kagan, 1966).

Kagan suggests that the reflectivity-impulsivity cognitive style dimension exerts its influence at two points during the decision process. The first is during the selection of hypotheses and the second is during the stage where the solution hypothesis is being validated (Kagan *et al*, 1966).

The conceptual tempo cognitive style has been shown to be stable across time and across tasks (Ragan *et al*, 1979). However, Kogan (1971) has shown that although conceptual tempo is stable over time (e.g. from one year to the next) there is a developmental effect for children from five to eleven years old. Response latency (time to take a decision) increases and the number of errors decreases with age. Despite this tendency to become more reflective with age, the effect is not dramatic and it has been suggested that individuals who might be classified as impulsive on this dimension could be at a disadvantage, both educationally and in other respects. For this reason there have been attempts to alter the decision-making behaviour of impulsive individuals. The first attempt was made by Kagan, Pearson & Welch (1966) who succeeded only in increasing the decision time. The number of errors, the most important element, did not decrease. A later attempt by Yando & Kagan (1968) produced the same effect. However, Drake (1970) found differences in scanning strategy for the Matching Familiar Figures (MFF) test; used by Kagan to classify impulsives and reflectives) between impulsives and reflectives. Furthermore, Heider (1971) showed that by teaching impulsives the appropriate scanning strategy the number of errors made during the MFF test could be reduced.

Although Heider's result is interesting, there is no indication that the impulsives cognitive style was altered. It appears likely that only the impulsives cognitive strategy was altered as cognitive strategies are task specific and amenable to change. Clearly, training an impulsive individual how to search for information in every situation is not a realistic possibility, but what does this result indicate about the possible nature of the conceptual tempo cognitive style?

Kagan's explanation of the difference between individuals at either ends of the conceptualtempo cognitive style dimension was to suggest that anxiety about possible errors had a greater effect upon reflectives than upon impulsives. Few studies support this hypothesis (see Sato, 1983). For example, Achenbach (1969), using tasks involving analogical reasoning, showed that impulsives provided answers more often than reflectives, yet were also more often incorrect. This, together with other evidence, suggests that the difference between impulsives and reflectives is less to do with immediate perceived social situations (i.e. anxiety over incorrect decisions) and more likely reflects a difference in individual cognitive decision-making processes. To some extent this view is implied in Messick's (1976) definition of conceptual tempo:

> "[Conceptual tempo]..involves individual consistencies in the speed and adequacy with which alternative hypotheses are formulated and information processed." (Messick, 1976).

Conceptual tempo is then, a cognitive style in the true sense of the term. That is to say that, it manifests itself in the strategies an individual adopts in selecting and processing information and appears unalterable, with only its associated strategies amenable to change. Furthermore, it has been suggested that this cognitive style may be related to *cognitive control* in human information processing (Messick, 1976). The thoroughness with which an individual selects and examines the validity of alternative solution hypotheses may depend upon the cognitive processes employed. Which processes are employed and which are not is determined, according to conventional thought, by the control mechanism within the human information processing system. Therefore, there remains the possibility that the conceptual tempo cognitive style represents differences in cognitive control between individuals. This is however, a theoretical question and not an issue that is addressed here. An issue which is more relevant to the research is the question of the reliability of the test used to measure conceptual tempo (the matching familiar figures test).

Few within the cognitive styles research field doubt the validity of the conceptual tempo cognitive style, but unfortunately the MFF test is not completely proven. Kagan (1965), reported the reliability of the MFF test to be low to moderate. However, in the absence of other test tools this test might still provide useful data in an experimental situation, but its unreliability needs to be borne in mind.

The Visualiser-Verbaliser cognitive style dimension: The purpose of this section is not to detail all of the findings and theories which relate to visual and verbal models of processing and storing information, but to briefly describe a measure of cognitive style developed by Richardson (1977). This section is based entirely around Richardson's paper (1977). Firstly however, some of the evidence that first prompted Richardson to attempt to develop the Verbaliser-Visualiser Questionnaire (VVQ) needs to be considered.

Griffitts (1927) noticed that in verbal accounts of problem-solving, subjects appeared to think in either a *concrete visual* style or *auditory-motorverbal* style. Two years later Golla & Antonovitch (1929) published an account of differences in breathing patterns between verbal and visual thinkers. This last finding has been supported by many other studies, but most notably by Short (1953) who used an electric thermocouple beneath subjects' nostrils to record breathing patterns. The cause of this relationship has never been fully understood. However, it is suggested that the irregular breathingpatterns of verbalisers during problem-solving tasks is related to movements of the tongue and larynx which are assumed to accompany verbal thought processes (see Richardson, 1977). Furthermore, the evidence for differences in verbal and visual processing has not been confined to physiological associations. Bartlett (1932), while investigating perception and memory processes, noticed differences in the response strategies employed by subjects. Moreover, Roe (1951) noted consistent verbal and visual differences in the way that eminent research scientists from different subject areas conceptualizedproblems.

Although consistent individual differences in verbal and visual modes of processing have been shown to exist, the question of the measurement of these tendencies has proved to be something of a problem. Kocel *et al* (1972) have found that verbal and visual differences are associated with lateral eye movements. Verbalisers consistency turn their eyes to the right and visualisers tend to turn their eyes to their left. Kocel *et al's* finding has been supported by Bakan & Strayer (1973) who have confirmed the habitual nature of lateral eye movements. This tendency has been used by Richardson (1977) to pick out discriminating questions from Pavio's (1971) Ways Of Thinking (WOT) questionnaire in order to construct a short questionnaire to discriminate verbalisers from visualisers; the VVQ (verbaliser-visualiser questionnaire). Although the questionnaire is based upon eye movement studies, it has been shown by Richardson (1977) to be stable over time and to correlate to the findings of the breathing studies. Richardson (page 124, 1977) concludes:

> "The VVQ has been found to provide a stable index of an individual's cognitive style which can be used to predict theoretically relevant events of an experiential, behavioural and physiological kind. It can be concluded that a useful research tool has been constructed; it may be employed with reasonable confidence in the study of many problems concerned with the sequential and parallel processing of cognitive events." (Richardson, 1977).

Richardson (1977) notes the association of verbal and visual strategies to the cognitive functioning of the left and right hemispheres of the brain. Bradshaw & Nettleton (1981) however, argue that the relationship between visual and verbal processing and specialization in hemispheric functioning is not a simple one, but that the integrated functioning of both hemispheres involved in processing both verbal and visual information. This does not mean that there is not such a cognitive style as verbaliser-visualiser, only that the relationship between the verbaliser-visualiser cognitive style and hemispheric functioning is a complex one. Nevertheless, for the purpose of this research, the VVQ may be considered as a potentially useful indicator of cognitive strategy and information presentation preference at the interface.

3. 4. 4 The usefulness of cognitive style

Some of the history and the literature relating to cognitive style has been described. The purpose of this was to set the scene for the first study; an experiment which investigates the potential usefulness of cognitive style. To this end, the review presented in this chapter should be considered only as background to the more important arguments that are dealt with in the next chapter.

Chapter 4

The First Study: Information Presentation Format and Cognitive Style

4. 1 Overview

In the previous chapter the literature regarding the visualiser-verbaliser and conceptual tempo cognitive style dimensions was reviewed, and it was suggested that these two measures of individual difference might be of use in predicting some aspects of user behaviour and preference. In this chapter the first experimental study is reported. This study was designed to examine the potential usefulness of the two cognitive style dimensions with respect to human-computer interaction. It was suggested that verbalisers would, on the whole, express a preference for information in a tabular format, while visualizers would prefer information in a graphical format. Furthermore, during the experimental task, verbalisers were expected to more often use information in a tabular rather than a graphical format, while visualisers were expected to prefer and use information in a graphical format. Impulsive (on the conceptual tempo dimension) visualisers and verbalisers were expected to show greater preference for information in their predicted format (either graphical or tabular) than reflective subjects.

The experiment involved the administering of the visualiser-verbaliser questionnaire and the matching familiar figures test (for conceptual tempo) followed by the business-related experimental task. Subjects from both business and non-business backgrounds were included in the experiment in order that any effects due to business training and experience might be observed.

It was found that the visualiser-verbaliser and conceptual tempo cognitive style dimensions were not accurate predictors of either display format preference nor information use as a consequence of display format. There were, however, differences between business and non-business subjects. Subjects from a business background appeared to use both a different vocabulary and different information selection strategies to non-business subjects.

The results of the experiment appear to cast doubt upon the utility of the two cognitive style dimensions in predicting either preferences or behaviour at the human-computer interface. Furthermore, there were indications that other factors such as a user's experience of a relevant knowledge domain might be more important in influencing behaviour at a task than individual difference effects such as cognitive style. Overall, the results appeared to suggest that behaviour at the human-computer interface may be determined by a number of factors, that may interact in potentially complex ways, rather than any single over-riding factor.

4. 2 Introduction

It has been suggested in the literature that cognitive style may be of use in predicting some aspects of user behaviour and preference at the human-computer interface (Robertson, 1982, 1985; Fowler *et al*, 1985). The purpose of the experiment reported here was to examine the effects of cognitive style, measured on two dimensions, upon display format preference and information use during a business-related task. In the following sections the rationale behind the experiment is explained and the results presented. Firstly, however, the literature relating to information display formats will be briefly considered.

4. 2. 1 Display format.

The initial objective of the research, as stated in chapter one, was to explore the potential usefulness of two cognitive style measures with respect to human-computer interaction research and design. The experiment reported in this chapter was designed to investigate whether the visualiser-verbaliser and conceptual tempo cognitive style tests (the visualiser-verbaliser questionnaire and the matching familiar figures test; the VVQ and the MFF respectively) could be used to predict aspects of individual preference and behaviour at the human-computer interface.

Many commercial software packages aimed at the business community advertise graphical data presentation as a selling point. However, the general growth in the use of information display techniques in IT systems has not been matched by a proper consideration of their effects (Wickens & Kramer, 1985), and at the moment, the evidence regarding display format preference is somewhat contradictory. Benbasat & Schroeder (1977) have found that graphically displayed information was preferred by operationsmanagement students in an experiment where they had to play the role of a warehouse manager. Lucas (1979) on the other hand, discovered that subjects in a study where they were required to act as a buyer for a company, found data presented in a tabular format more helpful. One possible explanation for these apparently contradictory results is that the usefulness of a particular display format may be dependent upon the perceived requirements of a particular task or situation. Powers *et al* (1984) for example, have demonstrated that tabular displays increase the speed of performance while combinations of both graphical and tabular displays slow performance, but increase accuracy and comprehension on the part of the subject. Nevertheless, it remains difficult to reconcile the conflicting evidence regarding the usefulness of graphical and tabular display formats and alternative explanations (i.e. that preferences may be related to cognitive style) have not been fully explored.

It was hoped that the experiment would help to clarify the situation by demonstrating, if possible, that display format preferences could be related to subjects' positions on the visualiser-verbaliser cognitive style dimension. The possibility of using cognitive style to predict format preferences was prompted originally by Fowler *et al* (1985) who showed that dialogue style preferences could be related to an individuals position on the field-dependent/independent dimension.

It was suggested that verbalisers would prefer and use information displayed in a tabular format while visualisers were expected to be inclined to favour information displayed in a graphical format. A further possibility was that individuals may not only prefer information in one format or another, but might pay more attention to information in their preferred format (as determined by the individual's position on the visualiser-verbaliser cognitive style dimension). This possibility might also be related to how individuals scan information. Some people search carefully through information (reflective individuals) while others are more impulsive and reach their decisions in a shorter space of time, often without considering all of the information. Therefore, it was suggested that those individuals classified as impulsive on the conceptual tempo (impulsivity-reflectivity) cognitive style dimension may only consider information displayed in their preferred format while reflective subjects might consider information in both preferred and non-preferred formats. However, before we go further, it may be useful to briefly reconsider and outline the concept of cognitive style and the two dimensions used in this study.

4. 2. 2 Business knowledge

The possibility arose that subjects with experience in business, through their training and experience, may have inadvertently been taught to deal with information in a particular format. No evidence for this view had been found in the literature, but still the possibility was considered important enough to warrant serious consideration. Therefore, in order to observe these effects, should there have been any, subjects in the experiment were not only tested to determine their cognitive style, but were also selected from both business and non-business backgrounds.

4. 2. 3 Hypotheses

1 Subjects' stated information display format preferences should reflect their score on

the visualiser-verbaliser cognitive style test (i.e. a visualiser for example should preferinformation in a graphical or related format while a verbaliser is expected to prefer data in a tabular format).

2. When explaining their decisions during the experimental task, subjects will refer more often to information in the format predicted by their score on the visualiser-verbaliserquestionnaire.
3. For impulsive subjects, as determined by the conceptual tempo cognitive style test, the effects predicted in hypotheses one and two will be greater than the average while reflective subjects should show a lesser effect.

4. 3 Method

In this section the experimental method is described. The study was performed in April, May and June 1986 involving thirty subjects, most of whom were staff from Huddersfield Polytechnic. The experiment was based around a pencil and paper type task which lasted from one-and-a-quarter hours to two hours per subject (including administering the cognitive style tests).

4. 3. 1 Design

A mixed model design was adopted. The experiment was designed to reveal the effects of different display formats (within subjects) together with different psychological types (between subjects) and individuals from business and non-business backgrounds (between subjects) upon display format preference and information use during performance at a business-related task. The variables were as follows:

Independent Variables:

1 display format,
2 the visualiser-verbaliser cognitive style dimension,
3 the conceptual tempo cognitive style dimension.

Dependent Variables:

1 display format preference,
2 information use.

In addition to these variables there were two conditions in the experiment in which the ordering of the experimental tasks was different. Subjects were randomly assigned to these conditions which were included in the design to account for any possible ordering effects. The variables given above will now be considered in more detail.

Display format: In the experimental task eight pieces of information were presented to the subject. One of these pieces of information was always presented in the same graphical format while a further three were varied between graphical and tabular formats. The remaining four pieces of information were presented as single figures towards to bottom of each presentation (see appendix 1).

The visualiser-verbaliser cognitive style dimension: A subject's position on the visualiser-verbaliser cognitive style dimension is determined using the VVQ (the visualiser-verbaliser questionnaire). This fifteen statement questionnaire is scored in such a way that a high score (maximum 15) is considered to exemplify a visualiser while a low score (minimum 0) is typical of a verbaliser. The classification of individuals on this dimension was as follows:

0 - 6	verbal
7 - 8	middle group
9 - 15	visual

These categories were based around the median score for all thirty subjects.

The conceptual tempo cognitive style dimension: A subjects position on the conceptual tempo dimension can be assessed using the MFF (the matching familiar figures test). There are, however, two possible ways to score the test; either by counting the number of errors committed by the subject during the test or by measuring the total time taken to complete the test. The correlation between these two measures is not always strong, although they did quite highly correlate in this experiment (see section 4.3.2). It was decided that both would be used, and the categories, again based around the median scores for the whole group, were as follows:

Errors:	0 - 2	reflective
	3	middle group
	4+	impulsive
Time (in seconds):	950+	reflective
	891 - 949	middle group
	below 890	impulsive

Display format preference: Towards the end of the experimental session before debriefing, subjects were asked to express their preferences regarding display format. As three pieces of information presented during the experimental task had their display format varied, the subject was asked to express a preference for each of the three pieces of information. As a result each subject provided three responses regarding their preferences. The percentages and totals presented later in this chapter are based upon these ninety replies (thirty subjects x three replies).

Information use: As it was not possible to determine exactly which pieces of information subjects had or had not taken into account during the decision-making component of the experimental task, the first three pieces of information that subjects mentioned (out of a possible eight) in explaining the reasons for their decisions, were noted. These pieces of information and their corresponding display formats formed the basis for the *information use* dependent variable.

The categorization of subjects as being from either a business or non-business background was based upon the subjects qualifications. A subject who had obtained a qualification which was, in the opinion of the *subject, strongly business related* or included a *substantial business component* was classified as having experience of the business knowledge domain.

	Business subjects	Non-business subjects	Total
Visualiser	1 male between 20-30 yrs 2 males between 31-40 yrs 1 male between 41-50 yrs Total = 4	1 female between 41-50yrs 4 males between 31-40yrs 3 males between 41-50yrs 2 males over 50yrs Total = 10	14
Middle Group	1 female between 20-30 yrs 1 male between 41-50 yrs Total = 2	1 female between 31-40yrs 1 male between 31-40yrs Total = 2	4
Verbaliser	1 female between 41-50 yrs 1 male between 31-40 yrs 3 males between 41-50 yrs 1 male over 50 yrs Total = 6	1 female between 41-50yrs 1 male between 41-50 yrs 4 males over 50yrs Total = 6	12
Total	12	18	30

Figure 4.1: The distribution of subjects within the various categories.

4. 3. 2 Subjects

A total of thirty volunteer subjects were used for the experiment, twenty-eight of whom were staff at the Polytechnic. A male visualiser who had business qualifications aged between 41-50 years and a male verbaliser who also had business qualifications aged over 50 years who were employed in senior management positions in a local companywere included in the experiment. All of the subjects, except one female aged between 20-30 years who was classified as belonging to the middle group and had business qualifications, were managers of some sort. The subjects in the experiment were classified into the categories shown in figure 4.1. It may be noted that half of the business subjects were classified as verbalisers while over half of the non-business subjects are classified as visualisers. This association, however, is not large (lambda ^{AB} = 0.07).

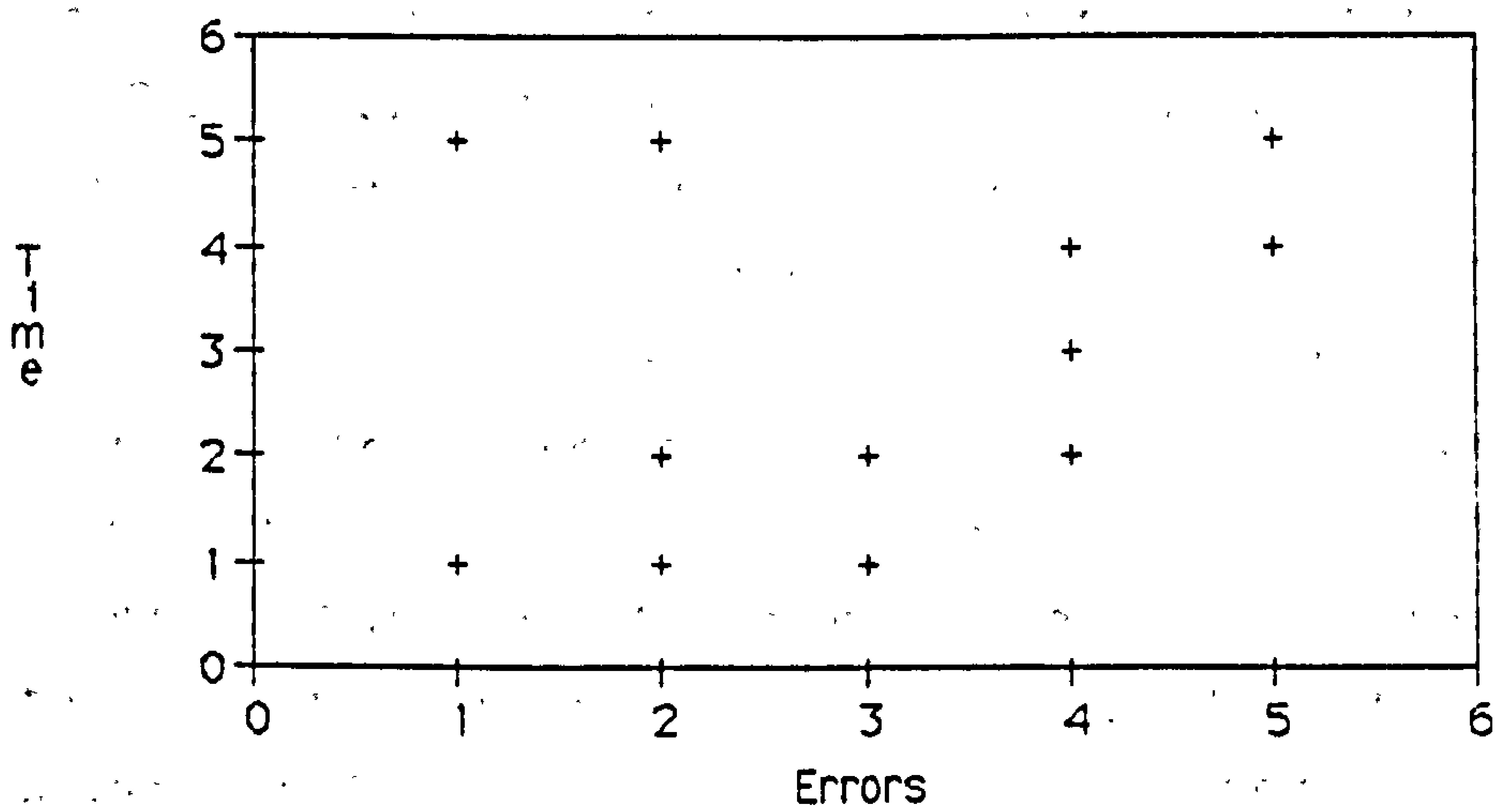


Figure 4.2: A scattergram showing the correlation between the two measures of conceptual tempo (time and errors) for the thirty subjects (several subjects occupy the same point, and hence only twelve points are shown).

The distribution of subjects within the categories of the conceptual tempo cognitive style dimension has not been included in figure 4.1 in order that the diagram might remain relatively uncomplicated. However, using errors as a basis for scoring the MFF, of the business subjects 6 were classified as impulsive, 6 as reflective and 0 as belonging to the middle group; of non-business subjects 8 were classified as impulsive, 8 as reflective and 2 as belonging to the middle group. There is no predictive association between the business and non-business subjects and their distribution within the categories of the conceptual tempo cognitive style dimension when scored using errors (lambda ^{AB} = 0). Using time as a basis for scoring the MFF; of the business subjects 4 were classified as impulsive, 8 as reflective and 0 as belonging to the middle group: of non-business subjects 8 were classified as impulsive, 6 as reflective and 4 as belonging to the middle group. There is no strong association between the business and non-business subjects and their distribution within the categories of the conceptual tempo cognitive style dimension when scored using time (lambda ^{AB} = 0.13). Furthermore, as we might expect, the correlation between the two measures of the conceptual tempo cognitive style dimension is quite high: Spearman's rho = 0.433 for 30 paired observations, $p < 0.02$ (see figure 4.2 for a scattergram).

4. 3. 3 The experimental task

The experimental task involved choosing between three possible production strategies using the information provided in the presentation regarding an imaginary product. This business-related task was checked by an associate with business qualifications and experience to ensure that the terms and concepts used closely resembled the terms and concepts commonly used in business and that the task was meaningful from an expert point of view[1]. Subjects could choose between fluctuating production, constant production with storage or constant production without storage. There were in total, twelve presentations for twelve imaginary products (see appendix 1). The figures for these presentations were based upon three original sets of figures. The other sets of figures were produced by multiplying all of the figures in the originals by constants so that the absolute figures changed while the proportionate differences between figures and the relationships between different pieces of information did not alter. The merits of these different production strategies will be expanded upon later, but first the information included in each presentation needs to be briefly explained. (The presentations used in the experiment can be seen in appendix 1).

The *turnover* figures allowed the subjects to view the product's performance over the last six years relative to other competing products in the market. The *expected sales performance* figures provided a forecast of future market share depending upon the availability of the product. One set of figures related to the expected market share if there were no problems with supply while a second set of figures showed the expected market share where production was limiting supply of the product. The *expected demand fluctuation* figures provided information on the expected demand for the coming year. The *expected production costs* figures provide information regarding the cost of production per unit with varying degrees of fluctuation ranging from constant production (0%) to 40% fluctuation. In addition to these figures the selling price per unit, the break-even point per unit, the monthly storage cost per unit and the number of units that the production department expected to have to produce in the coming year

[1] The author would like to thank Kee Fan of the Department of Management and Administrative Studies for his advice with respect to the experimental task.

were provided towards the bottom right-hand side of each presentation (see appendix 1).

In this complex task the subjects had to make *best guess* decisions using the information that was presented to them. If a fluctuating production strategy was chosen then the advantages would have been that market share would be fulfilled, but the disadvantage was that production costs would be higher, dependent upon the degree of fluctuation. If a constant production strategy without storage was chosen then production costs would be minimized, but the potential market share might not have been fulfilled. A constant production strategy with storage on the other hand, would have held production costs down, would have enabled a larger proportion of the potential market share to be fulfilled where demand fluctuations were not too great, but would have incurred storage costs.

The purpose of the experiment was to encourage subjects to make *best guess* decisions for a complex task and to observe the effects of display format upon business and non-business subjects with differing cognitive styles. In the pilot studies some subjects deliberated, in the most extreme case, for over twenty minutes per presentation and explained that they were attempting to calculate the possible financial outcomes of the three possible production strategy options while others quickly produced the *best guess* decision that was expected of them. If some subjects were attempting to calculate financial outcomes while others were making *best guess* decisions then this was inadvertently introducing a further condition into the experiment. In order to control the experiment and to ensure that all subjects were making *best guess* decisions, a time limit of one minute for the decision regarding each presentation was imposed. Many subjects were able to reach their decisions well within this time.

4. 3. 4. Procedure.

All instructions were tape recorded, although subjects were allowed to ask questions of the experimenter when there was a need for clarification or further explanation. Firstly, the subject filled in a one-page biographical questionnaire after which the visualiser-verbaliser and conceptual tempo cognitive style tests were administered. Subjects were assigned randomly to either of the two presentation order conditions (with the proviso that equal numbers of subjects were in each condition). Following this, subjects were told to imagine themselves as management executives whose task it was to choose a production strategy for each of their company's twelve products. The tape recorded instructions for these sections and the experimental task can be seen in figure 4.3.

The Experimental Instructions.

The purpose of this experiment is to investigate the relationships between performance on two psychological tests and performance at a management decision-making task. The experiment should take no longer than 2 hours. Could you now fill in the questionnaire you have been given.

For the VVQ: The first stage of this experiment involves a self-assessment questionnaire which should be in front of you. This questionnaire is not a personality test, nor is it an intelligence or ability test; it's purpose is to assess your preferred way of thinking. If you wish you can be told the results of the test at the end of the experiment. There are fifteen statements which you should read one at a time. Your task is to decide how accurately each statement reflects your own thinking and to indicate agreement or disagreement with each statement by ticking the appropriate box. If you do not understand anything then please ask. After you have finished this section then please hand the booklet over to the experimenter without turning the page.

For the MFF: The next section of the experiment is a small test which involves matching familiar figures. Your task is to point to the figure at the bottom of each page that is exactly the same as the figure at the top. Only one of the figures in the bottom half of the page is exactly resembles the figure at the top. To begin with there are two practice items one of which should be in front of you now. Again, if there is anything that you do not understand then please ask.

For the three practice presentations of the experimental task: For the final section of the experiment you are asked to imagine yourself as a management executive whose task it is to decide the company's production strategy for the next year. There are twelve different products and you are required to make the same decision for each one. Before you start these, however, there are three practice products for you to consider. The figures for the first product will be explained to you while the second practice product will give you an opportunity to question the experimenter on anything that you do not understand. For the third practice product you will have to make your decision at the end of one minute, as you will for the experiment proper. If at any point there is anything that is not completely clear then please do not hesitate to ask. If you wish to hear the instructions again at any time then the tape can be rewound and played again.

Subjects were shown the first practice presentation/product.

For each product you will be provided with four sets of figures: Firstly, figures on turnover for the product and it's rivals. Turnover is the total amount of money that the company received from sales for the product. Secondly, figures on the predicted sales performance. There is one set of figures for sales if the product is always going to be available and another set if the product happens to be unavailable for any significant length of time. Thirdly, figures on the expected demand fluctuations for the coming year and fourthly, figures on the production costs for stable and various degrees of fluctuating production. In addition to these the selling price, the break-even point and the monthly storage cost are shown. Furthermore, the total expected production figures predicted by the production department are also shown.

Because of the nature of the manufacturing equipment the company must commit itself to either fluctuating or constant production for the following year for each product. For some of the products it is better that you opt for fluctuating production while for others it is better that you choose constant production. Your task in each case is to make this decision for each of the twelve products.

Fluctuating production is always more expensive than constant production because of such factors as overtime payments, the need to maintain higher stock levels and the problems and delays in training more staff when needed. If you opt for constant production then this will maximize the profit from each item sold, but could erode the product's market position. If you opt for fluctuating production then the product's market position is less likely to be damaged, but the profit from each item sold will be less, and in some cases the company could even make a loss. An alternative choice is to opt for constant production but with storage. This will reduce the amount of profit made on each item because of the storage costs, but the chance of making a loss is not as great as it is for fluctuating production and the chance of the sales department running out of supplies of the product is slightly less.

To summarize then; your task is to evaluate the figures provided by production staff and sales staff and choose between constant and fluctuating production. The alternatives open to you should be in front of you on a separate sheet of paper. You should try to avoid choosing the same alternative for all twelve products.

For the twelve presentations: We will now begin the experiment proper. For each product the figures will be displayed for only one minute after which you will be required to declare your decision. Please try to give reasons for each decision if you can.

Figure 4.3: The tape-recorded instructions that were played to the subject.

The task was to choose a production strategy using the information presented to the subject on a single sheet of A4 paper. This task was performed by the subject twelve times using different information on each occasion. Before this subjects were shown figures for three practice products (tasks) where they could ask for further explanation. Subjects were then asked to perform the experiment proper and were given one minute to make the decision for each product. After each decision subjects were asked to explain their decision/choice of production programme and the type of information mentioned by the subject was noted as well as the order in which it was mentioned. At the end of the experiment subjects were first asked to express their information presentation preferences for each type of information used in the experimental task and were then debriefed.

4. 4 Results

4. 4. 1 Hypothesis One: Display format preference

After the subjects had finished the experimental task they were asked to express their display format preferences for the sales, demand and production figures (see section 3.3.4). The results can be seen in figure 4.4. Goodman & Kruskal's (1963) measure of predictive association, lambda (^{AB}), was used to assess the relationship between a subject's cognitive style as measured by the VVQ and their expressed display format preferences. The values for lambda (^{AB}) were as follows:

Sales	= 0.071
Demand	= 0.074
Production	= 0.250
Combinedtotal	= 0.023

	Verbaliser	Middle group	Visualiser	Total
tabular preference	sales = 5 demand = 2 production = 3 total = 10	sales = 2 demand = 1 production = 4 total = 7	sales = 3 demand = 2 production = 7 total = 12	sales = 10 demand = 5 production = 14 total = 29
graphical preference	sales = 7 demand = 6 production = 8 total = 21	sales = 2 demand = 3 production = 0 total = 5	sales = 9 demand = 10 production = 5 total = 24	sales = 18 demand = 19 production = 13 total = 50
no preference	sales = 0 demand = 4 production = 1 total = 5	sales = 0 demand = 0 production = 0 total = 0	sales = 2 demand = 2 production = 2 total = 6	sales = 2 demand = 6 production = 3 total = 11
Total	sales = 12 demand = 12 production = 12 total = 36	sales = 4 demand = 4 production = 4 total = 12	sales = 14 demand = 14 production = 14 total = 42	sales = 30 demand = 30 production = 30 total = 90

Figure 4.4: the distribution of display format preferences amongst subjects.

The association between subjects' positions on the visualiser-verbaliser cognitive style dimension and their expressed display format preferences was not strong for the sales figures, demand figures and combined total of all expressed preferences.

There was, however, a higher degree of association between subjects' positions on the visualiser-verbaliser cognitive style dimension and their expressed preferences for the production figures. However, this effect is in the opposite direction to that hypothesized. That is to say, that more visualisers preferred tables to graphs while most verbalisers preferred graphs to tables. These results were also analyzed using chi squared. Tabular information was preferred by visualisers 12 times (n = 14) and by verbalisers 10 times (n = 12). Chi squared for a 1 x 2 table = 0.02 (df = 1) NS (not significant). Graphical information was preferred by visualisers 24 times (n = 14) and by verbalisers 21 times (n = 12). Chi squared for a 1 x 2 table = 0.01 (df = 1) NS.

The relationship between expressed display format preference and a subject's position on the visualiser-verbaliser cognitive style dimension was not statistically significant and this is taken to indicate that the visualiser-verbaliser questionnaire alone should not be considered as a reliable indicator of individual display format preference. In other words; the first hypothesis, which stated that display format preferences would be related to a subject's position on the visualiser-verbaliser cognitive style dimension, has not been supported.

4. 4. 2 Hypothesis Two: Information use

As mentioned in the previous section, for each presentation the subject was asked to explain his or her decision and in each case the first three pieces of information that the subject referred to were noted. The total number of references to graphically presented information and the total number of references to information presented in a tabular format were noted as well as those occasions where the subject did not make any reference to any of the information. In short then; if a subject was to mention the sales figures in his explanation and this information was in a graphical format, but then the subject did not mention any other types of information then he would score one graphical reference and two in the *nothing mentioned* category. In total each subject could make up to 18 graphical references, 18 tabular references or could mention nothing up to 36 times. The results can be seen in figure 4.5. The value of lambda (^{AB}) for these figures = 0.01, showing little predictive association between the visualiser-verbaliser conditions and information use. The results were also analyzed using chi squared. Tabular information was mentioned by visualisers 88 times (n = 14) and by verbalisers 81 times (n = 12). Chi squared for a 1 x 2 table = 0.02 (df = 1) NS. Graphical information was mentioned by visualisers 111 times (n = 14) and by verbalisers 122 times (n = 12). Chi squared for a 1 x 2 table = 3.37 (df = 1) NS.

	Verbaliser	Middle Group	Visualiser	Total
Graphical Information	122	41	111	274
Tabular Information	81	34	88	203
Nothing Mentioned	229	69	305	603
Total	432	144	504	1080

Figure 4.5: The number of references by subjects to information in differing formats.

The second hypothesis stated that the information that subjects referred to would be related to both the format of the information and the subject's position on the visualiser-verbaliser cognitive style dimension. This hypothesis has not been supported. These results appear to suggest that the visualiser-verbaliser cognitive style dimension alone is unlikely to prove useful as a reliable indicator of information use when the information is in either a tabular or graphical format.

4. 4. 3 Hypothesis 3a: Display format preference and the conceptual tempo and visualiser-verbaliser cognitive style dimensions

The third hypothesis stated that a subjects position on the conceptual tempo cognitive style dimension would effect the extent to which the visualiser-verbaliser dimension might determine display format preferences and information use. In other words; that the conceptual tempo dimension would mediate the effects of the visualiser-verbaliser cognitive style dimension. More specifically, it was suggested that impulsive visualisers and verbalisers would show greater preference for graphical and tabular display formats respectively than reflective subjects.

As the MFF test can be scored using either time or errors as the measurement criteria, then analyses are presented for both. If the MFF test is scored by errors; tabular information was preferred by visual impulsives 6 times (n = 6), by visual reflectives 5 times (n = 8), by verbal impulsives 6 times (n = 6) and by verbal reflectives 4 times (n

= 6). Chi squared for a 1 x 4 table = 1.02 (df = 3) NS. If the MFF test is scored by time; tabular information was preferred by visual impulsives, 3 times (n = 4), by visual reflectives 4 times (n = 6), by verbal impulsives 6 times (n = 5)-and by verbal reflectives 4 times (n = 7). Chi squared for a 1 x 4 table = 3.58 (df = 3) NS. For both of these analyses some of the expected cell frequencies fall below '5' and so the values of Chi squared in this case must be treated with some caution.

If the MFF test is scored by errors; graphical information was preferred by visual impulsives 8 times (n = 6), by visual reflectives 17 times (n = 8), by verbal impulsives 7 times (n = 6) and by verbal reflectives 14 times (n = 6). Chi squared for a 1 x 4 table = 3.53 (df = 3) NS. If the MFF test is scored by time; graphical information was preferred by visual impulsives 5 times (n = 4), by visual reflectives 12 times (n = 6), by verbal impulsives 7 times (n = 5) and by verbal reflectives 14 times (n = 7). Chi squared for a 1 x 4 table = 1.40 (df = 3) NS. None of these results are statistically significant.

With respect to display format preference the third hypothesis has not been supported. That is to say, that there is no proven relationship between subjects' display format preferences and their positions on the visualiser-verbaliser and conceptual tempo cognitive style dimensions. As a result, it can be stated that the utility of the visualiser-verbaliser cognitive style dimension and the conceptual tempo cognitive style dimension in predicting information display format preferences at the human-computer interface appearsdoubtful.

4. 4. 4 Hypothesis 3b: Information use and the conceptual tempo and visualiser-verbaliser cognitive style dimensions

The second hypothesis was concerned with information use as a consequence of display format and a subject's position on the visualiser-verbaliser cognitive style dimension. In the terms of the third hypothesis, it was considered likely that impulsives would mention only information in their predicted format while reflectives would mention information which had been presented in both their predicted and non-predicted format (predicted means; as predicted using the VVQ). Again two analyses are presented, one for conceptual tempo scored using errors and the other scored using time.

If the MFF test is scored by errors; tabular information was mentioned by visual impulsives 40 times (n = 6), by visual reflectives 48 times (n = 8), by verbal impulsives 52 times (n = 6) and by verbal reflectives 29 times (n = 6). Chi squared for a 1 x 4 table

= 7.23 (df = 3) NS. If the MFF test is scored by time; tabular information was mentioned by visual impulsives 25 times (n = 4), by visual reflectives 34 times (n = 6), byverbal impulsives 41 times (n = 5) and by verbal reflectives 40 times (n = 7). Chi squared for a 1 x 4 table = 3.58 (df = 3) NS.

If the MFF test is scored by errors; graphical information was mentioned by visual impulsives 51 times (n = 6), by visual reflectives 63 times (n = 8), by verbal impulsives 72 times (n = 6) and by verbal reflectives 50 times (n = 6). Chi squared for a 1 x 4 table = 7.53 (df = 3) NS. If the MFF test is scored by time; graphical information was mentioned by visual impulsives, 41 times (n = 4), by visual reflectives 43 times (n = 6), byverbal impulsives 52 times (n = 5) and by verbal reflectives 70 times (n = 7). Chi squared for a 1 x 4 table = 4.30 (df = 3) NS.

The results, whether the MFF is scored using errors or time, are not statistically significant. With respect to information use, it appears as though the third hypothesis has not been supported. In other words, there is no predictable relationship between display format, information use and the conceptual tempo and visualiser-verbaliser cognitive style dimensions. The usefulness of the visualiser-verbaliser and conceptual tempo cognitive style dimensions in predicting information use as a consequence of display format at the human-computer interface has not been demonstrated.

4. 4. 5 Information use and display format preference

Although neither display format preference or information use have been found to be related to either of the two cognitive style dimensions, there was still the possibility that individuals might predominantly use information in the format for which they stated a preference, regardless of their cognitive style. Using Spearman's rank correlation for subjects stated preferences and their information use of tabular information r_s = +0.202 (N = 30) NS. For the correlation between the subjects stated preferences and their information use for graphical information r_s = +0.203 (N = 30) NS. It appears as though subjects' stated display format preferences did not determine the information they used during the experimental task to explain their decisions.

4. 4. 6 Display format preference, information use and business subjects

As mentioned in the introduction, it was considered possible that either display format preferences or information use might have been influenced by subjects experiencein business. For information preference: Tabular information was preferred by business subjects 11 times (n = 12) and by non-business subjects 16 times (n = 18).

Chi squared for a 1 x 2 table = 0.01 (df = 1) NS. Graphical information was preferred by business subjects 19 times (n = 12) and by non-business subjects 32 times (n = 18). Chi squared for a 1 x 2 table = 0.07 (df = 1) NS. For information use: Tabular information was mentioned by visualisers 71 times (n = 12) and by verbalisers 134 times (n = 18). Chi squared for a 1 x 2 table = 2.24 (df = 1) NS. Graphical information was mentioned by business subjects 104 times (n = 12) and by non-business subjects 170 times (n = 18). Chi squared for a 1 x 2 table = 0.40 (df = 1) NS. Experience of the business knowledge domain does not appear to significantly influence information display format preference or information use during the experimental task.

4. 4. 7 Display format preference, information use and non-business subjects

If business subjects are excluded from the analysis then the results are as follows: Tabular information was preferred by visualisers 8 times (n = 10) and by verbalisers 6 times (n = 6). Chi squared for a 1 x 2 table = 0.02 (df = 1) NS. Graphical information was preferred by visualisers 18 times (n = 10) and by verbalisers 11 times (n = 6). Chi squared for a 1 x 2 table = 0.02 (df = 1) NS. For information use: Tabular information was mentioned by visualisers 61 times (n = 10) and by verbalisers 41 times (n = 6). Chi squared for a 1 x 2 table = 0.21 (df = 1) NS. Graphical information was mentioned by visualisers 82 times (n = 10) and by verbalisers 55 times (n = 6). Chi squared for a 1 x 2 table = 0.30 (df = 1) NS. If business subjects are excluded from the analysis then the results remain non-significant.

4. 4. 8 Information redundancy

At the end of the experiment subjects were asked: "If, for each set of figures in the presentation, the information had been displayed in both a tabular and a graphical format would you be (a) happy with this situation, (b) unhappy with this situation or (c) do you believe that this is not necessary." The results were as follows; 4 business subjects and 14 non-business subjects stated that they would be happy with redundant information, 8 business and 4 non-business subjects stated that they thought redundant information unnecessary, while none of the subjects stated that they would be unhappy with information redundancy. The question was asked without the subjects being given any examples of what a presentation with information displayed in both formats might look like and so can only be considered as a rough indication of true preferences.

4. 4. 9 Differences between business and non-business subjects

During the experiment clear differences emerged between subjects with and subjects without experience in business. Typically, the latter appeared to concentrate upon the absolute differences between the figures relating to production and other costs. Business subjects on the other hand, looked for trends and relationships between all of the sets of figures. This subjective account is viewed as tentative support for the idea that subjects with experience in business approached the experimental task in a way which was different from non-business subjects. Further support for this notion was found by analyzing crude differences in word use during the experiment by business and non-business subjects when they were explaining their reasons for the decisions. If subjects used either of the words *market* or *competitor* more than three times during the experiment then they were classed as having adopted a *market view*. Of the business subjects, 10 adopted a market view while 2 did not. Of the non-business subjects, 2 adopted a market view while 16 did not. Using Fisher's exact probability test it can be seen that statistically the result is highly significant ($p < 0.0002$), thus supporting, to a limited extent, the idea that business subjects approached the experimental task in a way which was different from non-business subjects. However, there were no difference between business and non-business subjects in their use of information in different formats. Business subjects mentioned graphical information 104 times ($n = 12$), while non-business subjects mentioned graphical information 164 times ($n = 18$). Chi squared for a 1 x 2 table = 0.19 ($df = 1$) NS. This may be because business subjects, although apparently wholists in their approach, would use all of the information regardless of format.

4. 5 Discussion

4. 5. 1 Display format preference and information use

The original aim in designing the experiment was to show the effects of the visualiser-verbaliser and conceptual tempo cognitive style dimensions upon information display format preference and information use during a decision-making task. Overall, the experiment was viewed as a limited test of the notion that behaviour may be determined by over-riding factors, such as the user's cognitive style. The results have shown that the measures of the two cognitive style dimensions (the VVQ and the MFF) could not be used to predict display format preferences or information use.

One possible explanation for these results is that the effects of subjects applying their business knowledge *swamped* the effects of cognitive style. However, if this was the case we might have expected subjects without experience in business alone to show an effect, but there are no differences between verbal and visual non-business subjects in terms of either their stated display format preferences or their use of information during the experimental task (see section 4. 4. 7). Another explanation is that the task was too complex for cognitive style to show an effect. The results of this experiment cannot be used to either confirm or refute this suggestion. It might be noted, however, that many tasks at the human-computer interface are complex, and it is suggested that the visualiser-verbaliser and conceptual tempo cognitive style dimensions must be shown to have an effect upon complex as well as simple tasks if they are to be of use in HCI.

To reiterate; for this complex experimental task the visualiser-verbaliser and conceptual tempo cognitive style dimensions were not found to have significantly influenced display format preference. Furthermore, no clear relationship emerged between display format preference and any other measurable factor in the experiment. Display format preferences were widely distributed for both business and non-business subjects and these preferences were not found to effect the information subjects chose to use in explaining their decisions during the experimental task. As a consequence of these results, it might have been proposed that information displayed at the human-computer interface should be presented in more than one format. However, the *poll* carried out at the end of the experiment indicated that, although no subjects stated that they would be unhappy with the display of redundant information, a number indicated that they did not believe it to be necessary.

4. 5. 2 The differences between business and non-business subjects

There is already evidence to demonstrate that people with experience of the same knowledge domain share common representations of that domain (Adelson, 1984, 1985; Weiser & Shertz, 1983). Furthermore, there is evidence to show that representation of a particular knowledge domain strongly affects how individuals reason about issues and problems associated with that domain (Newell & Simon, 1972; Chi *et al*, 1981; McKeithen *etal*, 1981; Riesbeck, 1984). Therefore, it is not unreasonable to expect people who share a common experience of a particular knowledge domain to behave in a similar fashion when faced with a task associated with that domain. For although disparities in expertise are likely to give rise to differences in behaviour between individuals with a common background (Rector *et al*, 1985), these are unlikely to be as great as the differences between those who have and those who do not have an understanding of a particular area of knowledge.

Therefore, according to the literature there should have been marked differences between business and non-business subjects, both in terms of how they talked about the task, as well as their overall approach to it. Indeed, this is what was found. Business subjects appeared to consider all of the information in the display while non-business subjects concentrated upon comparisons between one or two sets of figures. Moreover, business subjects tended to talk in terms of *markets* and *competitors* while non-business subjects did not. The contrasting behaviour of those with and those without experience of the business knowledge domain suggests that those with business qualifications shared a common representation of the domain relative to those subjects without qualifications and furthermore, that this representation influenced their behaviour at the experimental task. Moreover, the differences between business and non-business subjects suggest that user groups may differ from one another in their representation of the task as well as from the systems' designers representation of the task. All of this, however, is well known within the field and the observations reported here can only be considered as weak support for what has been accepted within the literature for some considerable time (Newell & Simon, 1972; Adelson, 1984, 1985; McKeithen, 1981; Rector *etal*, 1985; Riesbeck, 1984; Weiser & Shertz, 1983).

4. 6 Conclusion

The results of this study have suggested that the visualiser-verbaliser and conceptual tempo cognitive style dimensions are unlikely to be of use in predicting display format preferenceat the human-computer interface. A possible explanation for these results is that simple main effects, such as those that might result from a user's cognitive style, are not great enough to significantly influence the user's behaviour at a complex task. Such an explanation is consistent with other findings in HCI, and the idea that behaviour at a complex task is often determined by the interaction of a number of variables (Barnard & Hammond, 1982; Eason, 1984). The idea of classifying users, at least with respect to the instance used in this experiment, appears to miss this subtle interaction.

Overall, the experiment appears to have lent support to the idea that behaviour at a complex task is not determined by any single over-riding factor. It appears as though behaviour arises out of the interaction of a number of factors, and that this interaction might be both complex and difficult to predict. In essence, it seems as though it may not be possible to consistently predict the detail of behaviour, because of the complexity of its origins.

Chapter 5

Using Errors to Understand the User

5. 1 Overview

A single theme seems to emerge from the previous chapters: that predicting detailed user behaviour in a consistent fashion is likely to prove problematic for two reasons. Firstly, cognitive grammars intended to predict various aspects of user behaviour (or cognitive complexity) have a number of theoretical and practical problems which may mean that they are only suitable as research tools (see chapter 3). Secondly, it appears unlikely that any single over-riding factor, such as cognitive style, determines user behaviour at the human-computer interface (see chapter 4).

In this chapter an alternative to using cognitive formal methods for modelling the user is proposed, where formal analysis concentrates upon the errors or dialogue failures that occur during human-computer interaction. It is suggested that errors or dialogue failures that occur during interaction reveal the model mismatches that present formal methods attempt to predict, and that an approach to modelling the user, that is based upon an analysis of such errors might have a number of advantages. Firstly, such an analysis could focus upon mismatches that actually occur between users and systems rather than mismatches that might or should be present. Secondly, an approach such as this might fit easily into present design practice, where almost all systems are iteratively developed, and many are presently tested with users. Thirdly, a consideration of those points where user-system errors occur might help to direct attention and resources to those parts of a design that require attention. Bearing these potential benefits in mind, a set of criteria that any user-system error analysis technique should meet is suggested. The criteria are specifically aimed at techniques that might attempt to classify errors according to their causes.

A classification scheme, that formed the basis for the research into the approach, is described, together with its evolution. To this end, human error classification schemes are considered, together with a number of suggested schemes for classifying user-system errors. The classification scheme, that was derived from these schemes together with other approaches within the literature, is then described.

The classification scheme that was developed, and is described here, ECM (Evaluative Classification of Mismatch), is a four-stage classificatory scheme for describing model mismatches between a computer system and its user. At the first stage a failure is identified in the dialogue between the computer system and the user. At the second stage the mismatch between the computer system and the user (as shown by the dialogue failure) is related to either an object or operation within the computer system. The problem with either the object or the operation is then classified as a concept or symbol mismatch at the third stage in the classification. The fourth stage involves the description of the mismatched element and its contribution towards the dialogue failure. At a more detailed level, it is expected that there would be two benefits to be gained from classifying model mismatches in this way: First, ECM could help to clarify situations where more than one mismatch may have contributed towards a dialogue failure. Second, a standardized terminology for discussing model mismatches might assist communication between the different members of a design team.

The overall purpose in developing the scheme was to use it as a vehicle for research into an approach concentrating upon user-system errors. The purpose of this chapter, then, is to present the case for concentrating upon user-system errors, outline the classification scheme, and its development, as a backdrop to the later studies.

5. 2 Introduction

Having decided that cognitive grammars are unlikely to be suitable for use within the design and development process, and that the use of cognitive style measures may possesses certain drawbacks, this presents the question of, *how can we model the user in design?* One possible answer might be to concentrate upon the errors that occur during human-computer interaction. An analysis of such errors might expose the underlying and fundamental mismatches that often exist between the designer's and the user's model of the task.

5. 2. 1 What are user-system errors?

There are a number of possible terms that might be used to describe user-system errors. Indeed, it is questionable whether the term "error" is really the most appropriate in this context. A better term might be "misunderstanding". Even this term, however, might not be as precise as we would like, for it appears to imply that such occurrences are trivial. Consequently, it is suggested that events where the user's actions and the systems responses are not wholly compatible, should be considered as *dialogue failures.* A *dialoguefailure* is a breakdown in communication between the system and the user; it is where either the computer or the user do not understand one another, or some information about the nature and structure of the task is not properly communicated.

A dialogue failure can be considered as evidence of a mismatch between the user's and the designer's model of the task and system. That is to say, that dialogue failures can reveal model mismatches. In other words, while present formal methods attempt to predict potential model mismatches, dialogue failures directly reveal actual mismatches that exist between the user and the designer.

Although the general notion of a dialogue failure may prove to be of use in a formal sense, in practice an operational definition may be of use. Here, a dialogue failure is considered to have occurred if:

- the user reports any degree of misunderstanding during the interaction (i.e. the system does not do what he or she wanted it to do),
- the user asks for help in any form,
- the user enters an illegal command that is not purely the result of a keystroke error, mental slip or lapse.

5. 2. 2 The advantages of considering user-system errors

The advantages of considering user system errors, or dialogue failures, is that they directly expose actual model mismatches between the user and the designer. Freud (1922) described human errors as "..windows to the mind." Within human-computer interaction user-system errors might be viewed in a similar light; as a means by which the clashes between conflicting models of a task might be exposed.

One of the central aims of cognitive formal methods is to expose potential model mismatches that might or should happen. The problem with this approach is that the detail of user behaviour is difficult to predict. By concentrating upon user-system errors, however, this problem is avoided, and it appears as though such an approach might allow more accurate modelling of the user. Furthermore, such an approach might also allow us to model just those aspects of a task that require analysis; those parts where the user and the designer disagree. In essence, a consideration of user-system errors can help to highlight those parts of a design that require attention.

A further advantage is that such an approach might fit readily into present design and development practice. Most systems are now developed in a iterative manner (see Bellotti, 1988), even if these systems are not always tested with users in any systematic fashion. Moreover, the trend within the major IT companies is towards more rigorous usability testing of systems as they are iteratively refined. In many cases user-system errors are already considered, but in a manner that often lacks formal analysis. Consequently, a formal technique that was intended for the analysis of such errors might fit easily into both current and future design practice.

5. 2. 3 How might we analyse user-system errors?

While we may be clear about the place of a formal user-system error analysis tool within the design and development process, the form that such a tool might take is not as immediately obvious. One possible approach, adapted from approaches to human error, might be to employ classification techniques. In essence, to classify user-system errors according to their origin and so progress towards a greater understanding of the root cause of any model mismatch.

Directly using human error frameworks, however, does not appear to be a viable proposition. For example, consider Rasmussen's (1976, 1980) distinction between skill-based slips, rule-based mistakes and knowledge-based mistakes. How might we apply this to actual user-system errors? Subjects using the Memomaker word-processing system (see chapter 6 for more detail) complained that the "backspace" key did not delete the text, it only moved the cursor backwards; acting as a further cursor key. It does not appear as though classifying such a user-system error as either skill-based, rule-based or knowledge-based might help to understand the model mismatch in any greater depth. The reasons for this are twofold.

Firstly, models of human error and their associated techniques are aimed at analyzing errors within cognition, while a user-system error is a mismatch between two models, it is not necessarily an error within either of the two models. In other words, a user-system error is a mismatch between systems, not an error within just one. Secondly, although the terminology used within the human error field might reflect human cognition in a useful fashion, such terminology does not necessarily translate to the design and architecture of computer systems. It appears as though any classification scheme might need to use a set of concepts and terms that might have currency within both the computing and cognitive domains.

5. 3 Criteria for a user-system error classification scheme

Although the requirements outlined above may give some idea as to the nature of a user-system error classification scheme, it may be useful to outline in more detail the desirable characteristics of a classification scheme. Some of these criteria are not necessarily specific to a user-system error classification, but might be viewed as general requirements of a classification scheme.

5. 3. 1 Usable

The term *usable* is taken, in this context, to mean usable by the designer rather than the system user. It is suggested that any evaluative classification should be usable as part of the design process and for this to be possible two conditions need to be met: First, it must fit into current or future design practice (Castell, 1986; Damodaran, 1983). If for example, in using the classification the designer is forced into an unaccustomed and unacceptable role then, in the real world, it is doubtful whether designers would be likely to utilize the scheme. Second, it should be relatively simple to employ. If the classification is too time consuming, too complicated or requires great intellectual effort to apply then it is improbable that designers or researchers will choose to use it.

5. 3. 2 Generalizable

Most of the classification schemes for user-system errors to date have tended to be system specific (eg Davis, 1983a & b; Welty, 1985). That is to say, that these classifications tend to reflect the system structure rather than more basic or fundamental classes of errors. These classifications were never intended as anything more than system specific tools and although interesting, their generalizability to other systems appears to be limited. If a classification scheme is to be widely applicable then it needs to *generalizable,* and to achieve this it is suggested that it should reflect fundamental classes of errors rather than any one system structure in particular.

5. 3. 3 Comprehensive

The term *comprehensive* is taken to mean that it should be possible to classify all user-system errors under one scheme. A classification that is not shown to be comprehensive may either have classes missing, or the scheme itself (or its associated definitions) may be inaccurate and unuseful as a reflection of fundamental classes of errors. A scheme where some errors are not easily classified or which leads to residual categories of unclassifiable errors is clearly undesirable.

5. 3. 4 Specific

A classification of errors which is intended as a potential tool for system evaluation must not only classify error types, but must also show where errors arise. In other words, the process of using a classification scheme needs to show the points in both the dialogue and the system where improvement is needed. For example, it is not enough to say that parts of a system show inconsistencies, it is important to note which parts as well as how they are inconsistent. Many of the studies regarding error classification published to date were designed to reveal frequency data regarding different types of errors in particular systems, rather than identify specific points where errors arise. As a consequence many of these classification schemes are unsuitable as tools for evaluation for they do not relate to *specific* points in a systems design where change and improvement is needed (other schemes will be dealt with in the following chapter).

5. 3. 5 Detailed

In creating a classification scheme which is both generalizable and comprehensive it is possible that the classes of errors in the scheme might be so broad that they do not provide the designer with the level of detail that is required to effectively pinpoint the problems within the system. In other words, having shown that there is something amiss with either an aspect of the system, there is then a need for a *detailed* description of what is wrong. It appears improbable that a classification of user-system errors which did not provide an adequate degree of description would be of practical use as a design tool. It is suggested that an evaluative classification should provide the design team with detailed information relevant to the task of creating a usable system.

5. 3. 6 Classifying from the user's perspective

As Lewis & Norman (1986) point out, the term *error* is not entirely appropriate to the task of evaluating user-system performance. When a system will not perform an operation in the way a user intended then a better term for this occurrence might be *misunderstanding* (although this term also has its drawbacks, as mentioned earlier). These misunderstandings arise from the different conceptualizations (or models) of the task held by the user and the system, and so these occurrences might be said to arise from *model mismatches.* It was suggested earlier that model mismatches will be revealed as failures in the dialogue between the system and the user. Here, it is suggested that an evaluative classification, to be of any use, should not simply document and chart what the system interprets as *errors*, but should aim to classify the causes of any misunderstanding. That is to say, that there is a need to understand dialogue failure and inefficiency within the context of one simple conceptual framework. Furthermore, this framework should provide the basis for both categorizing *errors* and communicating the reasons for their occurrence.

Although Norman's classification (1983) is aimed at explaining at least some of the causes of user-system misunderstandings, many of the error classifications that have so far been published do not. Consequently, they tend to fulfill the role of charting *errors* rather than explaining their clauses. In short, an evaluative classification scheme must classify from the user's perspective if dialogue failures are to be understood.

5. 3. 7 Summary

An evaluative classification of user-system model mismatches should consist of fundamental categories which are generalizable across systems and classify system concepts from the user's perspective. These categories should provide detailed information related to specific points in either the dialogue or design where improvement and change in needed. Finally, it must be easy to understand, convenient to use and easily applicable.

5. 4 The Origins of ECM

Having outlined the criteria a user-system error classification scheme should meet, it may be now useful to chart the development of the classification scheme that was employed. As has been mentioned, the idea of directly employing a human error classification was rejected as inappropriate. Nevertheless, it may still be useful to briefly review human error classification in order to identify those elements that might be employed in the classification of user-system errors.

5. 4. 1 Human error classifications

A popular distinction, within the human error literature, is between *slips* and *mistakes* (Norman, 1981a & b; 1983; Reason & Mycielska, 1982). A *mistake* is where the intention or plan is inappropriate, while a *slip* is where there is an error in executing a plan. For example, if we imagine that we have been asked to make a cup of tea then a number of errors can arise. We may misunderstand the request and make a cup of coffee instead; this would be a *mistake.* On the other hand we might select the correct plan (making a cup of tea), but then accidentally put coffee in the teapot; this would be a *slip.*

Rasmussen (1976, 1980) has distinguished between *skill-based, rule-based* and *knowledge-based* slips. At the *skill-based* level we receive *signals,* and we respond to these signals automatically, using pre-programmed sets of instructions embedded within schemata. An example might be driving a car. At the *rule-based* level we respond to signs, using *if <situation> then <action>* rules. For example, if we were to monitor a swimming pool and we notice the pH is falling, then we add an alkali substance to raise the pH to neutral. At the *knowledge-based* level we manipulate symbols in order to form and plan action sequences.

These distinctions have been used for some time within the human error field, and more recently Reason (1987) has attempted to combine these distinctions into a comprehensive model (GEMS: Generic Error-Modelling System). Their utility might be attributed to the way in which they allow us to consider an error within an overall framework (i.e. occurring at either a skill-based, rule-based or knowledge-based level). In short, such distinctions, when properly constructed, allow us to break down errors is such a way that we can look for commonalities between them, and identify the domains within cognition that gave rise to them.

The substance of the argument is that an approach to user-system errors requires equivalent distinctions, in order that we can identify the common elements and causes of these errors. However, as mentioned in the previous chapter, directly applying the classifications or distinctions from the human error field may not be appropriate for two reasons. Firstly, user-system errors are mismatches between two models of a task, they are not necessarily errors within any one cognitive system. Secondly, the terminology used in human error classifications does not translate to the computing domain in a useful way. In essence, human error classifications provide an example to follow, but are not, themselves, applicable to the study of user-system errors.

5. 4. 2 User-system error classifications

Nevertheless, Norman (1983) has developed a classification that is based upon human error classifications (see Riley & O'Malley, 1984, for an example of its use). We will examine the limitations of this approach later, first we need to describe it.

Under Norman's (1983) scheme we have the same major distinctions that are used in human error classifications. That is to say, that errors can be considered as either slips or mistakes (see earlier definition). However, two new categories have been added. The first is a *mode error,* where a person believes a system is in one mode when it is in another, and, consequently, performs an action that is inappropriate for that particular mode. The second new category is a *description error,* where there is insufficient specification of an act, resulting in an erroneous action. An example Norman (1983) provides is where a row of identical switches have been provided, and distinguishing these switches is difficult.

This classification may prove to be a useful way of considering human error in a computing environment, but it cannot be used to classify those errors (or misunderstandings, or dialogue failures) that do not result from human error. For example, on Prime™ systems the command "spool" must be used to print a file. However, many users find the use of such a term confusing and would prefer the command "print". This is a user-system error to the extent that it causes dialogue failures, but how might it be classified under Norman's scheme? It may not necessarily be a description error because the user may just stop working and ask for help, and under Norman's scheme a description error is where inadequate description leads to an erroneous act. It may not be described as a mode error, as the user was in the correct mode.

Thus, Norman's (1983) classification may be useful when considering human error at computer systems, but may not be an appropriate means of considering user-system errors. In essence, it is suggested that we should consider *human error* and *user-system error* to be different beasts. A human error arises from a fault in cognition, whereas a user-system error is a communication failure caused by a mismatch between two models. This does not necessarily mean that both might not be considered within one model or classification. Nor does it mean that there is not some overlap and indistinct areas that lie between the two. The area of human error is vast, however, and, although human error at computer systems is an important and interesting field of concern, it cannot be addressed here. Accordingly, human error will not be considered, but the focus of attention will be restricted to user-system errors.

Returning to the question of Norman's classification for a moment; the point that is being made is that Norman's (1983) classification, and schemes like it, are addressing the question human error at computer systems, and they are not an appropriate means by which user-system errors might be considered. Davis (1983a & b), however, does attempt to classify user-system errors, although he refers to these errors as "human errors".

Davis (1983a & b) used the distinction between syntactic and semantic levels, from Moran's (1981) CLG, to classify errors. Typically, errors were classified as being either *typing, syntactic, semantic, mode* or *intra-mode*. However, in common with other user-system error classifications (see Welty, 1985) much of Davis's scheme is based around the architecture of the system in question. As a result, his classification scheme will not generalize from one system to the next. Furthermore, the central aim of Davis's scheme is to provide statistics regarding the relative frequencies of errors relating to various aspects of the system, it is not to analyse each error as a means to identifying the root cause. As a consequence, the most useful aspect of an error classification system (the analysis of root causes) is excluded from this type of number-crunching scheme.

5. 4. 3 Deriving fundamental terms

We have established that present error classification schemes are either aimed at considering human error, and therefore are inappropriate, or are based around the system's architecture, and so cannot be generalized to other systems, and do not classify in such a way so as to expose root causes. If we are to derive a classification scheme, aimed at analyzing user-system errors, that avoids the pitfalls of present schemes, then we need to identify a variety of fundamental concepts that might be used within a classification framework. It is important to identify fundamental concepts in order that we can truly analyse user-system errors. Furthermore, only fundamental concepts are likely to transfer easily from one system to another.

An obvious starting point is to consider using Moran's (1981) distinction between the syntactics and the semantics of interaction. However, classifying the causes of errors with respect to whether they are semantic or syntactic is not as easy or as obvious as it first seems. In classifying a user-system error we need to get to the root cause, from the user's perspective. The problem may be with an action the system has performed or an object within the system; distinguishing the semantic and syntactic components of any problem may not necessarily clarify the situation, as there is no reason to believe that the user separates the semantics of the situation from the syntactics of operating the system. This distinction is a notional one that is of use within some areas of human-computer interaction, it is not necessarily a practical division. In other words, as the classification has to be from the user's perspective to be meaningful, then dividing up elements of a system into semantic and syntactic components may be a pointless abstraction.

For instance, if we consider the action of ejecting a disk from the Apple® Macintosh™ computer, then we have two possibilities. One is to select the disk and choose "Eject" from the "File" menu, or the other is to take the disk icon and drag it to the wastepaper basket. Imagine that we are new to the system, and we do not know how to eject the disk. That is to say that the operation is not obvious to us. Is this a semantic problem or a syntactic one? It may be semantic as we know what we want to do, but do not know the sequence of action that we need. On the other hand, it may be a semantic problem, as the actions required to eject the disk are not obvious.

This sort of reasoning is not only time-consuming, but does not necessarily guarantee any greater meaningful and useful understanding of the problem. Nevertheless, although Moran's (1981) distinction between the semantic and syntactic components of interaction may not be of great use, some of the other components of interaction that Moran and others (e.g. Clarke, 1986) suggest appear to lend themselves to a possible classification scheme.

Of particular interest is the notion of *objects* and *actions* within a system. An *object* within a system might be a file or a directory. In essence, *objects* are things which have actions performed upon them. *Actions* are things that are performed upon objects. However, the term *action* is a broad term that can sometimes be taken to mean more than simply changing the state of an object within a physical or virtual system. For this reason a more limited term was sought, and the term *operation* was chosen. Accordingly, within the ECM (Evaluative Classification of Mismatch) scheme, which will be described formally a little later, *operations* are performed upon *objects* within the system.

The advantage of using such fundamental concepts, as explained earlier, is that they easily transfer from one system to another. Within the ECM classification scheme user-system errors are attributed to operations and objects within the system. In other words, operations or objects that are at fault, as far as the user is concerned, are identified.

However, once a problem with either an object or an operation has been identified there appears to be some need of further analysis. That is to say, that the faulty aspect of the object or operation needs to be identified. One possible means of distinguishing different aspects of either an object or an operation is to attribute failures to either the *symbol* or the *concept* of either the object or operation. In other words, either the symbol representing the object or operation is incorrect from the user's perspective, or the actual concept of the object or operation is inappropriate.

The ECM classification, as it has so far been described, was trialed informally with some success. That is to say, that dialogue failures could be classified according to whether it was an object or operation that was at fault, and whether it was the symbol representing the object or operation or the concept of the object or operation that was inappropriate.

Unfortunately, one further aspect of the scheme did not fair so well during the informal trials[1]. The notion of context was introduced to account for failures that arose from, among other things, confusions over mode. The idea being that an object or operation might be in an inappropriate context from the user's perspective. This, however, proved problematic. For instance, a user complained that an operation was placed in an illogical position within a menu, and that another menu would have been better suited. Thus, the concept of the operation was in the correct context, but the symbol was not.

Therefore, not only was either an object or operation said to be in an inappropriate context, but that aspect of the object or operation (either the concept or the symbol) was identified as being in the wrong context. There were, however, further problems with the notion of a *context* mismatch. It became impossible to distinguish, in many cases, between a mismatch where the concept was in an inappropriate context and where the concept itself was incorrect from the user's perspective.

The solution that was adopted for this problem was to abandon the notion of *context* as a formal category. At the end of the classification process it was proposed that a dialogue failure (model mismatch) should be described with respect to its position and relationship with other failures. To compensate for the lack of a formal *context*,

[1]These informal trials simply consisted of the author attempting to classify dialogue failures using the ECM scheme.

categoryit was proposed that any aspect of the failure that related to context problems should be mentioned as part of this description.

5. 4. 4 The theoretical underpinning of ECM

The ECM approach draws from two traditions; cognitive grammars in HCI and human error classifications. Grammars describe systems in terms of constructs that embody a particular view of the world. For example, if we use TAG, as described in chapter two, then we might view tasks in terms of *simple tasks* and *rule schemata.* Similarly, within ECM, there is a particular perspective on how tasks are constructed (or alternatively, how they might be decomposed), in terms of operations, objects, symbols and concepts. Essentially, ECM represents an attempt to use some of the concepts and definitions developed for early evaluation of systems (i.e. using cognitive grammars) and apply them in the context of late evaluation.

There is, however, a significant point of departure: grammars are used to describe whole systems (or parts of systems). That is to say, that every simple task and rule schemata is identified. When employing ECM, rather than describe the whole system, the intention is to identify only those components of the system that are mismatched with the user's perspective. In this respect the use of ECM entails classification and, consequently, has a good deal in common with the human error classifications of Reason (1989), Rasmussen (1986) and Norman (1983b). The objections to using these human error classifications have been set out earlier. However, there is a further point of theoretical departure for ECM, from these classification schemes: while such schemes might identify what type of error occurred during the performance of a task (e.g. rule-based, lapse, slip or mode error), ECM identifies the component of the system that is at fault, from the user's perspective. Human error classifications are aimed at identifying where a human's mental processes contributed towards an error, whereas ECM takes the user's understanding as a starting point, and assumes that the user's understanding and mental processes are correct, and that the system is in error. In essence, the objectives, in using a human error classification scheme and ECM are different; the first is concerned with identifying mental processes that lead towards a human error, while the second is aimed at identifying mismatched systems components.

Overall, although ECM might share a common approach with human error classification schemes, it is probably better viewed as a natural development from the cognitive grammars that have been used in HCI. When grammars were first suggested towards the end of the last decade, it was proposed that whole system could be analysed using them. Since then the impracticality of such an approach has been recognized. For example, Moran (1986) has suggested that they should be used to analyse only small parts of a system; those parts that require attention (how such parts might be identified is not addressed.) Consequently, ECM might be viewed as a further development in this direction. Instead of identifying every system component in a particular part of a system as Moran (1986) suggests, only those components that appear to be mismatched with the user's model are identified.

The classes of entity *object* and *operation,* in the second stage of ECM, have been initially draw from Moran's (1981) Command Language Grammar. In this, Moran distinguishes *actions* and *objects.* The term *action,* however, implies a great deal more than the term *operation,* and as ECM is only concerned with the system's actions this latter term was adopted (given that the system is not ascribed intentionality, etc.—see Searle, 1979; 1980.) Nevertheless, the distinction within ECM between operations and objects is essentially the same as that drawn by Moran (1981) between actions and objects. Moreover, the distinction between concepts and symbols implicitly recognizes Moran's distinction between the semantic and the syntactic.

The classes *concept* and *symbol,* in the third stage of ECM, require greater explanation. Johnson-Laird defines a concept in the following way:

> "A concept is a mental construct—a symbolic representation—that enables you to categorize your experiences." (Johnson-Laird, 1988).

Arnheim, while discussing images, has this to say about symbols:

> "An image acts as a *symbol* to the extent to which it portrays things which are at a higher level of abstractness than is the symbol itself.." (Arnheim (1969).

The notions of concept and symbol appear to be inextricably linked, as it is commonly accepted that concepts are internal symbolic representations—they are themselves abstractions. Moreover, our concepts often defy accurate definition (Bechtel, 1988; Lakoff, 1987; Miller & Johnson-Laird, 1976; Rosch, 1975.) This is because our understandings of concepts do not rely upon straightforward definitions, as those in a dictionary might, or common elements, but upon networks of similarities (Wittgenstein, 1953.)

Consequently, when considering a mismatch between a user's and a system's concept of an object or an operation we might be dealing with a difference that is both subtle and complex. Symbol mismatches, however, are less likely to be complex. This is because we are not concerned with the internal symbolic representations of concepts that users and designers hold, only with the symbols that have been used at the *interface* to represent concepts. In short, we might expect concept mismatches to be more complex and difficult to verbalize and rectify than symbol mismatches, as the latter may often be straightforward differences in the way that a concept is named or drawn. In short, a concept is an internal representation held by the user or designer (or embodied within the system) while a symbol is an external representation—it is the currency of communication at the interface.

In summary, Moran's (1981) distinctions between actions and objects is reflected in ECM's distinction between operations and objects, while his distinction between the semantic and syntactic levels of interaction is implicit within the concept/symbol distinction of ECM. Consequently, at a theoretical level ECM has much in common with cognitive grammars in HCI, and a certain degree in common with human error classifications. ECM differs in purpose, however, from both of these traditions. Cognitive grammars are used to develop a picture, from a particular perspective, of part of a system. The intention here is to expose inconsistencies and unnecessary complexity. Embodied within ECM there is also a particular perspective on the world (i.e. in terms of operations, objects, symbols and concepts), although the purpose is to expose those system components that are mismatched with the user's model, rather than identify all of a system's components as we might if we were using a grammar.

Human error classifications attempt to classify errors with the aim of identifying causes. Likewise, ECM is aimed at identifying the causes of errors. However, while human error classifications attempt to identify the human mental processes that lead towards errors, ECM attempts to identify those system components that were misunderstood by the user. Consequently, the classifications of ECM can be directly related to system changes, whereas the classifications of human error schemes can only be related to system changes on a secondary basis.

5. 5 An Evaluative Classification of Mismatch (ECM) of User-System Errors: A Formal Definition

The classification scheme proposed here is intended as a tool for discovering the causes of dialogue failures (user-system errors) between the user and the system during learning, it is not intended as a tool to investigate overall function, task allocation or social impact issues. The purpose of the technique is to provide a workable method for detecting and classifying model mismatches. In other words, the scheme should provide a framework and terminology in which the designer can couch his observations and insights regarding human-computer dialogue failures.

In essence, the technique is a form of *evaluationfrom the user's perspective.* What is *right* and *wrong* in the system relies entirely on the user's view of the system and task. The assumption that underlies the proposed technique is the idea that mismatched model elements will be revealed as observable dialogue failures between the system and its user. In the context of usability testing; the purpose of the scheme is to provide a uniform and helpful way of thinking and talking about the different elements (causes) of user errors.

As stated earlier, ECM is a four category classificatory scheme for describing model mismatches between the system and the user. The classification process involves the observer noting the systems operations and matching these with the user's expressed intentions. However, as the proposed classification is concerned with mismatches between the machine's and the user's model of the task, it may be useful to first consider a working definition of a model.

5. 5. 1 Models

As has been mentioned earlier, users' models of tasks and systems are incomplete, unstable, confusable and parsimonious. These models are not fixed, but are ever changing as user's recruit knowledge from related domains according to the demands of the task (Barnard & Hammond, 1982). Bearing this in mind, for the purpose of this classification it is suggested that a model *is constructed by a user to represent the salient features of the interaction with the system.* A user's model may be considered as *an imprecise set of concepts and associated symbols which relate to the objects and operations of the task and system.*

5. 5. 2 Classes of Mismatch

The classificatory scheme has four stages. However, these stages are not hierarchical representations of the user interface or of human-computer interaction generally; the four stages only illustrate the stages through which the proposed classification process should proceed (see figure 5.1). The four possible classes of mismatch following the identification of a dialogue failure can be seen in figure 5.2. In the following sections the classes of mismatch will be defined and discussed. Examples will be provided to illustrate the definitions and a practical *rule of thumb* will be suggested for distinguishing the different categories of mismatch.

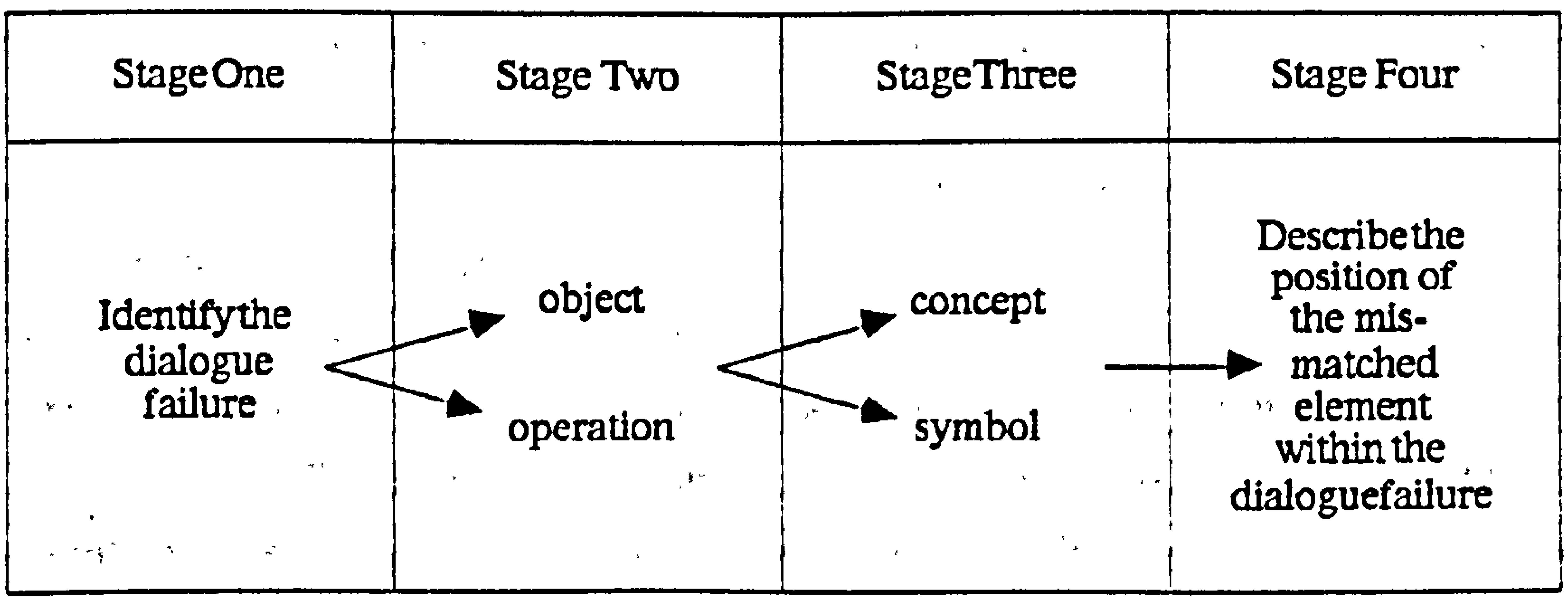


Figure 5.1: The process of classification.

	Object	Operation
Concept	object-concept mismatch	operation-concept mismatch
Symbol	object-symbol mismatch	operation-symbol mismatch

Figure 5.2: The four classes into which a mismatched element should fit.

5. 5. 3 Stage 1: Identifying a dialogue failure

The first stage of the classification process involves the identification of a model mismatch. In practical terms this means the detection of a failure in the dialogue between the system and the user. It is suggested that a dialogue failure is *a breakdown in communication between the system and the user; it is where either the computer and the user do not understand one another or some knowledge about the nature and structure of the task is not properly communicated.* This definition need not imply the communication has to cease for a dialogue failure to have occurred. The user may recover from the failure, the system may ask for clarification or the user may carry on unaware of the misunderstanding, only to discover the difficulty later in the communicative process. The emphasis is not necessarily concentrated upon those points where communication between the system and the user breaks down completely, but on misunderstandings of any magnitude.

To briefly illustrate what a dialogue failure might look like here are two examples of system functions which can lead to dialogue failures. Both of these can be found on the Hewlett Packard word processing package *Memomaker.* If the user has a document consisting of several paragraphs and chooses to re-justify the whole text, then *Memomaker* will join all of the paragraphs together as part of the re-justifying operation. Most users do not intend to join the paragraphs together, and this can be thought of as an example of a system performing an operation in a way that was not intended by the user.

The second example, taken from the same package, involves the *underline text* operation. After a user has instigated the underline operation the system displays the relevant text in italics to indicate the portion of the text which will be underlined when the document is printed. However, most users misunderstand this and believe that the text will be printed in italics and will not be underlined. This example demonstrates the possibility of dialogue failures arising as a result of the system display being misleading as far as the user is concerned.

An operational definition: Although the definition of a dialogue failure given at the beginning of this section may prove to be of use in a formal sense, in practice an operational definition may be of more practical use. Here, a dialogue failure is considered to have occurred if;

- the user reports any degree of misunderstanding during the dialogue (i.e. the system does not do what he or she wanted it to do),
- the user asks for help in any form,
- the user enters an illegal command that is not purely the result of a keystroke error, mental slip or lapse.

For this definition to be effective users must be informed at the beginning of the evaluative session that they should report any misunderstanding between themselves and the system. It is suggested that during evaluation users should be encouraged to *think out loud* and should be frequently questioned by the observer regarding their intentions.

5. 5. 4 Stage 2: Identifying an object or operation

Having identified a dialogue failure then the next stage is to identify the object or operation associated with the failure. In other words; either an object or an operation within the system which does not fit with the user's perspective.

Objects: A data file in a system, an applications package, a figure, character or window on a screen might all be considered to be objects. An object is, in essence *a thing to which something is done or about which something acts or operates,* (Oxford Universal Dictionary, 1944, third edition). An object mismatch can take the form of one of two possible types; an object-concept mismatch or an object-symbol mismatch. These terms will be defined and explained in greater detail in the section 5. 5. 5. However, to briefly summarize these types of mismatch; an object mismatch might be said to have occurred where an object is unfamiliar to a user, or has unexpected and unwanted properties, or an object appears in the wrong mode or situation as far as the user is concerned (concept mismatch), or an object is misrepresented (symbol mismatch).

Operations: An operation *is an action which is performed upon an object or objects within the system.* Saving a file, deleting a character, changing a shape in a graphics package, creating text in a word processing package are all operations. As before, there are two types of operation mismatch; operation-concept and operation-symbol mismatches. An operation-concept mismatch is where the system cannot perform the operation in the way that the user intends or an operation cannot be performed in the way that a user would like in a particular situation or mode. An operation-symbol mismatch is where an operation is misnamed as far as the user is concerned. The notions of concept and symbol will be explained in greater detail in the next section.

A rule of thumb: A general rule of thumb for distinguishing objects from operations is that an operation is something that is done to an object whereas an object has operations performed upon it. Therefore an applications package is an object while the act of loading the package is an operation. A data file is an object which when examined will be seen to contain further objects. This view necessarily implies that users never see an operation, they only observe its effects upon the objects within the system.

5. 5. 5 Stage 3: Identifying the mismatch type

Once the dialogue failure and the associate (an object or operation) of the dialogue failure has been identified then the next stage is to classify it regarding its cause. In other words, the nature of the mismatch between the system's view of the object or operation and the user's conception of the object or operation is classified as either a concept or symbol mismatch.

Concepts: For the purpose of this classification it is suggested that *a concept may be either an object or operation whether represented mentally (the user) or in lines of code (the computer).* A concept mismatch is *a fundamental difference in the understanding and representation of system or task objects or operations.*

One example of an operation-concept mismatch is to be found on the Apple Macintosh text editors, where naive users sometimes have considerable problems positioning the cursor in the desired position. When there is no text on the screen the cursor usually reverts to the nearest position on the left when the mouse is clicked instead of the position on the screen where the user clicked the mouse. Two further examples of operation-concept mismatches can be found on the Hewlett Packard *Memomaker* word processing package. Most users expect the *backspace* key to delete as well as move the cursor backwards. The key will not delete and is, in effect, only a further cursor control key.

The second example was mentioned in an earlier section and relates to the justificationoperation.Torecapitulate; if the user has a document consisting of several paragraphs and chooses to re-justify the whole text, then Memomaker will join all of the paragraphstogetheraspart of the re-justifying operation. As it is the system's concept of the operation and what it entails that is at odds with the user's view, then this is classified as an operation-concept mismatch.

An object-concept mismatch is where the object is not in a form that the user would expect. If the object were a data file then it might contain information which is unnecessary as far as the user is concerned. Alternatively, it may lack information which the user feels is important. Both of these cases would be classified as object-concept mismatches. As mentioned earlier, an example of an object-concept mismatch can be found on the ICL Perq running PNX. In order to print a file to a quality standard the user must issue a command not only to print but to create a print file (a configured file). Many first-time users have little idea as to how to create a print file and usually no idea as to why it is needed.

Symbols: The notion of matching symbols at the human-computer interface has received less attention than ideas relating to matching models generally. However, the symbols used to represent the concepts of a model are just as important to the communicative process as the concepts and structure of the model itself; a prerequisite of good human-computer communication is that the two parties to the communicative process agree terms (Mick, 1980; Spiegler, 1983) and this should include the symbols used in the dialogue. The term *symbol* is taken to mean *a word, character, sign, figure, shape or icon employed by either the user or the system to represent an object or operation within the system.* A symbol mismatch is not one of fundamental understanding, *but occurs where the computer and the user adopt different terms to represent the same concept.*

Two examples of an operation-symbol mismatch are to be found on the Prime 750 system. To print a document the user must type *spool* while the command to save a document is *file.* Most naive users appear to believe this to be an illogical choice of terms. A more extreme case is the commands needed to format and then print a file on the ICL Perq running PNX which is; 'nroff -ms *filename* > *printfilename*' followed 'lpc *printfilename*'. It was mentioned previously that the need to create a print file (a configured file) in this system could be considered as an object-concept mismatch and taken together with this possible operation-symbol mismatch, it can be seen that a dialogue failure, in this case, could arise as a result of more than one mismatch.

An object-symbol mismatch is where an object which is part of the task or system is what the user would expect if it had not been misnamed; that is to say that the object is correct but the symbol used to represent it is inappropriate as far as the user is concerned. An example of an object-symbol mismatch can be found in some library systems. Some naive users may search for a *catalogue* number for a publication only to discover later that the system they were using refers to these as *accession* numbers. Another example of an object-symbol mismatch can be found on the Hewlett Packard *Memomaker* word processing package (and was mentioned in an earlier section), where text which will be underlined when printed is displayed on screen in italics with no underline creating confusion amongst first-time users.

A rule of thumb: A dialogue failure may occur and be associated with either an object or an operation within the system, but how can symbol mismatches be practically distinguished from concept mismatches? A possible rule of thumb for distinguishing symbol from concept mismatches might be whether the users' symbols for the systems objects and operations easily map onto the designer's version of the task or system. For example; if it can be seen that the *spool* and *file* operations on the Prime 750 system mentioned earlier, are in essence, the same concepts as the user terms 'print' and 'save' then the mismatch might be considered to be of the operation-symbol variety. If terms cannot be easily substituted then the problem may be of a more fundamental kind, where the designer's and user's conceptualizations of the concepts or structure of either the system or the task (their models of each) are inharmonious.

5. 5. 6 Stage 4: Describing the position of the mismatched element

Once a dialogue failure has been identified, and its causes classified, these causes (or mismatched elements) need to be described with respect to their position within the dialogue failure. But what exactly is an element?

An element is considered to be part of a model, of either the task or system. A mismatched element is that point at which the user and the system do not agree. For example, the symbol *spool* that is used on the Prime system to represent the print operation can be considered as an element. In other words, an element is either a concept or a symbol. A model is considered, at this simple level, to consist of these elements. In other words, a model consists of a whole series of concepts of either objects or operations, and symbols representing these concepts.

When a mismatched element is described, this simply means that its contribution to the dialogue failure is considered. In the example given earlier, the symbol *spool* appears to be almost totally responsible for the dialogue failures that relate to it. However, in more complex dialogue failures a number of mismatched elements may have contributed towards the misunderstanding. If this is the case then the description of the position and role of the mismatched element is more important. The crucial question that is addressed at this stage in the classification process is: *what role did the element play in the dialogue failure?*

5. 6 Using ECM

From the examples of potential mismatches presented it should be apparent that ECM is intended as a tool to evaluate individual system operations and objects rather than evaluate overall functionality or address task allocation issues (for a summary of the classification technique, as it was presented to the individuals who were required to use it, see appendix 2). As a consequence of this piecemeal approach to evaluation it is necessary to actively search for dialogue failures and this entails setting tasks in such a way that every operation in every mode[2] in the system is used during the evaluative sessions.

It should be clear from the examples given earlier that to evaluate a system in this way requires a sound knowledge and understanding of the system. The important point about distinguishing mismatches is that to properly classify a dialogue failure the observer must not only discover what happened (i.e. which operations were instigated) but must also elicit the user's view of the task and system. For example; if a user intended an action that the system can perform then this is an operation-symbol mismatch, but if the user desired an operation which is not possible then the mismatch is more fundamental (a concept mismatch). In short; *the physical actions and consequences may remain constant, but the classification depends upon the user's view of the task and system.*

5. 6. 1 Training

Prior to using a prototype, a user will often have to be given at least some degree of training. However, there are two problems with training users before they use a machine. Firstly, if the instruction that the user receives provides information about the operation of the system which they would not acquire if they were *typical* users then there is a danger that mismatches that would occur when the system is in commercial use will not be detected. In other words; the user should not receive privileged information or a higher degree of training which would not normally be acquired by other users. The second difficulty is, in essence, the reverse of the first; if too little training is provided then it is likely that some of the mismatches detected would not occur in real life. It is suggested that those subjects who are used during the evaluation receive *only as much instruction and information as everyday users might be*

[2]Although there is a move in design away from modes Monk (1986) points out that even systems claimed to be modeless have modes. Furthermore, Young & Harris (1986) argue that modes are not necessarily unhelpful and that some systems would be unmanageable without them.

reasonably expected to acquire.

5. 7 A summary of definitions

Model: A model is constructed by a user to represent the salient features of the interactionwith the system. A user's model may be considered as an imprecise set of concepts and associated symbols which relate to the objects and operations of the task and system.

Dialogue failure: A dialogue failure is a breakdown in communication between the system and the user; it is where either the computer and the user do not understand one another or some knowledge about the nature and structure of the task is not properly communicated. In operational terms a dialogue failure is said to have occurred if the user reports any degree of misunderstanding during the dialogue, if the user asks for help in any form, or if the user enters an illegal command that is not purely the result of a keystroke error, mental lapse or error.

Object: An object is, in essence a thing to which something is done or about which something acts or operates. An object-concept mismatch might be said to have occurred where an object is unfamiliar to a user or has unexpected and unwanted properties, or an object appears in the wrong mode or situation as far as the user is concerned. An object-symbol mismatch is where the object is misrepresented.

Operation: An operation is an action which is performed upon an object or objects within the system. An operation-concept mismatch is where the system cannot perform the operation in the way that the user intends or an operation cannot be performed in the way that a user would like in a particular situation or mode. An operation-symbol mismatch is where an operation is misnamed as far as the user is concerned.

Concept: A concept may be either an object or operation whether represented mentally (the user) or in lines of code (the computer). A concept mismatch is a fundamental difference in the understanding and representation of system or task objects or operations.

Symbol: The term symbol is taken to mean a word, character, sign, figure, shape or icon employed by either the user or the system to represent an object or operation within the system. A symbol mismatch is not one of fundamental understanding, but occurs where the computer and the user adopt different terms to represent the same concept.

5. 8 Conclusion

Having decided that the use of both cognitive grammars and cognitive style measures in design and development might prove problematic, an alternative, but potentially complimentary approach, might be to focus analysis upon user-system errors. In other words, to develop formalisms aimed at modelling the user by analyzing the results of interaction. Despite the potential drawbacks to such an approach, it would appear to offer a means by which the user might be modelled in a way that is appealing both theoretically and practically. The theoretically attractive aspects of the proposed approach are that it focuses attention upon actual mismatches that occur between the user's and the. designer's model of the task and system, rather than predicted mismatches that might be present. The practical appeal of considering user-system errors is that it might easily fit into present design and development practice. Furthermore, it might help to focus attention upon those parts of a design that require attention.

The criteria that a user-system error classification scheme should meet have been outlined as a general guide as to the nature of such a scheme. These criteria, however, are not the central focus of concern, although they will be referred to in the reports of studies in later chapters; they provide only an outline of requirements. Of greater interest is the question of whether an approach that concentrates upon user-system errors is likely to be worthwhile.

A scheme for classifying the causes of these errors has been proposed and explained. This scheme might have been considerably more complicated. However, a more complicated scheme might not transfer readily from one system to another. In essence, a scheme has been suggested that is fundamental (i.e. considering concepts and symbols), but also relatively straightforward, so that it might be easily applied.

The purpose of this scheme was to act as a vehicle to investigate the possibility of using such classifications as a means to understanding user behaviour at the human-computer interface. The next step in this investigation was, then, to consider whether the scheme might be applied in a slightly more formal setting, via a pilot study (described in the following chapter). The purpose of this study was threefold. Firstly, it was a further initial test of the scheme, following the informal trials. Secondly, it provided some indications as to whether the scheme might meet the criteria set out in the previous chapter regarding evaluative classification schemes for user-system errors. Thirdly, it provided a means by which the range of user-system errors might be considered. A potential problem for evaluative classifications of user-system errors is thateach individual might produce a unique set of user-system errors, thus making an analysis pointless. In order to justify an analysis of user-system errors there needs to be a common core of errors that affect all users or at least a large proportion of users. The purpose of the pilot study was to look for this common core of user-system errors.

Chapter 6

A Pilot Study of ECM

6.1 Overview

In the previous chapter a classification scheme, ECM, was suggested and defined, as a vehicle for the investigation into using evaluative classification schemes. The purpose of the pilot study reported here was to provide indications as to the feasibility of employing user-error classification schemes as a means for considering user behaviour at the human-computer interface. As part of this overall objective, ECM was considered against some of the criteria for evaluative classification schemes suggested in chapter 5. Moreover, this pilot study was used to provide information on possible common cores of errors (errors that occur to almost all users, or at least significant groups of users).

The results indicated that it should be possible to employ ECM, hence evaluative classification schemes, as a means of understanding user behaviour at the interface. The scheme appeared to be *comprehensive* in its classification and provided information that was both *specific* and *detailed* (see chapter 5 for the six criteria). Overall, the results of the pilot study indicated that a further consideration of user-system error classification schemes (i.e. a test of the usability of ECM) was worth pursuing.

6. 2 Introduction

In the last chapter the development of an evaluative classification scheme for dealing with user-system errors was described, and formally defined. It was suggested that this scheme might serve as a vehicle for research into the potential use of such schemes for considering user behaviour at the human-computer interface. The ECM scheme was refined during informal trials, and the purpose of the pilot study reported in this chapter was to assess whether the scheme might prove useful and usable. In essence, the purpose was to establish whether further research was likely to prove fruitful.

6. 2. 1 The three criteria

In chapter 5 six criteria that an evaluative classification of user-system errors would need to meet were outlined. Here, the three criteria that are relevant to the pilot study will be re-stated in practical terms.

Comprehensive: In chapter 5 the term *comprehensive* was taken to mean that *it should be possible to classify all user-system errors under one scheme.* In other words, for the pilot study all dialogue failures should be classifiable using ECM, there should be no *unclassifiable* mismatches.

Specific: By *specific*, what is meant is that ECM should identify points in the design or dialogue where problems occur, and thus where change and improvement might be needed. It was expected that mismatches could be associated with certain objects and operations in the system. Failing this, ECM should be able to identify groupings of mismatches associated with certain classes of operation or object.

Detailed: Once the associate of a dialogue failure is identified then ECM needs to provide detailed information about the associate. In other words, having shown that there is something amiss with either an object or operation within the system, there is then a need for a detailed description of what is wrong. This means that the misunderstanding of an object or operation is classified according to whether the user and the system do not agree on the *concept*, or the *symbol* of the object or operation. It is this part of the classification that is expected to provide the detail necessary to suggest changes to the computer system being evaluated. Therefore, the test of the detailed nature of ECM is whether it is possible to identify *how* the system's objects and operations differ from the user's view of the task.

To recapitulate; the purpose of the pilot study was to test whether ECM was *comprehensive* and whether it provided information about a system which was *specific* to certain points in the design and *detailed* how these points differ from the user's model of the task. In short, it should be possible to classify all dialogue failures using ECM and it should be possible to provide specific recommendations for changing the product under review, if the scheme is comprehensive and provides detailed and specific information.

These criteria, however, were not the only interest of the study. one further issue was also considered to be important. There was a question as to whether a significant proportion of mismatches might be associated with differences between companies in terms of their conventions regarding their products. For example, company A might producea system where the user must always use a scroll-bar to view text on a word processor while the products of company B might require the user to employ control keys to scroll. It can be seen that the mismatches that arise as a result of these differing conventions will depend upon the system the user has previously experienced. This type of problem lies with the differences in conventions and the value of charting these mismatches might be seen as doubtful.

It was anticipated that this argument would not prove to be a valid criticism of the use of evaluative classification schemes on the grounds that some mismatches will be common to all users regardless of their background (although a some may be backgrounddependent). Furthermore, because the number of mismatches resulting from differences in company conventions were expected to comprise only a small proportion of the total number of mismatches identified.

Some mismatches, however, may be peculiar to certain individuals, and some may be related to the user's background. The important point is that there should be a *common core* of user-system errors that are common to all, or at least common to certain groups. Without this common core of mismatches there is a danger that altering the system to accommodate one user will disadvantage several others. In essence, we might expect that only those common core mismatches, or user-system errors whose "fix" would not affect other users, might be addressed in an iterative cycle.

6. 2. 2 Researchquestions

1 Will it be possible to classify all dialogue failures identified during the evaluative sessions using ECM (the *comprehensive* criteria)?

2 Will the process of classifying dialogue failures under ECM pinpoint objects and operations within the system that do not agree with the user's perspective (the *specific* criteria)?

3 Will the process of classifying dialogue failures under ECM suggest how the mismatched objects and operations differ from the user's model of the task (the *detailed* criteria)?

4 Will a *common core* of mismatches emerge, despite the widely differing background of the subjects?

6. 3 Method

6. 3. 1 Design

A between-subjects design was adopted which involved one group of staff from the Polytechnic, one group of first year students and two other individuals. This allowed differences between these two groups in terms of numbers and types of mismatches to be examined. The variables were as follows:

1 Background; either staff or student (IV). The criteria that was considered to be most important was that all of the staff had considerable experience of several computer systems while subjects from the first year BA Computing in Business degree course all had very limited experience.
2 The numbers of dialogue failures (DV) that occurred during the evaluative sessions for each subject.
3 The numbers of different types of mismatch (DV) that occurred for each subject during the evaluative sessions.

6. 3. 2 Subjects

As has already been mentioned, the subjects in the *staff* category all had experience of several computer systems while the subjects in the *student* category had only limited experience of a package called *Open Access* running on IBM PC compatible Hewlett Packard Vectras. Four members of staff were aged between 30 and 40 years while the remaining member was a technician aged 22 years. All members of staff were male while two of the five students were female. Of the students, all were aged between 19 and 22 years. The other two subjects in the experiment comprised a computer-naive postgraduate student aged 22 years and a second year HND student aged 21 years. The latter had experience of several computer systems.

6. 3. 3 The Task

Subjects were required to perform twelve tasks on the Hewlett Packard *Memomaker* word-processing package running on an HP 150 touchscreen PC. With the exception of the first task, all of the tasks related to a piece of text which can be seen in appendix 3. The exact instructions the subjects were given can also be seen in appendix 3. These tasks were as follows:

1 To type a brief letter.
2 To replace two words in a paragraph with another word.
3 To insert a paragraph at the beginning of a piece of text.
4 To adjust the text so that two paragraphs were joined to form one large paragraph.
5 To underline the title at the beginning of the text.
6 To change the margins so that they were only four inches apart.
7 To re-justify the whole text.
8 To replace a word in one paragraph with another word.
9 To center the title at the beginning of the text.
10 To remove the two words from the text.
11 To move the title at the beginning of the text back to the left of the screen.
12 To move a paragraph to the end of the text.

6. 3. 4 Procedure

First, subjects filled in a short questionnaire concerned with their experience of Hewlett Packard Computers and following this, were read the instructions shown in figure 6.1.

The purpose of this experiment is to evaluate the performance of the *Memomaker* word-processing package. During the experiment you will be asked to complete twelve tasks using the package. Please try to think out loud and explain what you are doing and what you expect to happen as you perform the tasks. After each task you will be asked to comment on how easy *Memomaker* is to use.

If at any time you do not understand anything about the machine please use either the help system, the manual or the quick reference guide. However, if you become stuck then please ask what to do. If you cannot perform any part of the task easily on *Memomaker* or the machine does anything you do not expect then please inform me. It is important to report any misunderstanding between yourself and the system. From time to time you may be asked to stop what you are doing and to explain your intentions, when this happens please do not continue until asked to do so.

Figure 6.1: The instructions that were read to the subjects.

During the evaluative sessions any signs of dialogue failure were noted by the evaluator. Users were asked to explain their intentions as well as how the system differed in any way from what they wanted or expected. Any misconception of the system that was revealed by the user's explanation was noted as a dialogue failure (misunderstanding). Following this, that part of the system that had been misconstrued was explained to the user. They were then asked for further comments. At the end of the experimental session the users were asked to comment upon the system, the tasks and anything else they considered relevant. Following this, subjects were debriefed. The length of any session varied between forty five and ninety minutes.

6. 4 Results

Each research question will be dealt with in turn. The list of all mismatches, together with a record of which subjects experienced these mismatches, can be seen in appendix 4.

6. 4. 1 Research question 1

The first research question asked whether all mismatches would be classifiable under ECM. It was found that for a large proportion of mismatches the process of classification appeared to be *easy* and *obvious*. A small proportion (less that 20%), however, required much more thought before a classification could be decided upon. Therefore, the results of the pilot study appear to suggest that ECM is *comprehensive* in its classification of dialogue failures, although it should be remembered that this was the author's subjective impression.

6. 4. 2 Research question 2

The second research question asked whether the process of classifying dialogue failures under ECM would pinpoint objects and operations in the system that do not agree with the user's perspective. In other words, in each case it will be possible to identify an object or operation that is the root cause of a mismatch.

In total there were seventy-five mismatches distinguished during the first study and many of these mismatches were experienced by more that one subject (mean = 3.01, median = 2, SD = 3.03). Of the 75 mismatches identified, 39 were experienced by more than one subject. For all of these seventy-five mismatches an object or operation was identified as the associate of the dialogue failure. In other words, in no case was an object or operation not identified. Therefore, ECM does appear to be *specific* in its classification, according to the definition suggested in chapter 5. Although, again, it must be remembered that the author was responsible for the classification of the mismatches, and so this conclusion, along with any others, must be treated with caution.

6. 4. 3 Research question 3

The third research question asked whether the process of classifying dialogue failures under ECM would suggest how the mismatched objects and operations differed from what the user either expected or wanted. In other words, would it be possible to identify a *concept* or its *symbol* that are at odds with the user's understanding. All of the seventy-five mismatched objects and operations identified during the evaluative sessions were classified according to whether they differed from what the user expected or wanted in terms of their *concept* or their *symbol*. No mismatch could not be classified according to its type. Therefore, the results of this pilot study appear to suggest that ECM is detailed in its classification, and that this is consistent with the definition of *detailed* given in chapter 5.

	Students n = 5	Staff n = 5
operation-concept mismatch	38	52
operation-symbol mismatch	32	34
object-concept mismatch	5	5
object-symbol mismatch	11	15
Total	86	107

Figure 6.2: The frequencies of different categories of mismatch. The two subjects who were not part of either of these groups have been excluded from this figure.

6. 4. 4 Research question 4

The fourth research question asked whether there would be any significant differences between experienced and inexperienced users in terms of the numbers of different types of mismatches. The results can be seen in figure 6.2. As expected, none of these results are significant. Consequently, it appears as though the background of the subject (the subject's experience on other systems) does not significantly influence the numbers of different types of mismatch. In other words, those subjects with experience of computer systems produced by companies other than Hewlett Packard did not experience a significantly greater of lesser number of mismatches than subjects without this experience.

Nevertheless, although it appears as though the the two groups, shown in figure 6.2, do not differ in terms of overall numbers of errors or numbers of different types of error under the classification scheme, there was still the possibility that those subjects with experience of other systems might have experienced greater difficulty with just parts of the *Memomaker* system. However, when mismatches are grouped around central system concepts (see appendix 4), then it can be seen that these mismatches were not as isolated as they might at first seem. The results regarding the differences in the number of mismatches between those with and those without significant experience of other systems can be seen in figure 6.3. Of those results that are possible to analyse, none are statistically significant. In other words, there are no significant differences between subjects of differing experience, in terms of the numbers of mismatches they experienced, when these mismatches are grouped around central concepts which relate to the system's structure and function.

	Students n = 5	Staff n = 5
Deletion	7	9
Word-wrap	11	7
Underline	7	6
Joining paragraphs	2	8
Margins	15	18
Center line	5	5
Selecting text	8	4
Justifying text	10	17
Menu	8	10
Insert	3	4
Copy/cut & paste.	3	5
Total	79	93

Figure 6.3: The frequencies of mismatches grouped around system concepts. The two subjects who were not part of either of these groups have been excluded from this figure.

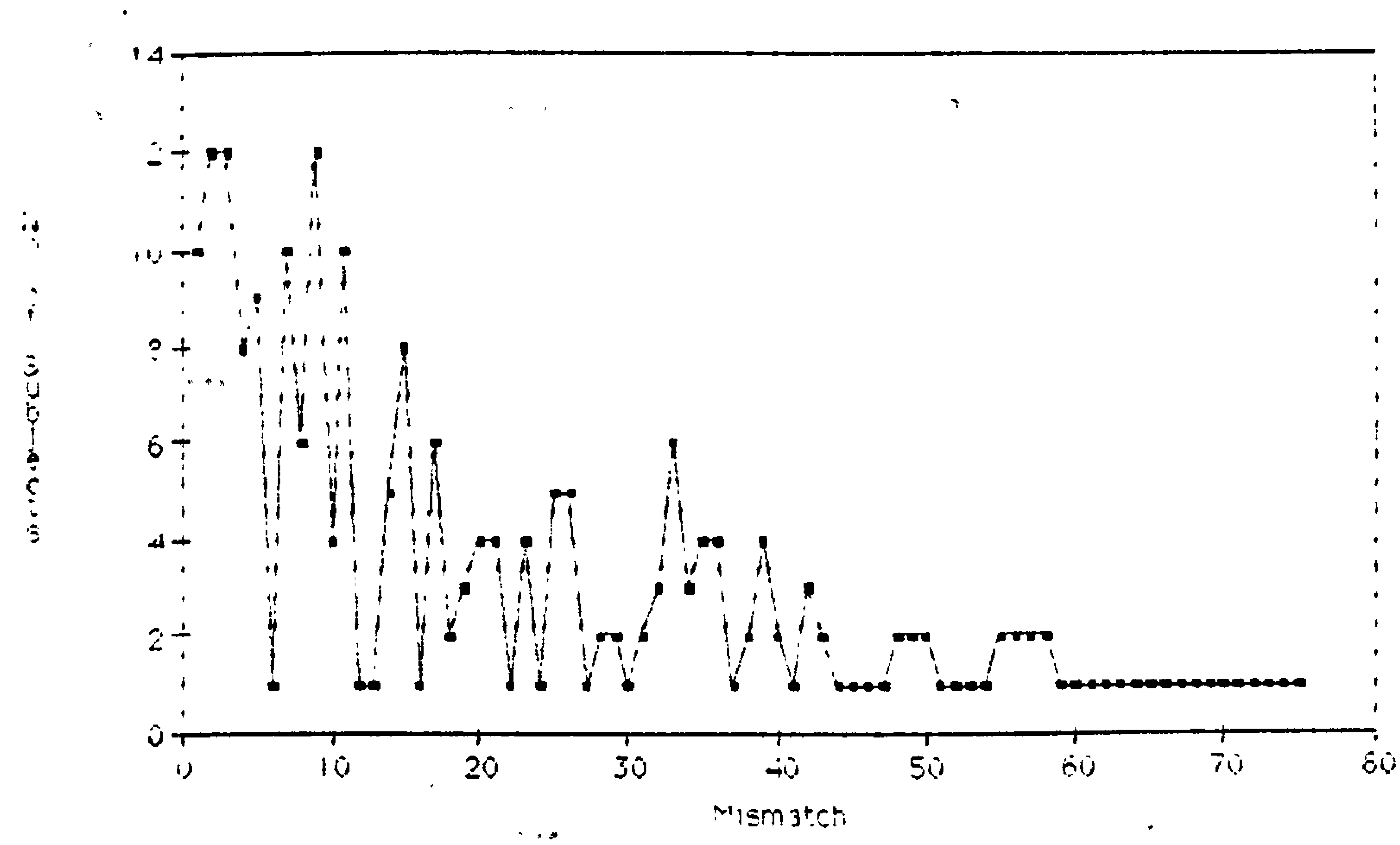


Figure 6.4: The chart shows each mismatch (numbered from 1 to 75) against the number of subjects who experienced the mismatch. Towards the end of the experimental sessions subjects would demonstrate mismatches that had already occurred, and so the frequencies of these earlier mismatches would rise. However, the new mismatches that these subjects demonstrated were more likely to be unique to that individual (as the common ones had already shown up), and this accounts for the trailing-off in the chart towards the later mismatches.

Nevertheless, this still leaves the question of whether there were *common core* user-system errors, that might justify a consideration of such errors. Even though the subjects, as a group, were from diverse backgrounds, there would still need to be a common core of mismatches to justify changing any system, as many systems have just such a diverse set of users.

As can be seen in figures 6.4 and 6.5, many user-system errors were peculiar to just one or two individuals. However, there was a common core of mismatches as 23 user-system errors were experienced by 4 users or more, while 7 of these mismatches were experienced by 10 subjects or more. In short, there was a common core of user-system errors found for the *Memomaker* system, despite the varied and diverse backgrounds of the users.

Some examples may be of interest here: Mismatch 1, where the user expected the "backspace key" to delete, was found to be common (10 subjects). Mismatch 9, where text that is to be underlined when printed appears in text on the screen confusing users, was very common (12 subjects). Mismatch 33, where subjects complained that the "centre-line" command should be in one of the sub-menus, and not as part of the main menu at the bottom of the screen, was experienced by half the subjects (6 subjects).

Mismatch 50, where the subject experienced difficulty with the "CAPS" key, not realizing that it switched capitals on and off rather than allowing them to be produced only when held down, was comparatively uncommon (2 subjects).

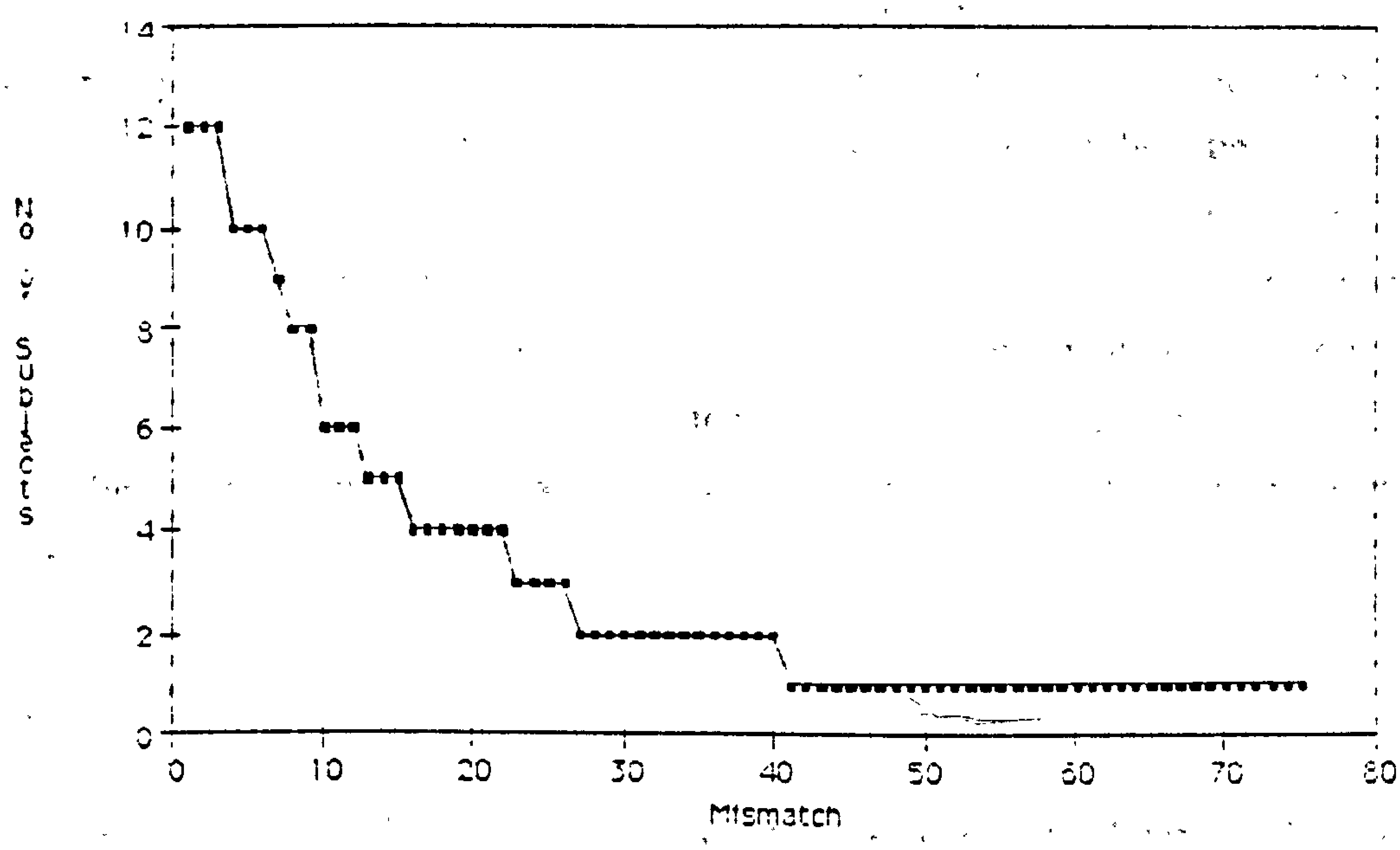


Figure 6.5: The chart shows the mismatches arranged with the number of subjects who experience each mismatch in descending order.

6. 5 An analysis of the *Memomaker* word processing package

It was suggested earlier that, if ECM did provide information that was truly detailed and specific, the it should be possible to derive recommendations for change for the system in question. In this section such recommendations have been provided to demonstrate the potential utility of evaluative classifications and the types of inferences that might be drawn when using them. Of course, this set of recommendations, produced by the author, cannot be considered as support for the argument that evaluative classifications can produce worthwhile information regarding user behaviour, only as an indicator of the potential use of such techniques.

The major causes of mismatch using the *Memomaker* word processing package were as follows:

6. 5. 1 Deletion

The major cause of mismatch with respect to the deletion facilities was the operation of the *backspace* and *deletechar* keys. Both of the mismatches associated with these operations were operation-concept mismatches. In other words, the operation did not occur as the user wanted or expected. Ten of the twelve users were surprised to find that the *backspace* key did not delete as it moved backwards while four users expected the *deletechar* key to delete text to the left rather than the right of the cursor.

Recommendation: The *deletechar* key should be supplemented by the *backspace key* which should cause text to be deleted as it moves back along a line.

6. 5. 2 Word Wrap

Eight of the twelve users did not expect text to wrap around onto the next line when the cursor reached the end of the present line. This is an operation-symbol mismatch as users were pleased with this facility, only the system had not made them aware of it. If text on a line was being deleted however, the text on the next line would not *wrap back* to fill the space (an operation-concept mismatch) which eight out of the twelve users expected to happen. Furthermore, although the text the cursor will wrap around while text is being written, during ordinary movement of the cursor it will not wrap around on to the next line (an operation-concept mismatch).

Recommendation: The wrap around facility should be signaled to the user in some way. Text should wrap-back as well as forward and the cursor should be allowed to wrap around at all times.

6. 5. 3 Underlining

Two users were unaware of the underline operation (an operation-symbol mismatch). All twelve users were confused when text which had been underlined was displayed on the screen in italics (an object-symbol mismatch).

Recommendation: Users should be made aware of the facilities the package has to offer and text which has been underlined, and will be printed as underlined, should be displayed on the screen as underlined.

6. 5. 4 Joining Paragraphs

The method of joining one paragraph to another was completely unclear to all users and no user managed to join paragraphs together during the evaluative sessions. Six of the twelve users tried to use the *delete char* key to join the paragraphs together (an operation-concept mismatch). Several users mentioned that they would have tried to use the *backspace* key to delete the carriage return, but knew already that *backspace* would not delete.

Recommendation: It should be made possible for paragraphs to be joined together using the backspace key/operation.

6. 5. 5 Margins

All twelve users encountered difficulties because they were unaware of the positions of the margins (an object-symbol mismatch) together with the problem that the default right margin position is too far to the left (an object-concept mismatch). Nine of the twelve users encountered difficulties when they could not alter text which was shown on the screen but that was outside of the margin (an operation-concept mismatch). Two users thought that the command *set tab* would reset the margin (an operation-symbol mismatch) while a further two users found the relationship between setting the margin and using the cursor confusing (an operation-concept mismatch).

Recommendation: Margins should be marked in some way other than the shading of text. The default margins should be set to allow text to be typed in with approximately 2.5 cm of margin to both the right and the left. Text should automatically re-justify to fit within the margins, while the process of changing the positions of the margins needs to be simplified or at least made more obvious.

6. 5. 6 Centre line

Two users were unaware of the *centerline* facility (an operation-symbol mismatch) while two more were unaware of the relationship between the center line facility and the cursor position (an operation-concept mismatch). Four users expected the center line command to be matched by a move text to the left command (an operation-concept mismatch).

Recommendations: Users should be made aware of the systems facilities, as has already been mentioned. Rather that one command there should be a series of commands to move text to the center, left and right of a line.

6. 5. 7 Selecting text

Four users were confused by the term *define block* when the system meant select the text (an operation-symbol mismatch). Two users found the process of selecting text confusing. Once an operation which involves selecting text has been evoked then the selection of text begins from where the cursor rests. If users wish to begin selection at a different point then they must cancel the operation, move the cursor and then evoke the operation again (an operation-concept mismatch). Three users did not select all of the text when cutting out a paragraph. In other words; it was not clear to these users that they needed, not only to select the text, but also to select the invisible carriage return at the beginning of the next line (an object-symbol mismatch).

Recommendations: The instructions the system gives to the user regarding the selection of text need to be changed. Users should be able to begin the selection of text from any point they wish while objects such as carriage returns should be visible for users to see when they are selecting text.

6. 5. 8 Justifying text

All twelve users wanted the text to automatically re-justify as operations are performed (an operation-concept mismatch). Furthermore, all twelve users did not expect the *align* operation to join all of the paragraphs in the selected text together (an operation-symbol mismatch). Two users did not expect the *align* operation to move the heading of the text to the left of the line (an operation-concept mismatch). Two users did not realize that blank areas counted as text when aligning text (an operation-concept mismatch) while four users thought that the *justify* key on the screen was an operation rather than an option (an operation-symbol mismatch).

Recommendation: Text should automatically re-justify after every operation while the re-align operation should not join paragraphs together. The command for re-aligning text should be re-named justify and all invisible text (eg blank spaces) should be made visible during selection of text.

6. 5. 9 Menus

Several users thought that the *underline* operation, the *align* operation, the *center line* operation, the *insert* mode, the *set margin* operation were in the wrong menu (operation-symbol mismatches). Furthermore, the command to return to the main menu was not considered to be clear or obvious by users (an operation-symbol mismatch).

Recommendations: The menu structure and menu names need to be re-thought and changed.

6. 5. 10 Insert

Four users were unaware that the *insert char* key selected a mode and was not an operation (an operation-symbol mismatch).

Recommendation: The *insert char* key should be re-named (eg *insert mode on-off*).

6. 5. 11 Cut, copy and paste

Five users thought that the command to *cut* text would destroy rather than remove the text for later use (an operation-symbol mismatch). Two users did not realize that text would be pasted back onto the document from where the cursor was positioned within the text (an operation-symbol mismatch).

Recommendation: The operation cut should be re-named or explained to the user. The relationship of the cursor position to the paste operation should be made more obvious or explained to the user.

6. 5. 12 Beginning on *Memomaker*

Two users believed that to start using the word processing package a file would need to be created (an operation-concept mismatch).

Recommendation: Users should be signaled as to where they are within the system at all times as well as what is expected of them during use of the package.

6. 5. 13 The *CAPS* key

Two users did not realize that this key selected a mode rather than evoked an operation (an operation-symbol mismatch).

Recommendation: The *CAPS* key should be re-named (eg *caps lock* or *caps mode on-off*).

6. 5. 14 The "beep"

Four users were confused by the beep the machine made as the cursor approached the end of the line. These users stopped in the belief that they had *done something wrong* (an operation-concept mismatch).

Recommendation: The system should be changed so that the beep is normally turned off.

6. 5. 15 Summary

A number of recommendations have been made using the information produced by using ECM. Although it is not possible to test whether these operations might improve the design of the *Memomaker* word processing package, they do provide an indication that the process of using ECM might provide information that can be used to suggest improvements to the design of the system being evaluated.

6. 6 Conclusion

This pilot study has provided indications that evaluative classification schemes, such as ECM, may be of some use as a means of considering user-system behaviour. It appeared as though ECM was comprehensive, in that no mismatches were left unclassified. Furthermore, the scheme also seemed to fulfill the criteria of providing detailedand specific information as it was possible to identify specific points in the design that were not in accordance with the user's view of the task. In other words, for every dialogue failure either an object or an operation within the system was distinguished as being at fault. Moreover, not only were these points in the design identified, but the type of mismatch was also suggested in each case. That it to say, that how the object or operation differed from the user's model of the task was identified. Accordingly, recommendations for system change could be drawn as a result of employing the scheme.

There were no significant differences between users with different backgrounds and general experience in terms of the numbers and types of mismatch that they experienced. Although some differences between experienced and inexperienced users appear to exist, it appears as though these differences are small when compared to the gulf between the designer's and the user's model of the task.

Most importantly, the study demonstrated that there were likely to be common cores of user-system errors for systems, even when the background and experience of the users concerned was diverse. The results of this study must be treated with considerable caution, as only the author classified the mismatches, and all of the results are based upon these classifications. Unfortunately, independent referees who might have judged these classifications were not available. This was because it was not possible to find people with sufficient experience of the word-processing package used.

Nevertheless, despite the caution with which these results must be treated, this extended pilot study provided an initial indication that it was worthwhile to progress to a more detailed study of the usability of the ECM scheme. This next study is reported in the following chapter.

Chapter 7

An Investigation into the Usability of ECM

7. 1 Overview

In the previous chapter a pilot study was reported, the results of which suggested that common cores of user-system errors are likely to occur, and that ECM can be used to classify these dialogue failures. In this chapter a study into the usability of the ECM scheme is reported. The overall aim of this study was to assess whether an evaluative classification, such as ECM, might be usable by people other than the author, and provide information that might be perceived as being of use by designers. A further point of interest was whether ECM would prove to be generalizable from one system to another, having only been used to classify user-system errors from the *Memomaker* system.

ECM was used to classify user-system mismatches that occurred with a system that was under development by a design team working for Hewlett Packard Ltd, at Pinewood, Wokingham. The results of the study indicated that the scheme is both usable and generalizable. That is to say, that it was possible for designers to use the scheme, and their views were generally favourable towards it. Moreover, there was a moderate to high level of agreement between designers on their classifications of user-system mismatches. Furthermore, as user-system mismatches were classifiable under the scheme for the *Newwave* system it was concluded that ECM is generalizable from one system to another.

The pilot study of ECM provided indication that it was *comprehensive, specific,* and *detailed.* This second study, however, was viewed as a further and more rigorous test of these criteria. The major test of these criteria was the designers opinions of the usefulness of the classification scheme. More specifically, the designers were asked whether the scheme was comprehensive, specific, and detailed. The evaluative scheme scored positively on all of the criteria.

Furthermore, evidence from the transcript of the group discussion of the classifications of the user-system mismatches indicated that the use of the classification scheme, and the discussion of problems in terms of ECM, encouraged designers to explore user-system mismatches and to agree understandings of problems, and well as solutions to these problems. However, one drawback of the scheme was that it was possible to legitimately classify a problem in a number of ways, without there being a corresponding number of elements (or causes).

7. 2 Introduction

In the last chapter a pilot study was described and its results reported. Overall, these results provided indications that further investigation of the of evaluative classification, via the ECM scheme, was likely to be worthwhile. The two major criteria that this study was intended to consider ECM against were whether ECM was *usable* and whether it could be *generalized* from one system to another. The three principal measures of whether ECM is *usable* are; whether the designers *can* classify problems using the scheme, whether designers *do* classify problems using ECM, and the designers opinions as to the overall usability of the scheme. A further test of the usability of the classification scheme was whether the classification of problems was *natural* and *obvious* to the designers. The measure of this was the extent to which the designers agreed upon each classification before the discussion.

In the previous chapter, where a pilot study of ECM was described, it was concluded that it was possible to classify the dialogue failures that users experienced using the Memomaker package. However, if the classification scheme is to be of general use, we needed to discover whether it was possible to classify problems on other systems besides this word-processing system. In other words, we needed to know whether ECM is *generalizable* from one system to another, or whether it can only be used on some systems. The test of the scheme against this criterion was whether designers can classify user-system mismatches for the *Newwave* system using ECM. Furthermore, the designer's opinions of whether the scheme is likely to be *generalizable* to other systems provided a further indication of the likely usefulness of ECM in the longer term.

As stated above, the main purpose of this study is to consider ECM against the criteria of *usability* and *generalizability*. However, the study also provides a further test of the criteria considered in the pilot study. These criteria were; whether ECM was *comprehensive* in its classification, whether the classification process identified *specific* points in the design that required alteration, and whether the process of classification provided information that was *detailed* enough for the designers to suggest changes to the system. The major test of ECM against these criteria was the designers' opinions of whether the scheme was *comprehensive*, identified *specific* points in the design that required change, and provided *detailed* enough information that suggested *how* the system might be changed.

While it may be useful to consider ECM against the criteria set out in chapter 5 for evaluative classifications of user-system errors, there were further concerns that were addressed in this second study of ECM. In particular, the role the classification scheme might play within iterative design. For example, will the use of the scheme encourage a design team to explore the user's problems with a system in detail? Will the process of agreeing classifications of problems cause various members of the design team to alter their views of the problem? Furthermore, will the process of using the classification, and in particular of discussing these classifications, help the design team to identify solutions to the user's problems? These three questions are central to whether the classification scheme is likely to be of practical use during iterative design.

The purpose of this study is to go some way towards answering the questions that have been set out above. The major concern that is addressed here is; *can ECM contribute towards the process of iterative design by helping designers to more accurately identify the causes of the problems that users experience?* In other words, *is an evaluative classification of user-system errors a useful means by which we can consider user behaviour?*

This study, however, can only partially answer this question. While we might decide that ECM does help the designer to identify causes of user-system mismatches or dialogue failures, we cannot know, from this study, whether ECM helps designers to *accurately* identify the causes of the problems that users experience. If this process is inaccurate then we might expect the design that has been changed following an evaluation using ECM to be no better, or even worse, than the original design. If the use of ECM helps designers to accurately identify problems then we might expect a changed system to show an improved performance over the original system. This question of accuracy, however, will be answered by the next study, described in chapter 8.

7. 2. 1 Research Questions

The principle questions this study was aimed to answer were as follows:

1 Does ECM fulfill the criterion of *usable,* set out in chapter 5? In other words, can ECM be successfully used by people other than the author, particularly designers?
2 Do designers consider ECM to be *usable?*
3 Do the designers agree on their classifications?
4 Does ECM fulfill the criterion of *generalizable*, set out in chapter 5? In short, can ECM be used to classify the mismatches from a system other than that used in the pilot study?

5 Do designers consider ECM to be *generalizable?*

6 Do designers consider ECM to be *comprehensive?*

7 According to designers, does ECM provide *specific* information? In other words, does the use of ECM show up those points where improvement is needed in the opinion of designers?

8 According to designers, does ECM provide *detailed* information? In other words, does ECM help to expose *what* is wrong with any particular point in the design, and *how* it might be changed?

9 Does the process of discussing ECM classifications encourage the design team to explore the user's problems in detail?

10 Does the process of discussing ECM classifications cause various members of the design team to alter their views of the problem?

11 Does the process of discussing ECM classifications help the design team to arrive at solutions to the user's problems?

7. 3 Method

Before the method is described in detail it may be useful to consider an overview of the stages that comprised this study.

The author familiarized himself with the system under development.

Usability studies were used to formulate problem descriptions

Each member of the design team classified the problems using ECM

The design team met to discuss the problems and the classifications

Each member of the design team was interviewed

The transcript from the discussion was studied by independent judges.

Figure 7.1: The six stages of the study.

In the first stage the author familiarized himself with the software package under development. Following this a number of usability tests were performed. The user-system errors or mismatches that occurred were then used to provide a set of problem descriptions for the later stages. In the third stage the design team were each asked to use the ECM scheme to classify the user-system mismatches set out in the problem descriptions.

For the fourth stage the design team met for two hours and discussed each problem description and its possible classifications, with the overall aim of agreeing on the classifications of the problem. Directly following this meeting, and for the fifth stage, each member of the design team was briefly interviewed. The sixth and final stage involved the use of data from the discussion in stage four. A transcript of the discussion was studied by five independent judges who marked where members of the design team were apparently agreeing with one another, where they were disagreeing, and where individuals appeared to change their opinions. A diagram representing this six stage process can be seen in figure 7.1.

7. 3. 1 Design

Although this study involves no comparison between an experimental and a control group, it may, nevertheless, be helpful to outline the major measures (what we might have called the dependent variables) that will be used to answer the research questions stated in the introduction.

1 Designers' classifications of the user-system mismatches. Not only can the number of classifications be used, but also the relative numbers of different classes of mismatch, as well as the agreement between designers as to the classification of the causes of any particular problem.
2 Designers' opinions of the ECM scheme, its learnability, and usefulness.
3 The design teams discussion of the classification of problems, as interpreted by five independent judges.

7. 3. 2 Subjects

All of the subjects of the study were full-time employees of Hewlett Packard Ltd. The subjects who took part in the study were as follows: 3 male software engineers, all aged between 20 and 25 years, 1 male technical writer aged between 25 and 30 years, 1 male senior learning products manager aged between 30 and 35 years, and 2 human factors engineers. Of the two human factors engineers, one was male and aged between 20 and 25 years, while the other was female and aged between 25 and 30 years. All of the subjects had been employees of Hewlett Packard for between 1 and 10 years.

The independent judges who interpreted the transcript from the design team's discussion were as follows; 2 male research assistants, one aged between 30 and 35 years and one aged between 20 and 25 years, 1 male technician, aged between 20 and 25 years, 1 male lecturer, aged between 25 and 30 years, and 1 female psychology research student from Manchester University, aged between 20 and 25 years. With the exception of the research student, all were employees of Huddersfield Polytechnic. None of the judges had any formal or informal contact with Hewlett Packard. All had either no knowledge, or limited knowledge of the ECM scheme, and none of the judges had any experience or knowledge of the Newwave system under consideration.

7. 3. 3 Materials

The problem descriptions were supplied to the designers in the form of a questionnaire. Together with this questionnaire was a description of how to classify the causes of problems using ECM. This document can be seen in appendix 5. The questionnaire that was used as a basis for the short interviews that were conducted with each subject directly after the design team's discussion can be seen in appendix 6. The transcript of the design team's discussion, that was used by the five independent judges, can be seen in appendix 7, while the instructions and questionnaire issued to the judges can be seen in appendix 8. The design team's discussion was recorded, with the full knowledge of the design team, using a "Sony Walkman" tape recorder.

7. 3. 4 Procedure

While there is no experimental procedure in the strict sense of the word, there was a process and a discrete number of stages through which the study proceeded. These stages have already been outlined at the beginning of this method section. Now these stages will be explained in a little more detail.

Initially, the author familiarized himself with that part of the Newwave system that the design team were in the process of developing. This involved approximately one day of experimenting and playing with the prototype system. Approximately one month later the author took part in the usability testing of the prototype. In the usability testing sessions users, most often from employment agencies, were shown how to use the system and then asked to a perform a series of tasks. The users were video-taped as they attempted to use the system. Over a four-day period the author observed two sessions (two users) that were conducted by a Human Factors engineer, and conducted a further two himself, where subjects were questioned as to their intentions and understandings when *errors* occurred.

These usability testing sessions were then used to construct problem descriptions. In each problem description a dialogue failure was reported. Twenty of these problem descriptions, together with instructions on how to apply the ECM scheme and examples of classification, were provided in the form of a questionnaire (see appendix 5). These questionnaires were distributed to the subjects outlined in the earlier section.

Once the subjects had completed the questionnaire, and classified the problems described in the questionnaire, a meeting was arranged where the author was present. The purpose of this meeting, as it was explained to the subjects, was to agree upon classifications of the problems described in the questionnaire. This discussion was tape recorded, and the transcript of the discussion can be seen in appendix 7.

Immediately following the meeting all of the subjects were interviewed. This brief interview was based upon the questionnaire shown in appendix 6. All of the subjects were interviewed within two hours of the meeting ending.

The transcript of the design team's discussion was then given to five independent judges, together with a questionnaire. This questionnaire, together with the instructions to the judges can be seen in appendix 8. The judges were asked to read the transcript of the discussion for each problem description and to mark those utterances where a speaker is disagreeing with another speaker's classification or understanding of the problem, and those utterances where the speaker is indicating a change of view regarding the problem, or classification of the problem. Following this the judges had to indicate their agreement or disagreement with three statements regarding the results of the design team's discussion of each problem.

7. 4 Results

The results of the study will be structured and presented in accordance with the eleven research questions outlined in the introduction. The data that is described in this section can be seen in more detail in appendices 9, 10 & 11.

7. 4. 1 Does ECM fulfill the criterion of *usable*, set out in chapter 5?

The *usability* requirement set out in chapter 5 was that the classification scheme should a) fit into current design practice, and b) be simple and easy to employ. First we will consider the question of whether ECM might fit into the process of design. It was suggested in chapter 5 that a scheme that changed the designer's role and placed him or her in a less valued position (i.e. to simply implement that changes the Human Factors Engineer proposes) is likely to be rejected by designers. For this reason ECM was aimed at the designers themselves, as well as individuals closely related to the design team. The purpose of the scheme was to help designers structure the way they think about users' problems.

The subjective impression the author gained was that the scheme managed to achieve this aim, of being a useful and complementary design tool. This impression is supported to some extent by the replies to question 5 in the interview. Subjects were asked, "Are you likely to use this scheme in future discussions about user errors?" Software engineers 1 & 2 replied that it was "quite likely" that they would use the scheme in future discussions, while the two human factors engineers, the technical writer, and the senior learning products engineer all replied that they would "possibly" use it (software engineer 3 was not available for interview). All of those subjects who replied that they might "possibly" use the scheme also stated that it depended upon how the technique might fit into the company's overall human factors strategy, while human factors engineer 1, who was in the process of developing a course on human factors, stated that the scheme would definitely be used in some way, even if it was not explicit and only assumed within their other techniques and methods. Overall these responses were viewed as being positive towards the use of the classification scheme.

Although it appears as though the scheme should fit into the process of design, there is still the question of usability, although the subjects' responses to question 5, just reported, provides some indications as to this. In the introduction to this chapter it was suggested that the three principal measures of whether ECM is *usable* are; whether the designers *can* classify problems using the scheme, whether designers *do* classify problems using ECM, and the designers opinions as to the overall usability of the scheme. Furthermore, there is the question of whether the classification of problems was *natural* and *obvious* to the designers, and the measure of this was the extent to which the designers agreed upon each classification before the discussion. However, the agreement between designers on their classifications, as well as the designers' opinions of whether the scheme was usable, will be considered in later subsections. Here we will examine whether designers *could* use the scheme and whether they *did* use the scheme.

All of the subjects, except for two, were able to classify all of the twenty problems from the classification questionnaire. The two subjects who were not able to classify all of the problems were the technical writer, who could not classify problem 12, and human factors engineer 2, who could not classify problem 10. In addition to this human factors engineer 2 only classified up to problem 14, and claimed that he did not have enough time to spare to finish the questionnaire. Consequently, the lack of any classification from this subject for problems 15 to 20 was not viewed as a failure of the scheme. If we exclude these six problems (15 to 20) for human factors engineer 2, then there was a total of 134 possible classifications (7 subjects x 20 problems - 6 problems for human factors engineer 2). Of these 134 opportunities to classify, subjects classified in 132 cases (98.5%). This is taken as a demonstration that the ECM scheme was both possible to use and was used by the designers. However, whether the scheme might be used by the designers when not in a setting where the scheme was being triailed is not a question that can be answered here.

In summary, the classification scheme does appear to fit into the design process. It was *possible* for designers to classify the problems, and for the large majority of the cases the designers *did* classify the problems.

7. 4. 2 Do designers consider ECM to be *usable?*

Each subject was asked, "Do you believe that this classification scheme is generally usable?" Human Factors Engineer 1, Software Engineer 1, the technical writer and the learningproductsmanager all stated that the scheme was "quite usable", while Human Factors Engineer 2 and Software Engineer 2 stated that the scheme was "moderately usable". Software Engineer 3 was not available for interview. In general it appears as though the designers generally considered the scheme to be usable.

7. 4. 3 Do the designers agree on their classifications?

Afurther test of the usability of the scheme was whether the classification process was *natural* and *obvious* to the designers. The measure of this was the agreement between subjects/team members as to their classification of the problems that were presented. In figure 7.2 the number of team members who classified each problem in the four categories permissible is shown. There were seven team members and so if either 0 or 7 subjects classified a problem as being of that type then this indicates completeagreement. If 3 or 4 subjects classify a problem as being of a particular type then this indicates the most disagreement. Given that it was possible to classify each problem in more than one way if it was believed that it had more than one cause, a glance at figure 7.2 seems to indicate that there was considerable agreement for much of the time (see appendix 9 for more detail).

Problem Number	Operation-Concept	Operation-Symbol	Object-Concept	Object-Symbol
1	4	3	0	0
2	7	0	7	0
3	0	7	0	0
4	7	1	0	0
5	7	0	3	2
6	6	5	4	1
7	3	0	6	0
8	6	0	3	0
9	7	3	2	1
10	2	2	0	3
11	2	1	5	2
12	4	1	3	0
13	6	3	1	0
14	7	0	3	0
15	1	0	4	3
16	4	2	0	0
17	5	0	3	0
18	6	0	1	0
19	6	0	2	0
20	4	5	0	0

Figure 7.2: The table above shows the number of team members who classified each problem in the categories shown. As there were seven team members, and it was possible to classify each problem in more than one way if it was believed that it had more than one cause, then either a "0" or a "7" in any box indicates complete agreement. A "3" or a "4" indicates the most disagreement. The line after problem 14 indicates that it is after here that only 6 of the subjects had classified the problems, and so "6" and "0" indicates complete agreement.

If all of figures in figure 7.2 are considered, it can be seen that there was complete or near-complete (only 1 disagreeing) agreement in 53 (66.25%) of the 80 instances (or boxes) while there was some measure of disagreement in 27 (33.75%) of the 80 instances. When re-calculating this it should be noted that one subject did not classify problems 15 to 20. Consequently, for these problems, "6" and "0" are considered to be complete agreement while "5" and "1" are considered to be near-complete agreement.

Overall, these results appear to lend support to the idea that classifying the problems in most cases was *natural* and *obvious*, as there is complete or near-complete agreement for over 66% of the time. It is interesting to note that where there was not agreement then, during the meeting, this provoked long, stimulating and revealing discussions about what the user wanted and how the system fulfilled these needs.

7. 4. 4 Does ECM fulfill the criteria of *generalizable*, set out in chapter 5?

In chapter 5 it was stated that the classification scheme should reflect fundamental classes of errors rather than any one system structure in particular. That is to say, that if the scheme is to be widely used then it needs to be generalizable; it should be possible to use the scheme on any system. Consequently, as the scheme had already been used on the Hewlett Packard Memomaker system (see chapter 6), if the scheme could be used on the Newwave system then this is an indication of whether the scheme can be generalized from one system to another.

As stated in an earlier subsection (7. 4. 1), of all of the 134 opportunities to classify, subjects classified in 132 cases (98.5%). This is taken as a demonstration that the ECM scheme was generalizable, at least to the Newwave system. Nevertheless, the third and final study of ECM should serve to further confirm the generalizability of the scheme.

7. 4. 5 Do designers consider ECM to be *generalizable?*

Although the best test of the generalizability of the ECM scheme was seen as whether the designers *could* classify problems with a system other than that first used, the designers' opinions of whether the scheme was generalizable to other systems was considered to be of interest. The subjects (designers) were asked, "Do you think that the classificatory scheme could be used to classify user errors on other systems?" In response Human Factors Engineer 1 and the Learning Products Engineer replied "most certainly", Human Factors Engineer 2, Software Engineer 1 and the Technical Writer replied "quite likely", and Software Engineer 2 replied "possibly". These responses appear to suggest that, overall, the design team considered the scheme to be potentially applicable to other systems. However, several of the subjects qualified their answers by stating that they could not be sure until they attempted to apply the scheme.

7. 4. 6 Do designers consider ECM to be *comprehensive?*

The subjects (designers) were asked, "Was it possible to classify all errors under the scheme?" All six subjects that were available for interview replied "almost all". Their responses, however, may not be a reflection on whether the scheme was comprehensive, as it was possible to classify all of the twenty problems. The difficulty that the design team experienced, that their answers to this question appear to reflect, is that some problems could be classified in a number of conflicting ways. That is to say, that for some problems, several subjects had opted for one classification, while several others had opted for another classification. While these differences could often be quickly resolved, some problems proved more stubborn. Upon discussion it was sometimes discovered that, although there might only be one cause of a problem, it seemed legitimate to classify this cause in more than one way.

This is clearly a weakness of the scheme. A criteria that appears to be implicit within any classification scheme, but that was not explicitly considered in chapter 5, is that the categories of the scheme should be *exclusive*. In other words, if something is classified as one thing under the scheme, it should not be possible to classify the same entity as something else as well.

7. 4. 7 According to designers, does ECM provide *specific* information?

It was suggested in chapter 5 that the classification scheme must not only classify error types, but also show where errors arise. In other words, the process of using the scheme should help the designer to gain an appreciation of which parts of the design requireattention.

Subjects were asked, "Does the classificatory scheme adequately suggest ***which parts*** of the design need to be changed?" Software Engineer 1 and the Learning Products Engineer replied that the scheme was "very helpful" in this respect, while the remaining four subjects that were available for interview all replied that the scheme was "quite helpful". These responses appear to suggest that the scheme is *specific* according to the designer's opinions, in that it helps them to identify those parts of the design that require alteration.

7. 4. 8 According to designers, does ECM provide *detailed* information?

The *detailed* criterion proposed in chapter 5 suggested that the classification scheme must provide information that is detailed enough to allow the designers to pinpoint what is wrong with a particular part of the system, as well as how it might be changed. Subjects were asked, "Does the classificatory scheme provide enough detail for you to suggest design changes?" Later they were asked, "Does the classificatory scheme help you decide ***how*** different parts of the design might be changed?" All of the six subjects that were available for interview responded in the same way to both questions. All stated that this was not a proper function for a scheme like this to fulfill, and that solutions to problems require creativity not formalization. It appears as though this criterion, that the scheme should provide information that was detailed enough to suggest design changes, may have been inappropriate.

7. 4. 9 Does the process of discussing ECM classifications encourage the design team to explore the user's problems in detail?

As has been mentioned in section 7. 3, the design team's discussion lasted 2 hours, which may be taken as one indicator of the extent to which the subjects (designers) considered the user and the classification of the user's problems. In total there were 360 statements made during the discussion (see appendix 7 for a transcript of the discussion). A statement, in this for the purpose of this study, is considered to be any utterance, no matter how long or small, before another speaker makes an utterance.

For each problem there was a mean average of 18 statements (SD = 14.3). However, as the standard deviation figure indicates, there were considerable differences between problems as to the number of statements that were made. For example, for the discussions of problems 3 and 19 there were only 5 statements made for each, whereas there were 68 statements made during the discussion of problem 6.

The mean number of statements for each problem may have been artificially lowered, however, by the discussions of problems 16 to 20. It appeared that the design team were becoming tired and wished to end the meeting. Consequently, their discussions of these problems tended to be short, even when they disagreed about the problem or the classification of the problem.

From the number of statements, and the time the meeting took, it appears as though the discussion of the classification scheme does encourage a detailed exploration of the user's problems with the system where designers disagree about the classification of a problem. Where designers agree upon the classification, or were tired and looking forward to the end of the meeting, there discussions were much shortened and could not be described as detailed.

7.4.10 Does the process of discussing ECM classifications cause various members of the design team to alter their views of the problem?

The answer to this research question was intended to be provided by the independent judges' interpretations of the transcript of the discussion. The judges were asked to mark in blue pen any utterance where the speaker appears to be indicating that they are changing their view of the problem, and to mark in red pen any utterance where the speaker was disagreeing with another speaker. The full instructions given to the judges, together with the questionnaire they were required to fill in can be seen in appendix 8, while the transcript of the discussion can be seen in appendix 7. The criterion that was set for the judges interpretations was that at least 3 of the 5 judges must agree upon the interpretation of a statement for their answers to be considered.

Unfortunately, there was no agreement between the judges (except for 3 statements of the 360). This lack of agreement can be explained if we consider the nature of a technical discussion. All of the participants to the discussion had intimate knowledge of the system that was being considered. This *mutual knowledge* (Habermas, 1981), however, was not shared by the judges. Furthermore, much of the communication, in terms of indicating agreement etc., was non-verbal. For example, the author, who was present at the discussion, frequently witnessed subjects nodding to the speaker to indicate that they had changed their point of view and accepted the speakers interpretation of the problem and classification. Consequently, it was difficult for the judges to interpret the discussion statement by statement, although some stated that the gist of the discussion was easier to understand.

Although the research question that formed the title to this subsection cannot be directly answered, the judges' answers to the questionnaire they were given does provide some indications as to what a reasonable answer might look like. For each problem the 5 judges were asked to indicate their agreement or disagreement with each of the following statements; "The engineers managed to agree upon a classification", and, "The engineers managed to agree a common understanding of the problem". The judges had the further option of marking that they were *not sure*.

The criterion for considering the judges responses was the same as before, that is to say, that 3 or more of the 5 judges had to agree (had to provide the same response).

Using this criterion, it was judged that the engineers (subjects or designers) managed to agree upon a classification for 7 of the problems (problems 1, 2, 3, 5, 7, 14, 18). They did not agree upon a classification for 4 of the problems (problems 6, 13, 16, 19). For the remaining 7 problems the judges could not agree (problems 4, 8, 9, 10, 11, 12, 15, 17, 20). For the second question it was judged that the engineers managed to agree a common understanding of the problem for 12 of the problems (problems 1, 2, 3, 4, 5, 7, 10, 11, 14, 18,. 19, 20). The engineers did not agree a common understanding of the problem for just 1 problem (problem 13). For the remaining 7 problems the judges could not agree a common interpretation (problems 6, 8, 9, 12, 15, 16, 17).

For those problems where the judges could agree an interpretation of the discussion, the majority appear favourable. That is to say, that for 7 of the problems the subjects (designers or engineers) managed to agree a classification and for 12 of the problems subjects managed to agree a common understanding of the problem. Consequently, although it is not possible to objectively state whether subjects altered their views during the discussion, it does appear that a degree of consensus was reached during the discussion about the classification. Furthermore, and possibly more importantly, consensus was reached amongst the designers as to the understanding andinterpretation of the user's problems.

7. 4. 11 Does the process of discussing ECM classifications help the design team to arrive at solutions to the user's problems?

The answer to this problem, according to the designers was *no*. They argued that solutions to problems required creativity, and that this was not an appropriate role for a classification scheme. The judges' interpretations of the discussion appear to support this assertion on the part of the subjects. Using the same criterion of agreement between judges explained in the last subsection, the judges decided that the designers agreed a solution to a problem in 2 cases (problems 11, 20). They did not agree a solution for 12 of the problems (problems 1, 3, 5, 6, 7, 8, 9, 12, 13, 15, 16, 19). For the remaining 6 problems the judges could not agree (problems 2, 4, 10, 14, 17, 18).

It appears, from these results and the results reported earlier regarding the designers' (subjects) opinions, that not only does the use of the classification scheme not suggest solutions to the user's problems, but also that discussions of the user's problems, where the aim is to classify these problems, do not necessarily lead to solutions. Nevertheless, it is possible that solutions may have been found if the objectives of the discussion had been widened to include agreeing solutions. However, this is not a question that can be answered here.

7. 5 Discussion

The high percentage of problems classified (98.5%), together with the extent to which designers (subjects) agreed upon classifications, would appear to suggest that evaluative classifications of this sort can be used within the design and development process. The most interesting question that this appears to raise, however, is why such an evaluative classification might work.

One answer might be that the scheme encouraged designers to talk about the problem. Although, ironically, the most discussion occurred for those user-system errors where the design team disagreed upon classification. It did appear, even when some individuals could not agree upon the classification of a dialogue failure, that the process of considering their differences, nevertheless, helped them to clarify the user's problem in their own minds, and agree upon a solution to this problem. Indeed, this seemed to be a prominent aspect of the meeting that took place; that the scheme itself did not directly suggest solutions, but that it allowed designers to appreciate the problem better, out of which came suggestions as to how the system might be changed. But why might the classification scheme help designers to appreciate the problem better?

The answer does not appear to simply lie in the fact that designers were encouraged to discuss each problem, as some problems where designers agreed upon both the classification and the changes to the system were not greatly discussed. Moreover, these changes might not have been those that would have previously been considered. Consequently, the contribution the classification scheme made towards understanding the user's problems must have been greater than just encouraging discussion.

One strong possibility would seem to be that the scheme helped designers to think about the problem in a structured fashion, but in a way that was, to some extent, divorced from the structure of the system. It may be that, when considering changes to a system, designers normally tend to think of the user's problems from a system perspective, using the system's structure as a guide to analyzing these problems. ECM may have allowed the designers to step outside of this structure and view the user's problems from a perspective, where *right* and *wrong* were defined totally in terms of what the user wanted and expected. For the designer to claim (as all of us who have ever written a system do) that the user, "..shouldn't have done that..", or, "..attempted to perform an incorrect operation..", is not permissible within the ECM scheme; evaluation is from the user's perspective, and a system is measured against this perspective. In essence, the evaluative classification scheme in question may have helped designers to appreciate the user's problems better by providing a structure for their problems where the user's views and opinions were paramount.

Although the scheme appeared to have some definite beneficial effects, there were less positive aspects that emerged during the study. Not least of which is the problem regarding multiple classification of single elements. Any dialogue failure may have either one of more elements. An element is a single mismatch between either the concept or symbol of an object or operation within the system and the concept or symbol of an object or operation held by the user. A single element may cause a dialogue failure. Alternatively, a dialogue failure might be caused by a number of mismatched elements. Each dialogue failure need to be classified. The number of ways in which it can be classified should directly relate to the number of mismatched elements.

However, it became apparent, during the course of the study, that the designers were sometimes classifying problems with two mismatched elements in five different ways. By approaching the user's problems from slightly different perspectives each designer could justify his or her classification. Although some of these differences were resolved during the meeting as one designer might concede that the view that he or she had taken was not appropriate, this was not the case for all of the problems under consideration. One possible explanation is that the classification scheme was not properly explained. However, the discussions of the classifications revealed that designers had a very good understanding of the scheme. A second, and it appears more realistic explanation for this result, is that the scheme, although *theoretically exclusive* in its classification, is not *practically exclusive*. In other words, it is possible, in some circumstances, to classify one problem in a number of different ways. This is clearly a weakness in the ECM scheme. However, this need not be considered as a weakness in evaluative classifications of mismatch generally.

A further problem that appeared to emerge during the study was that mismatches could be treated in a piecemeal fashion under the ECM scheme. There appears to be a quite understandable desire amongst designers to rush off to fix a problem as soon as it has been identified. The problem that this creates is that a number of problems may have a single structural underlying cause, and that addressing each symptom in a piecemeal fashion cannot solve this problem. Yet ECM has no scheme or framework for linking mismatches together to form a coherent whole, or identify commonalities between different but related problems.

These two problems, of multiple classifications of single elements, and of a lack of an overall analytical framework, might be considered as criticisms of ECM as it presently stands. However, these need not be problems with future evaluative classifications of user-system errors. Indeed, overall the results have suggested that evaluative classification schemes may have a useful role to play within the design and development process, and that using evaluative classification schemes might be on route towards a better understanding of the user.

7. 6 Conclusion

Overall this study has brought to light a number of interesting results, several of which were unexpected. Using the designers' classifications of the twenty problems, and their opinions, as solicited during the interview after the discussion of the classifications, it appears as though the scheme fulfills the two major criteria that were the central aim of this study. In other words, the results of this study suggest that the ECM classification scheme is both usable and generalizable from one system to another.

Furthermore,there appeared to be relatively good classification agreement between the designers. It was suggested earlier that such an agreement might indicate that the scheme was *natural* and *obvious* to the designers. To a limited extent this idea seems to have been supported. However, a few problems caused great difficulty for the design team. It appears as though the ECM scheme is not exclusive in its classifications. That is to say, that it is possible to classify one element (cause) of a problem in more than one way. Moreover, the problems of thinking through these difficulties was likened, by one designer, to *mental gymnastics.* This comment appeared to reflect what several of the design team felt; that thinking through this small number of very complex problems was not always helpful, even if the scheme was useful in the majority of cases.

According to the designers' opinions, the ECM scheme is comprehensive, and helped to expose specific points in the design where change and improvement were needed. However, the idea that the scheme might show *how* the design might be changed was unanimously rejected, as designers argued that solutions to problems requiredcreativity, not formalization. This idea was further supported by the judges interpretationofthetranscriptof the discussion. Solutions to the user's problems were only agreed in 2 cases according to the judges, although this might have been different if the design team had been asked to agree upon solutions as well as classifications.

The discussion of the user's problems appeared to be most detailed where the designers had not classified the problem in the same way. In other words, disagreements over the interpretation and classification of the problem appeared to prompt long discussions, as we might have expected. However, when the subjects (designers) became tired they appeared less willing to enter into long discussions of the user's problems even when they disagreed upon the classification and interpretation of the problem.

From the number of statements, and the time the meeting took, it appears as though the discussion of the classification scheme does encourage a detailed exploration of the user's problems with the system where designers disagree about the classification of a problem. Where designers agree upon the classification, or were tired and looking forward to the end of the meeting, there discussions were much shortened and could not be described as detailed. For much of the time it appeared as though consensus, between the subjects, was reached during the discussion regarding the classification of the problem and the interpretation of the problem.

Overall, the results of the study have demonstrated that the scheme is *usable*, *generalizable*, *comprehensive* and shows up *specific* points in the design where change and improvement are needed. The discussion of the classification of problems appears to beneficial in that it allowed the design team to agree a common understanding of each problem and to discuss in detail the nature of the user's problems with the system. Although the scheme also failed to fulfill the *detailed* criterion, because designers did not believe that this was an appropriate role for the scheme, these drawbacks do not appear to be so great as to cause the scheme to be of no use.

On the contrary, the results have suggested that the ECM scheme may be of particular use during iterative design. One of the software engineers commented that he would have like to go back and question the users after using the classification scheme. The useful role of the scheme in creating interest and helping designers think through usability problems at the level of matching models (see chapter 1) appears to have been supported. However, this conclusion presupposes that the classification *accurately* classifies problems from the *user's perspective*, an issue that could not be addressed as part of this study. This question is the subject of the third and final study of the ECM classification scheme.

Chapter 8

An Investigation into the Usefulness of ECM

8. 1 Overview

The study reported in the last chapter demonstrated that evaluative classification schemes for user-system errors are likely to be of use in the design and development process. That is to say, that the ECM scheme was found to be both *generalizable* from one system to another and *usable* by designers. The purpose of the final study reported in this chapter was to discover whether ECM *accurately* characterizes the user's misunderstandings. In other words, does ECM successfully classify from the *user's perspective?*

To answer this question a small system was developed. This system was concerned with train timetables and ticket prices. Two users performed a series of tasks on the system, and their errors and comments were used to inform four designers. Two designers used ECM to analyse the problems and the other two designers did not. Each designer produced a list of recommended changes. An independent judge combined these recommendations to produce one list of changes from the designers who used ECM and another set of changes that were produced from the recommendations of designers who did not use ECM. These changes were implemented to produce two further systems. These three systems (the original, the non-ECM system and the ECM system) were then tested in an experiment where the major measures of system improvement were time to perform tasks, errors and user attitude scores.

The results indicated that the two modified systems were better, in terms of errors, time and attitude scores, than the original system, as we might expect. Furthermore, the system produced using ECM was shown to be better, in terms of errors, time and attitude scores, than the system that was produced where ECM had not been used. These results provide support for the idea that evaluative classifications does accurately classify user-system mismatches from the user's perspective. Furthermore, the results were viewed as an indication that ECM might be used during the evaluation of system to enhance the usability of systems.

8. 2 Introduction

In the previous chapter ECM was considered against several of the criteria described in chapter 5. It was found that ECM was *usable* and *generalizable,* according to the definitions of these criteria given in chapter 5. Moreover, the notion that ECM was *comprehensive* and helped to identify *specific* points in the design that required alteration was confirmed. However, it was discovered that ECM was not exclusive in its classification. Furthermore, designers stated that a technique and classification such as ECM need not be *detailed* in suggesting design changes. Indeed, they argued that changing the design of a system was a creative exercise and not an appropriate role for a classification technique.

Nevertheless, despite the failure of ECM in these last two areas, the overall success of the scheme, in helping designers to consider the user's problems, suggested that the classification scheme was worthwhile from a practical standpoint. The purpose of this final investigation was to discover whether ECM fulfills the remaining criterion, suggested in chapter 5; whether it classifies problems from the *user's perspective.* While it is possible to imagine that we always attempt to solve usability problems by looking from the user's perspective, the question we might ask is; have we succeeded in representing the user? In other words, the criterion relating to classifying from the *user's perspective* might be restated to ask; does classification using ECM *accurately* reflect the user's understanding of the system, or does it reflect the classifier's (designer's) misunderstanding of the user?

The major concern of the study reported in this chapter was to answer this question. If ECM accurately classifies problems accurately from the *user's perspective* then we might expect a system that has been modified according to the results of an evaluation where ECM was used to be better appreciated by users, as well as encouraging better user-system performance. If ECM does not accurately reflect the user's perspective then we might expect no improvement from the original system to the modified one.

There is, however, a difficulty with simply adopting this idea of improvement as a measure of whether ECM accurately characterizes the user's understanding of a system. The problem is this; if we test a system by asking users to use it and noting their problems, complaints and comments, then we might expect the next iteration of the system to be an improvement regardless of whether ECM accurately captures the user's misunderstandings of the system. In other words, if we test a system and modify it we should expect an improvement whether ECM is used or not.

Consequently, it was decided that two modified systems would be produced. The first was modified without the use of ECM, and acted as a control. The second was produced using ECM. If ECM classifies misunderstandings *accurately* from the *user's perspective* then we might expect the system produced using ECM to be better than the original. Furthermore, we might also expect the ECM version to be better than the modified version, where ECM was not used to analyse the results of the evaluation.

8. 2. 1 Research Questions

The principal questions this study was aimed to answer were as follows:

1 Will the version of the system produced using ECM enable users to perform the tasks faster that users of the system modified without the use of ECM?

2 Will users of the system produced using ECM make fewer errors than users of the system where ECM was not used?

3 Will user attitude ratings be more positive towards the system produced using ECM rather than the system produced without using ECM?

8. 3 Method

Before the method is described in detail it may be useful to consider an overview of the stages that comprised this study (see figure 8.1).

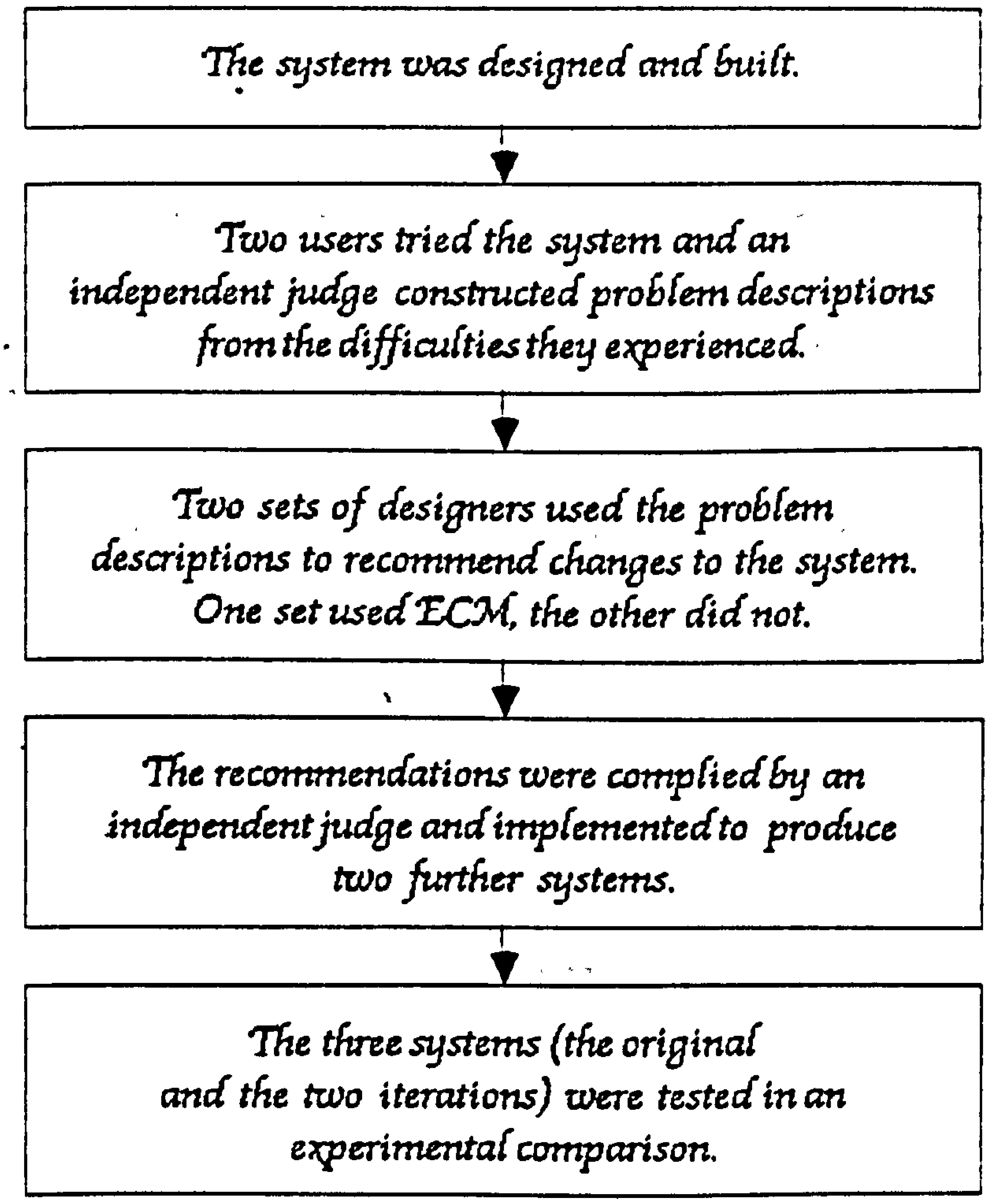


Figure 8.1: The five stages of the study.

In the first stage the system was designed and built by the author. For convenience the system was briefly titled Train Timetable and Prices System (TTPS). It was considered important that the system be kept as small as practically possible (both in terms of functionality and code). The small size of the system enabled a simple experimentalcomparison, without the added complexity of changing a large system. The system was quickly programmed in basic, and the listings of the original and the two iterations of the system can be seen in appendix 16.

In the second stage the system was tried by just two users. These users were given a set of instructions and tasks to perform (see appendix 12). This use of the system was supervised by an independent judge. This judge constructed problem descriptions from the difficulties the two users reported, and these were put into a list (see appendix 13).

In the third stage four "designers" familiarized themselves with the system, and with the instructions given to the users. Following this, they were given the problem descriptions and asked to use these descriptions to produce a list of recommended changes to the system. Two of the designers were asked to use ECM to analyse the problems before they produced their recommendations, while the other two designers did not use ECM.

In the fourth stage the recommendations of the designers were compiled by an independent judge to produce two lists: one list of changes recommended by the two designers who used ECM and another list of changes recommended by the designers who used only the problem descriptions (see appendix 15 for these two lists of design changes). These two lists of recommendations were then implemented by the author to produce two further systems: one system that was produced from the recommendations of the designers who used ECM, and one system produced from the recommendations of the designers who did not use ECM. At the time of programming these systems, in order to avoid a possible source of bias, the author was not aware of which recommendations were produced by which set of designers. This was known only by the independent judge.

In the fifth and final stage of the study the three systems (the original and the two iterations) were subject to an experimental comparison. Twenty four subjects (eight for each system) were run, and the major measures were the number of errors subjects made, the time subjects took to complete the tasks and attitude scores of the subjects towards the system they had just used.

8. 3. 1 Design

The independent variable in the experimental comparison was the system the users were asked to use; either the original, the iteration produced without using ECM, or the iteration produced using ECM.

The dependent variables were the errors subjects made, the time subjects took to complete the tasks, and the subjects' attitude scores on a questionnaire. This questionnaire was administered immediately after the subjects completed the experimental tasks on the system. In more detail:

1 Errors; these were defined in terms of *any incorrect entry into the system.* This

applied whether the entry was a keystroke error or not. For if the experimenter had to judge which were keystroke errors and which were not then this might have introduced a possible source of bias.

2 Time; this was the time from the subjects beginning the tasks, shown in appendix 12, to them finishing these tasks.

3 Attitude scores; these were the attitudes of the subjects to the system they had just used. These attitudes were measured using the questionnaire shown in appendix 14.

8. 3. 2 Subjects

The initial two subjects who provided the problem descriptions were both aged 23. One was female and worked as a nursery nurse, while the other was male and worked as a research officer for Kirklees council. Of the four "designers": the two who did not use ECM were both research assistants working in the HCI unit at Huddersfield Polytechnic. One was female, aged 32 years, while the other was male, aged 24 years. The remaining two "designers", who did use ECM, were both male, one aged 33 years and working as a Research Fellow at the HCI unit at Huddersfield Polytechnic, and the other was aged 24 years and held the rank of Captain in the Army Royal Medical Corps. The independent judge who observed these two subjects, and who later complied the lists of design change recommendations, was a female research student at Manchester University, aged 24.

Of the subjects using the original system all were male, aged between 19 and 23 years, and studying for either a HND or a degree in computer science at the Polytechnic. Of the subjects using the system changed with ECM seven were male aged between 19 and 23 years, and studying for either an HND or a degree in computer science at the Polytechnic. The one exception was a male research assistant aged 32 years, who was working in the HCI research unit at the Polytechnic. Of the subjects using the system changed using ECM seven were male aged between 19 and 23 years, and studying for either an HND or a degree in computer science at the Polytechnic. Again, the one exception was a male research assistant aged 30 years, who was working in the HCI research unit at the Polytechnic.

8. 3. 3 Materials

The experimental instructions can be seen in appendix 12. The instructions and problem descriptions that the four designers received can be seen in appendix 13, while the questionnaire that was used to assess the subjects' attitudes towards the system can be seen in appendix 14. This questionnaire was adapted from Poulson's (1987) questionnaire. The system was written in Basic on a BBC computer. The three program listings can be seen in appendix 16.

8. 3. 4 Procedure

Subjects were first asked for some biographical information, such as their age and occupation. Following this, subjects were asked to read the instructions shown in appendix 12. They were asked if they had any queries, and any questions about the system or the experiment were dealt with. Subjects were then reminded that they were being timed and that they "should try to be as quick and accurate as possible".

Once the subjects had completed the experimental tasks they were given the attitude questionnaire shown in appendix 14. When this questionnaire had been completed the subjects were debriefed.

8. 4 Results

The results of the study will be structured and presented in accordance with the three research questions outlined in the introduction.

8. 4. 1 Will the version of the system produced using ECM enable users to perform the tasks faster that users of the system modified without the use of ECM?

The times that the subjects from each group took can be seen in figure 8.2. If ECM does accurately classify user-system errors, and accurately identifies problems, then we might expect the system produced using ECM to enable users to perform their tasks faster than the system produced where ECM was not used. Furthermore, we might expect both the ECM and the non-ECM system to be faster than the original system.

Original System		non-ECM System		ECM System	
Subject Number	Time (in seconds)	Subject Number	Time (in seconds)	Subject Number	Time (in seconds)
1	630	2	608	3	280
4	753	5	618	6	260
7	790	8	582	9	305
10	445	11	370	12	272
13	702	14	640	15	314
16	611	17	495	18	377
19	852	20	552	21	378
22	715	23	603	24	291
Mean	687.250	Mean	558.500	Mean	309.625
SD	125.805	SD	88.457	SD	45.289

ANOVA TABLE - TIME

Source	Sum of Squares	Deg. of Freedom	Mean Squares	F-Ratio	Prob>F
Between System	589642.583	2	294821.292	34.412	0.0001
Error	179917.375	21	8567.494		
Total	769559.958	23			

Figure 8.2: The time taken by each subject to perform the experimental task, together with mean times and standard deviations. The ANOVA table is also shown.

An analysis of variance (see figure 8.2) shows that there is a highly significant difference between the three systems with respect to the time that user's took to complete the tasks they were set. Two planned comparisons were performed: one between the original system and the two re-designed systems, and one where the non-ECM and ECM systems were compared. The difference between the original system and the two re-designed systems was highly significant ($F = 39.91$, 1df, $p<0.0001$), demonstrating that the redesigned systems enabled faster user performance. The difference between the non-ECM and the ECM system was also found to be highly significant ($F = 28.92$, 1df, $p<0.0001$), demonstrating that the system re-designed as a result of the recommendations produced using ECM enabled faster user performance that the system where ECM was not used.

This result supports the idea that ECM *accurately* classifies user-system mismatches, as the system produced using ECM enabled users to perform their tasks in a shorter space of time than users of either the original system, or the non-ECM iteration of the original system.

8. 4. 2 Will users of the system produced using ECM make fewer errors than users of the system where ECM was not used?

The number of errors that each subject made can be seen in figure 8.3. If ECM accurately classifies user-system errors (or model mismatches) then we might expect the users of the ECM system to make fewer errors than both the original system and the non-ECM iteration of the original system.

An analysis of variance (see figure 8.3) shows that there is a highly significant difference between the three systems with respect to the errors that occurred during the experimental task. Two planned comparisons were performed: one between the original system and the two redesigned systems, and one where the non-ECM and ECM systems were compared. The difference between the original system and the two re-designed systems was highly significant ($F = 178.13$, 1df, $p<0.0001$), demonstrating thatthe re-designed systems enabled more error-free user performance. The difference between the non-ECM and the ECM system was also found to be highly significant ($F = 32.62$, 1df, $p<0.0001$), demonstrating that the system re-designed as a result of the recommendations produced using ECM enabled more error-free user performance that the system where ECM was not used.

This result further supports the idea that ECM *accurately* classifies user-system mismatches, as the system produced using ECM enabled users to perform their tasks with fewer errors than users of either the original system, or the non-ECM iteration of the original system.

Original System		non-ECM System		ECM System	
Subject Number	No. of Errors	Subject Number	No. of Errors	Subject Number	No. of Errors
1	6	2	4	3	1
4	7	5	3	6	0
7	8	8	4	9	0
10	8	11	2	12	0
13	7	14	3	15	0
16	6	17	2	18	2
19	9	20	3	21	1
22	6	23	4	24	0
Mean	7.125	Mean	3.125	Mean	0.500
SD	1.126	SD	0.835	SD	0.756

ANOVA TABLE - ERRORS

Source	Sum of Squares	Deg. of Freedom	Mean Squares	F-Ratio	Prob>F
Between System	178.083	2	89.042	105.35	0.0001
Error	17.750	21	0.845		
Total	195.833	23			

Figure 8.3: The errors made by each subject to perform the experimental task, together with mean numbers of errors and standard deviations. The ANOVA table is also shown.

8. 4. 3 Will user attitude ratings be more positive towards the system produced using ECM rather than the system produced without using ECM?

If ECM accurately classifies user-system errors then we might expect user attitudes towards the system produced using ECM to be more positive than those towards both the original system and the non-ECM iteration of the system. However, before the results are explained, it may be useful to briefly consider how we normally deal with attitude scores.

If we wish to refine a system then attitude scores can constitute a useful source of information. Each attitude score for a system can be treated separately in order that the strengths and weaknesses of the system can be exposed. That is to say, that the question of whether the system is easy to learn is treated separately to questions of whether the system performs all of the tasks the user requires. In other words, the information is being treated qualitatively in order than the design can be reformulated.

Original System		non-ECM System		ECM System	
Subject Number	Attitude Score	Subject Number	Attitude Score	Subject Number	Attitude Score
1	53	2	65	3	78
4	51	5	69	6	76
7	48	8	68	9	77
10	49	11	65	12	78
13	55	14	61	15	77
16	53	17	68	18	75
19	55	20	67	21	75
22	48	23	66	24	78
Mean	51.500	Mean	66.125	Mean	76.750
SD	2.928	SD	2.532	SD	1.282

ANOVA TABLE - ATTITUDE

Source	Sum of Squares	Deg. of Freedom	Mean Squares	F-Ratio	Prob>F
Between System	2571.583	2	1285.792	232.023	0.0001
Error	116.375	21	5.542		
Total	2687.958	23			

Figure 8.4: The attitude scores of each subject, together with mean attitude scores and standard deviations. The ANOVA table is also shown.

However, the objective here is not to improve any of the three systems that have been developed. The purpose is to show whether one system provokes better attitude ratings than another. Consequently, the scores for the questions from the attitude questionnaire (see appendix 14 for the questionnaire) have been added together to produce an overall score for each subject. A mark in the questionnaire which indicated a strongnegative attitude towards the system was scored as a "0", while an attitude that was strongly positive towards the system was scored as a "6". The scores in between these extremes reflect the varying strengths of the opinion expressed by the subject. The sums of these scores can be seen in figure 8.4.

An analysis of variance (see figure 8.4) shows that there is a highly significant difference between the three systems with respect to the attitude ratings that were provided by the subjects. Two planned comparisons were performed: one between the original system and the two redesigned systems, and one where the non-ECM and ECM systems were compared. The difference between the original system and the two re-designed systems was highly significant ($F = 358.9$, 1df, $p<0.0001$), demonstrating that the re-designed systems provoked far better attitude ratings than the original system. The difference between the non-ECM and the ECM system was also found to be highly significant ($F = 81.49$, 1df, $p<0.0001$), demonstrating that the system re-designed as a result of the recommendations produced using ECM provoked better attitude ratings than the system where ECM was not used.

This result further supports the idea that ECM *accurately* classifies user-system mismatches, as the system produced using ECM provoked better attitude ratings than either the original system, or the non-ECM iteration of the original system.

8. 5 Discussion

The most striking feature of the results is the extent of the differences between the systems concerned (the original, the non-ECM and the ECM system) All of the differences between the ECM and the other two systems were highly significant (all $p < 0.0001$). There is no suggestion that these results might be due to some statistical fluke, and the extent of the differences can be understood when the systems concerned are seen. The important question is why the systems are so different. One possible answer is that the differences observed might be due to the designers' abilities, as well as the extent to which they could adopt the user's perspective. One of the "designers" who used ECM had little computing experience, and it is possible that this individual could more easily understand the users' problems.

Another explanation is related to the imaginative skills of the "designers". It is commonly accepted that good design requires, amongst other things, a productive imagination, to invent novel and useful solutions to potential problems within the design. It is possible that the designers who used ECM happened to be better at producing useful ways of simplifying the user's task.

Overall, however, it is not believed that these two possibilities (differences in imaginativecapabilitiesandexperience) contributed greatly towards the differences that were observed between the systems in question. For instance, each set of recommendations were produced from the compilation of two "designers" recommendations, and the ECM designer with a non-computing background was paired with the research fellow with a strong computing background.

It might be argued that the diversity of this ECM pair helped to produce the better set of recommendations. In sum, the extent to which other factors affected the recommendations, that were used to produce the ECM and non-ECM iterations from the original system, can only be the subject of speculation. However, it is the belief of the author that, although these factors may have had some effect, they might also have easily worked in the opposite direction. Overall, it does appear that the evaluative classification scheme was a significant factor in improving the design of the system.

A further argument against the use of evaluative classification schemes might be that the extent of the differences between the ECM system and the other two systems might have been less if the system under consideration had been larger. The argument in support of this view is that; small changes to a small system can make a significant difference, whereas the same is not necessarily true of large systems. For instance, a majordifficulty in building large systems is that they are frequently inconsistent in the way they present tasks to the user.

Inconsistency, however, is less of a problem in small systems, where the whole system can be conceptualized to a detailed level by one individual. This might certainly be said to be true of the system used in this study. This system was particularly limited and straightforward, by comparison to even some word-processing systems. Yet the ECM scheme has no method for detecting inconsistencies throughout a system, it only evaluates from the user's perspective. The piecemeal approach of the ECM scheme might be particularly vulnerable to inconsistency problems within large systems, as mentioned in the last chapter. The idea being that responding to each mismatch might lead a design team to tackle a set of symptoms rather than the underlying cause of a problem. However, this need not be a problem for all evaluative classification schemes, although ECM clearly appears to possess this flaw.

The non-ECM Recommendations

2 There should be a short explanation of which buttons need to be pressed (including return) for the first menu.

4 The abbreviations for the stations should all be three letters long and should be displayed on the screen each time the user has to use them.

5 Each entry for the user should be checked for errors. The user should be allowed to retype just that entry which is incorrect, and should not be simply returned to the main menu.

8 The arrival times as well as the departure time should be shown.

9 Children's and pensioner's prices should be displayed as well as adult fares.

The ECM Recommendations

1 The first menu should allow the user the choose the stations (departure point and destination) he or she needs using just one key press. Furthermore, the user should not have to press return.

2 The first screen should explain which keys relate to which stations, and about what will happen once the keys are pressed.

3 All of the information should be displayed on one screen if possible, there should not be the different options that presently exist.

5 Each entry for the user should be checked for errors and the user should be allowed to retype just that entry which is incorrect.

9 The names of the stations should appear at the top of the screen when information is presented, and also above the departure and arrival times.

10 Children's and pensioner's prices should be displayed as well as adult fares.

Figure 8.5: The most significant recommendations made by the two sets of "designers". The full list can be seen in appendix 15.

In all, it appears as though the dramatic difference between systems that were demonstrated in this study might not be so apparent if the experiment was repeated with larger systems. The feasibility of such a replication is clearly in doubt, owing to the resources that might be required to develop such a system and its iterations. However, it is believed that the ECM scheme should still make a significant contribution towards the improvement of the iterations, even if this improvement might not be quite as marked as the differences between systems demonstrated in the experiment reported here.

If we consider the recommended changes shown in figure 8.5, then some clues seem to emerge as to why the system produced using ECM was better that the other two systems. Some of the recommendations, from the non-ECM and ECM "designers" are the same. For example, both recommended that the user's entries should be checked for errors and that the user should be able to retype entries, rather than start the whole process again. However, there are two key recommendations that appear to have been decisive in distinguishing the two re-designed systems that were produced. These are: recommendations 1 and 3 from the ECM "designers". Here, it was suggested that the stations should be selected using only one key-press, and that all of the information regarding the journey from the first station to the second should be displayed on one screen. Although this had the effect of making some of the later recommendations redundant,it nevertheless, dramatically increased the ease and speed with which users could get at the information they required when using the system produced using ECM.

Although the extent of the differences between the system produced using ECM and the system that was produced without ECM might be considered as an interesting result, this still leaves the question of why ECM has this effect. In other words, how does an evaluative classification scheme contribute towards better system design? The best answer appears to be the same as that suggested in the previous chapter. That is to say, that ECM helped the designers in question to think in a structured way that *did not* reflect the system's architecture. It might be that, even those with a HCI background, think of users' difficulties in terms of the computer system's architecture. Using an evaluative classification, however, may disrupt this way of thinking, and encourage designers to think about the user's problems from the user's perspective. In short, the use of an evaluative classification scheme may both enable and direct creative thinking when re-designing a system.

8. 5. 1 A Restatement of ECM

While it might have become apparent that ECM encourages creative thinking about a system, and facilitates design team discussions from original user-related perspectives, it must also be apparent that these positive effects, noted in this and in the previous chapter, cannot be entirely due to ECM. To a large extent, they arise, not out of the classification scheme itself, but out of the process of using the scheme. Indeed, this process, by necessity, is more extensive that the definition given in chapter five implies. Consequently, it might be helpful if the procedure of using ECM is outlined to provide some feel for the way in which its use has evolved throughout the research. (An overview of the stages that are about to be presented can be seen in figure 8.7.)

Selection of tasks: In an ideal situation we might want to consider every operation of the system in every possible situation. With a small system this might be feasible. However, with anything other than an almost trivial system such an objective is unrealistic. Consequently, there is a need to consider two types of task; those that are mostfrequentlyperformed and those that, although possibly infrequent, maybe critical in some respect. It is suggested that a *critical* task might be thought of as one that is crucial to the integrity and success of the whole, or a significant part, of a system.

Identifying tasks that might be critical depends upon a definition of criticality, and this, it is suggested, will vary from system to system. For example, a critical task in a chemical plant might be one where there is some potential safety hazard if the task is not performed correctly. In a banking system a critical task might be one where important data might be lost or funds incorrectly transferred/credited. Clearly, these tasks would need to be examined in any evaluation, however infrequent, given their consequences.

Frequently performed tasks might also be considered during evaluation. However, a pertinent question at this point is: performed by who? It might be wise to ensure that the system is tested on those who are likely to use it, and that tasks performed frequently by a secretary should be tested on a secretary rather than some other type of worker, given that different groups are likely to possess different knowledge of a task and are likely to approach it from differing perspectives. Consequently, it is helpful to present sets of tasks as stories, or as *task scenarios.*

> Task scenarios "..are descriptive stories about the intended use of a product. They provide a design vision, illustrate tasks that the system is intended to support, and provide the raw materials for usability assessments." (Booth & Marshall, 1989).

For each different type of worker there should be one or more task scenario. Within these scenarios there might be both the frequently performed tasks, which will involve all of the operations the user might normally be expected to employ, and critical tasks, which involve infrequently used but important operations.

Conduct and recording of usability trails: During evaluation there are a number of ways in which data might be collected. The method that is used will dictate, not only the type of data that is produced, but also the probable deficiencies in that data. For example, if concurrent verbal protocols are used then there is a possibility that this might, in some circumstances, interfere with the user's performance of the task. If a retrospective protocol is provided by users then there is a possibility that this data will not be as complete, and that users may have *rationalized* their behaviour; providing explanations for actions that had no part in the original decision to act (see Ericsson & Simon, 1980).

Bearing this in mind, the approach adopted within ECM is one where the user's opinions and explanations are explored to a large degree, at the risk of interfering with task performance. Users are provided with a task scenario. As they proceed through the tasks they have been set they might report when a problem has occurred. However, although subjects were always given the instruction that they should report any error, few subjects did this during the earlier part of the usability-testing session. This may have been because they were afraid of appearing foolish, despite the instructions that they received. Consequently, where errors or misunderstandings between users and the system occur, the person conducting the testing session should interrupt and ask the user for an account of what has happened. Empathy with the user is considered to be a central feature in this questioning—it helps users to assume that they are correct and the system is at fault, thus increasing their personal confidence and encouraging them to report fully on the problems they experience.

The problems that users experience should be noted down by the person conducting the testing session. In essence, this involves identifying a dialogue failure. To recapitulate: on the operational definition of a dialogue failure given in chapter 5:

A dialogue failure is considered to have occurred if:

- the user reports any degree of misunderstanding during the dialogue (i.e. the system does not do what he or she wanted it to do),
- the user asks for help in any form,

- the user enters an illegal command that is not purely the result of a keystroke error, mental slip or lapse.

As was stated in chapter 5: users must be informed at the beginning of the evaluative session that they should report any misunderstanding between themselves and the system. It is suggested that during evaluation users should be encouraged to *think out loud.*

The notes that are collected regarding these dialogue failures will later form the basis for the problem description that would be used to classify the error under the ECM scheme. However, given that there is the possibility that these notes might not accurately reflect any particular problem that occurred during the usability-testing session, it is also wise to video the session. Such a record has two potential uses. Firstly, it can be used to verify the notes made during the session. Secondly, it can be used to show errors/misunderstandings to design team members. One potential problem with the usability testing is that, although common errors can be exposed, these errors are not always fully believed by a design team who, of course, view the use of their system from a markedly different perspective to most other people. Consequently, a video recording can often be used to help the design team appreciate the problems users experienced with their system. Such "snapshot" recordings (only of the error concerned) can also be used together with the problem descriptions prior to classification under the ECM scheme. The video can thus help to provide the context for the error, and help those who were not present when the error occurred to appreciate the user's description of the problem, given that much of the communication between the user and the person who conducted the session is likely to be non-verbal.

Selection of common core errors: Essentially, selecting common core errors is a question of taking account of the number of users who experienced any particular problem. The threshold that is set may depend upon the number of users employed during the usability trials. For example, if 100 users were employed then we might want to only consider errors that occurred to 4 or more users. The pilot study of ECM demonstrated that there are likely to be many errors that are peculiar to each user. Consequently, a useful threshold for selecting common core errors is 2; where only errors that occurred to 2 or more users are considered. Such a threshold is likely to rule out a consideration of a large number of the errors that occurred (the so-called idiosyncratic errors.) Figure 8.6, for example, shows a histogram of the number of errors that were experienced by 1, 2, 3, etc. users during the pilot study of ECM, reported in chapter 6.

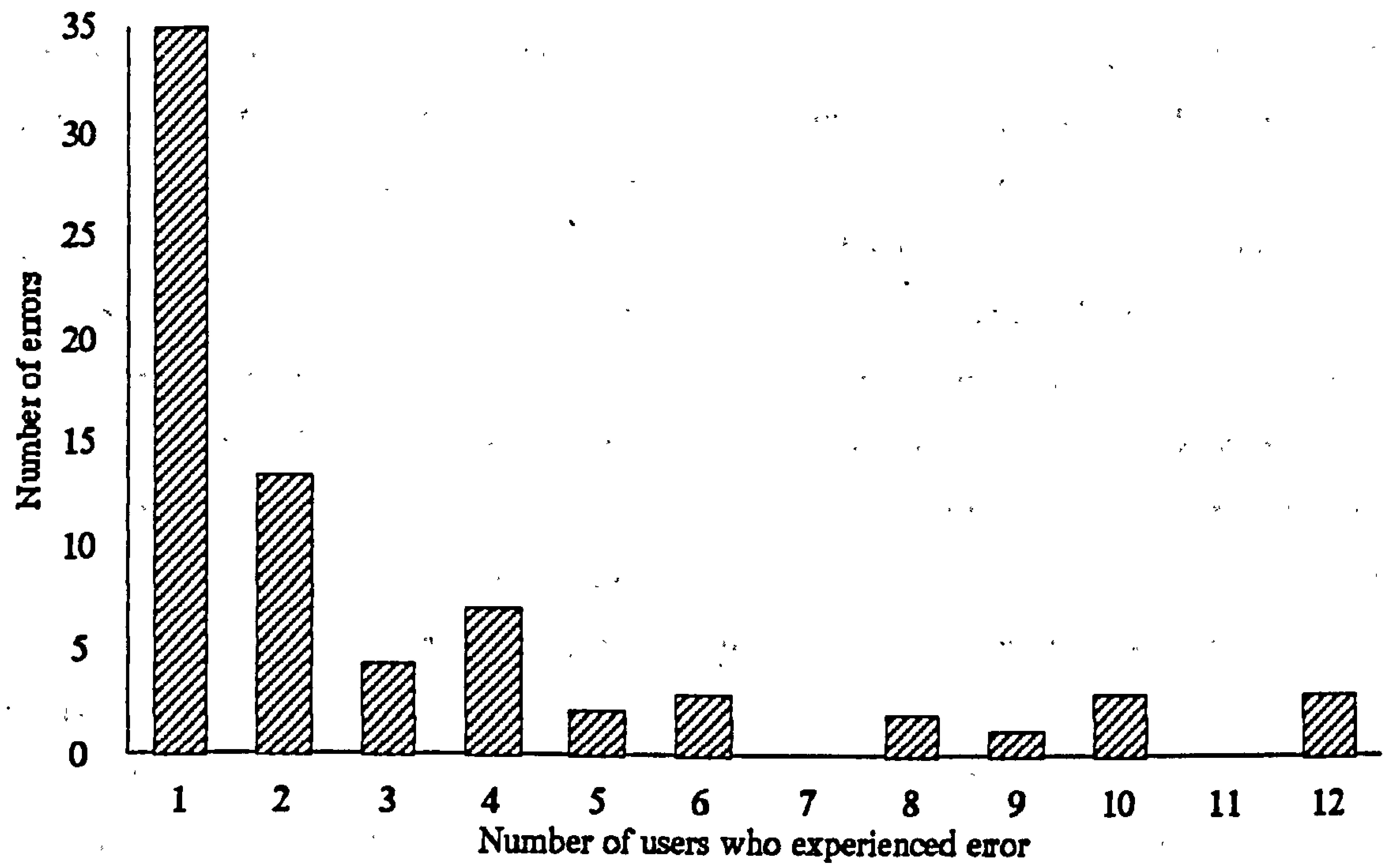


Figure 8.6: This shows the number of errors that were experienced by 1 user, by 2 users, by 3 users, etc., up to all 12 users during the pilot study of ECM, reported in chapter 6.

However, the errors that are addressed should not normally be limited to common core errors. Idiosyncratic errors (i.e. errors that fall below the threshold set) might also be considered where their consequences might be serious for either the user, the system or some third party. Consequently, each error might need to be screened with regard to its consequences before it is discarded. We might expect such screening to involve a consideration of the consequences, not only of the errors, but also of any series of possible operations that might normally follow these errors.

In summary, it is suggested that common core errors should be considered during analysis using ECM. Common core errors are those misunderstandings that occur to 2 or more users, although this threshold might be set at a higher number. In addition to the common core errors, it is suggested that those idiosyncratic errors that might have serious consequences for either the user, the system or some third party should also be the subject of analysis.

Classification of errors: The classification scheme suggested is identical to that proposed in chapter 5 in the formal definition. Armed with the problem descriptions (outlines of the dialogue failures) the object or operation that is at fault from the user's perspective is identified. To recap.: An object is *a thing to which something is done or about which something acts or operates.* An operation *is an action which is performed upon an object or objects within the system.* A general rule of thumb for distinguishing objects from operations is that an operation is something that is done to an object whereas an object has operations performed upon it. There may of course be more than one object or operation, if a multiple mismatch led to the dialogue failure.

Once the object or operation that is at fault is identified then the next stage is to identify whether the misunderstanding was a concept or a symbol mismatch. Once again, to recap.: A concept *may be either an object or operation whether represented mentally (the user) or in lines of code (the computer).* A concept mismatch is *a fundamental difference in the understanding and representation of system or task objects or operations.* A symbol is *a word, character, sign, figure, shape or icon employed by either the user or the system to represent an object or operation within the system.* A symbol mismatch is not one of fundamental understanding, *but occurs where the computer and the user adopt different terms to represent the same concept.* A rule of thumb for distinguishing symbol from concept mismatches might be whether the users' symbols for the systems objects and operations easily map onto the designer's version of the task or system. If terms cannot be easily substituted then the problem may be of a more fundamental kind, where the designer's and user's conceptualizations of the concepts or structure of either the system or the task (their models of each) are inharmonious.

In the final part of the classification process the cause (elements) of the error is described with respect to their position within the dialogue failure. This simply means that its contribution to the dialogue failure is considered. I.e. was this the only mismatch/elementthatcontributed towards the error? Where there others? What role did each play in leading the user to act as he/she did? (A summary of the classification scheme is outlined in appendix 2.)

Discussion of error classifications: In theory, the central aim of holding a design team discussion is to agree classifications for the different dialogue failures. Once the classification, and hence the nature of the problem, has been agreed then the system can be altered to ensure that such dialogue failures do not reoccur. This scenario, however, is far removed from the reality of what is likely to occur. The central benefit of a discussion is unlikely to be the classification consensus that might arise (although this may be helpful), but the exploration by members of the design team of: a) each others understanding of the problems users encountered, and b) each others understanding of the system. The discussion at Hewlett Packard, reported in chapter 7 (a transcription can be seen in appendix 7), suggested that discussing the user's problems in an attempt to fit them into a user-orientated classification scheme, helped the design team members to appreciate their system from the user's perspective. Moreover, as some details of the system were discussed it became apparent that, implicit with the discussion, there were differences of opinion as to what the system should be doing and how it should do it. The discussion helped to make explicit these differences. In this respect, and in the context of the Hewlett Packard design team, the discussion appeared to be both lively, constructive and fruitful.

Consequently, it is arguable that the most valuable pieces of information that the design team members took away from that meeting, were not the ECM classifications, but the improved understanding they gained of: a) each other's perspective on the system; b) the intended functioning of the system; c) the user's understanding of the system. Indeed, even where a classification cannot be agreed, the discussion is still likely to be of considerable benefit to the participants, and to the design of the system as a whole.

ECM output and re-design: The formal output of ECM is the classifications of the errors that are agreed. The informal product of the process of using ECM might be considered as the greater understanding, on the part of the design team, as has just been outlined above. First, we will deal with the formal output.

ECM classifications, as mentioned in chapter 5, directly identify system components, together with the nature of the user's misunderstandings of them. Consequently, an object or operation within the system is identified as being at fault from the user's perspective. As a result, it is possible to re-design a system with a clear set of system concepts and interface symbols that need to be changed in some respect. Specifying exactly what these changes should result in is not considered to be within the remit of any HCI methodology, as such changes require a degree of inventiveness and creativity that cannot be embodied within any technique. Thus, the quality of the changes to the system can only be affected by the use of ECM, ECM cannot directly determine the exact nature of the changes to the system, as this is dependent upon the imaginative abilities of the designer or designers.

Selection of Tasks

Need frequently performed tasks and critical tasks
These should be presented to usersin the form of task scenarios

↓

Conduct and Recording of Usability Trials

The user is encouraged to point out problems with the system. Empathy with the user is important. Dialogue failures are noted and written up in the form of problem descriptions. These problem descriptions can be verified and augmented with the use of videos of the errors that occurred.

↓

Selection of Common Core Errors

Common core errors are normally those that more than one user experience, although this threshold might be set at a higher level when large numbers of users are employed during the usability testing. Two types of error are considered during analysis: common core errors, and critical errors where the consequences of the error for the system, the user or for a third part are serious.

↓

Classification of Errors

Errors are classified according tothe ECM scheme. First an object or operation that is at fault, from the user's perspectiveis identified. Following this theproblem is then attributed to either the concept of the object or operation, or the symbol representing the object or operation. Finally, the contribution the mismatched element made towards the error is described.

↓

Discussion of Error Classification

The design team, having been provided with the problem descriptionsand asked to classify them alone, meets to agree upon the classifications of different problems. The central benefits of this exercise are that a classification is agreed, the team gain a greater understanding of one anothers perspectives on the system, the intended functioningof the system is often clarrified, and a greater understanding of the system from the user's perspective.

↓

ECM Output and Re-design

The output of ECM, the classificatons, pinpoint the symbols and the concepts within the system that are at odds with the user's understanding of the task and system. This helps to focus attention onto those points that require attention. The informal information (.e. the designer's greater understanding of the user's perspective on the system) should help to steer re-design towards solutions that are acceptable to the user.

Figure 8.7: An overview of the processes involved in using ECM.

Informally, the process of ECM produces other information, regarding the system and its use, that was mentioned in the previous subsection. This informal information (the greater understanding of different team member's perspectives on the system, the agreed functions and operations of the system, and the greater understanding of the system from the user's perspective) is expected to directly influence, but not determine, the quality of the changes to the system. Importantly, it is expected that changes that result from a greater understanding of *why* a user acted in a particular way, and *how* a user came to misunderstand part of the system, is more likely to result in changes that will suit the user than if the designer did not possess this knowledge. (Again, an overview of the stages of using ECM can be seen in figure 8.7.)

8. 6 Conclusion

The system produced using ECM has been shown to enable significantly better performance than both the original system, and the iteration of the original system produced without using ECM. Furthermore, the attitude ratings that the ECM system provoked were significantly better than subjects' attitude ratings for the original system and the non-ECM iteration of the original system.

Despite the reservations expressed in the discussion about the strength of the outcome of the investigation, the results appear to provide a strong indication that ECM does classify user-system errors *accurately*. That is to say, that mismatches between the user and the system appear to be classified from the *user's perspective*. In conclusion, ECM appears to have fulfilled the final criterion, set out in chapter 5, that it should classify user-system errors from the *user's perspective*. In other words, while the study described in the previous chapter provided indications that evaluative classifications may be usable within the design and development process, the study described in this chapter has provided strong indications that such schemes would be useful in design. That is to say, that ECM appears to have made a significant contribution towards improving the design of a small system, and as such, appears to encourage accurate characterization of the system's faults from the user's perspective.

Chapter 9

Conclusion and Future Directions

9.1 Overview

The purpose of this chapter is to make explicit the general theme that emerges throughout the thesis, and to highlight the relevant points. To this end, the work that has been reported in the preceding chapters will be partly reviewed. The arguments regarding the use of cognitive grammars and cognitive style tests to predict user behaviour will not be repeated at length, but they form a crucial component of the central theme of the thesis, and so they will be revisited.

The main conclusion of this thesis will then be considered; that the use of evaluative classifications to understand user behaviour in HCI is both theoretically and practically appealing. Following this, the question of why ECM works will be addressed. Finally, some of the problems with using ECM will be considered, and the future theoretical development of the scheme will be outlined.

9. 2 Introduction

Behaviour at a task, or at the human-computer interface, is not determined by any single over-riding factor, but by the subtle and potentially complex interaction of a number of factors. Not least of these factors is the context of the situation, and the knowledge the user possesses and brings to bear on the task. Consequently, human-computer interaction has certain dynamic qualities that make it difficult to predict using simple measures or sets of rules. This is the central theme that as emerged throughout the thesis, and that underpins the decision to concentrate upon past behaviour or, more specifically, user-system errors, rather than cognitive grammars or cognitive style.

9. 2. 1 Cognitive grammars

Unfortunately, although cognitive grammars have been the focus of much attention within the HCI literature, they have a number of theoretical and practical problems. These have already been considered in detail in chapter 3. However, to briefly recapitulate; the major theoretical problems relate to problems with the grain of analysis and assumptions regarding user consistency. The major practical problems are that present formalisms do not fit easily into the design and development process and address only one aspect of interaction each (e.g. task-action mapping), they are intimidating in their complexity, and they require a sound understanding of the task that is to be programmed (which frequently does not exist at the early stages of design). An important further point is that they do not involve the user, and so expose mismatches between the system's and the user's model of the task that *should* occur, rather than those that *do* occur. Consequently, such methods, it was suggested, might be best considered as research tools, or as fore-runners to more usable techniques that might be developed at some later date, rather than as techniques that can presently be used within the design and development process.

9. 2. 2 Cognitive style

Using cognitive style measures to determine individuals' cognitive processing dispositions, and hence predict various aspects of preference and behaviour, presents slightly different problems. Whereas cognitive grammars can be accused of assuming a formal consistency in behaviour, using cognitive style measures might be said to assume a certain simplicity in behaviour. In well controlled situations, such as those found within laboratories, behaviour does sometimes conform to the predictions that can be made from some of the cognitive style theories (Fowler *et al*, 1985; Witkin *et al*, 1979). In complex situations, however, behaviour appears to be less straightforward.

In the experiment, reported in chapter 4, the visualizer-verbalizer and conceptual tempo cognitive style dimensions were not found to significantly affect behaviour at the complexexperimental task. That is to say, that according to this experiment, cognitive style does not appear to be an over-riding factor that could be used to predict user behaviour, thus adding support to the notion that, in most complex situations, behaviour is unlikely to be determined by any one factor.

9.2.3 The use of *world knowledge* in predicting behaviour

This, however, does not necessarily mean that we cannot predict behaviour at all. Only that we are unlikely to be able to predict the detail of behaviour consistently using context-insensitive measures such as cognitive style tests. Although there is a mathematical argument for the belief that we cannot consistently predict the detail of behaviour (Jarvinen, in preparation), there is, nevertheless, an argument to support the statement that we can predict behaviour for some of the time. Consider behaviour at a bank for instance, we write a cheque and the cashier hands over money. Thus, we can predict a great deal of behaviour for much of the time when it is related to routine tasks. But to do this we use our knowledge of the tasks involved, and of previous similar situations. In essence, we use our knowledge of past events to predict future ones. Such knowledge, however, is not embodied within either cognitive grammars or cognitive style tests, and it appears doubtful whether such world knowledge could be usefully embedded within any test or grammar.

In essence, the argument is this: we cannot predict user behaviour in a consistent and reliable way using grammars or tests because we require world knowledge to make such predictions. Even when we have this world knowledge there is no guarantee that our predictions will be correct. Consider, for example, the way in which a designer might make predictions about user behaviour, where his or her use of world knowledge is skewed by a mechanistic view of the user's tasks. An alternative way to view this mismatch between the user and the designer is to say that they do not have *mutual knowledge* (Habermas, 1981) of the tasks in question. That is to say, that the designer and the user may share knowledge of how to cash a cheque at a bank, and so can predict each other's behaviour in this circumstance, but they do not have shared knowledge of the user's tasks.

9. 3 User-system errors

It appears as though a user's behaviour at a new system is determined according to the knowledge that is possessed and which portions of that knowledge are applied. To successfully predict the user we might need to know much more about the way in which human knowledge retrieval occurs. Which cues at an interface provoke what sort of retrieval pattern? Moreover, there is a danger, given that all of the evidence suggests that human knowledge retrieval is very complex (Johnson-Laird, 1988; Reason, 1989; McClelland, Rumelhart and the PDP research group, 1986a & b), that even if we fully understood this process we could still not accurately predict its outcome.

An interesting illustration of the way in which people sometimes use their knowledge was provided by a middle-aged gentleman who had been asked to attempt a number of tasks using a system that employed a desk-top metaphor. After some time on the system the man became both agitated and excited. His attention was directed mainly at the icon shown in figure 9.1. He repeatedly selected the icon using the mouse and, instead of opening it, would type in combinations of letters and numbers on the keyboard.Eventually, the man explained that it was obvious to him that this icon was something to do with the security in the system, and that he had been attempting to enter passwords that he thought might allow him access. The gentleman was particularly disappointed when it was explained to him that it was a mailbox icon and could be opened in the normal way.

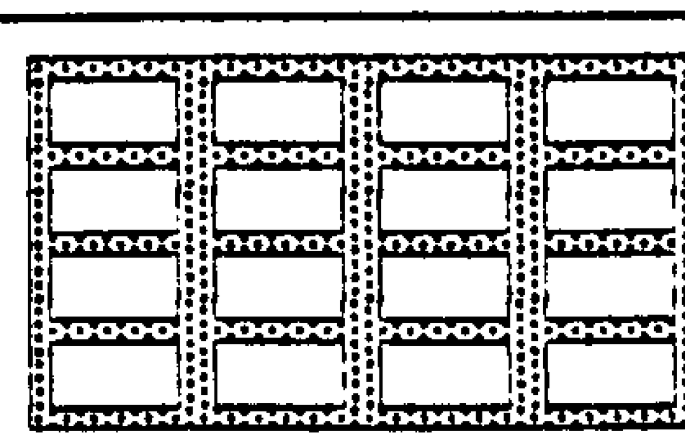


Figure 9.1: An icon from a system employing a desktop metaphor.

It is only when we stop to consider which elements of knowledge the gentleman was using that was begin to appreciate how he arrived at his conclusion. The mailbox icon is intended to represent a set of pigeon holes, but the man perceived these to be iron bars. The gentleman also stated that he had read about security problems in computer systems in the press and, seeing as he had been told that he could not break the system and was free to do as he saw fit, assumed that this icon represented some window to the security system that he ought to try.

From the user's explanation we can see that he used his knowledge about prisons/iron bars, and the scant knowledge of computers he had picked up from the press, to draw a quite sensible conclusion about an icon on the screen. It seems impossible to imagine that we might predict such a deduction on the part of a user. However, by examining the behaviour of users at systems, and questioning them, there is no reason to believe that we cannot gain some understanding of the areas of knowledge that users are likely to recruit when attempting to understand a system. Once we know which areas of knowledge are likely to be applied in understanding a new system we can then make some predictions regarding user behaviour towards that system. More specifically, we can pinpoint those areas where confusions are likely to occur and model mismatches develop.

9. 3. 1 The appeal of using errors

Such an approach to understanding user behaviour is both theoretically and practically appealing. The theoretically attractive aspects of the approach are that it does not involve attempting to predict user behaviour in a context-insensitive manner without some understanding of the areas of knowledge the user is likely to apply, as is the case when applying either cognitive tests or grammars. Another way of viewing this is to say that the user-system errors that occur during interaction reveal the model mismatches that present cognitive grammars attempt to predict. Consequently, such an analysis could focus upon mismatches that actually occur between users and systems rather than mismatches that might or should be present. Such an approach has its origins in the psychological study of human error, where behaviour is studied so that the knowledge that individuals applied can be deduced and the confusions or break-down points can be identified.

The practically appealing aspects of the approach are twofold. Firstly, there is a current need within design for techniques that allow us to evaluate designs as they proceed throughout the design process, as it is recognized that the specification of systems will never be an exact art. Furthermore, users' requirements change as they become more aware of both what the intended system should look like and the opportunities the technology offers them. An approach such as this might fit easily into present design practice, where almost all systems are now iteratively developed, and many are presently tested with users. Secondly, usability laboratories are currently being employed on an increasing basis within industry, yet there are few systematic means by which the data that is produced by these laboratories can be meaningfully analyzed. A consideration of those points where user-system errors occur might help to direct attention and resources to those parts of a design that require attention.

9.4 Evaluative Classification Schemes

Analysis could be based upon error classifications, such as those employed within the human error field, and a set of criteria that any evaluative classification of user-system errors should meet was proposed in chapter 5. It was suggested that any classification scheme should be *usable* by designers, *generalizable* from one system to another, should be *comprehensive* in its classification, should provide information about *specific* points in the design that require attention, should provide *detailed* information that might be used to suggest design changes, and should classify from the *user's perspective.*

An evaluative classification scheme for analyzing user-system errors was developed and formally defined in chapter 5 (ECM: An Evaluative Classification of Mismatch). The overall purpose of the ECM scheme was to act as a vehicle for investigation. In other words, it was to be used as a focus for research in an attempt to both demonstrate and investigate the potential usability and usefulness of such schemes within HCI. The pilot study and two investigations that followed generally demonstrated that evaluative classification schemes were likely to be both usable and useful within the design and development process. However, in the course of these studies a number of interesting issues arose, including the question of *common core* errors.

9.4.1 The pilot study

During the pilot study the question of *common core* errors was addressed. In essence, the argument is this; if user-system errors are specific to particular individuals then there is little point in considering them. For if this is the case, then altering a system to accommodate one user might inconvenience and confuse another. The results of the pilot study showed that, although many errors were peculiar to individuals, there was a common core of errors that were experienced by all, despite wide variances in the age, background and experience of the subjects who took part in the study. In short, it appears likely that there will be user-system errors that are common to either all users or certain groups of users, and as such, user-system errors would appear to warrant attention.

9.4.2 An investigation into the usability of ECM

In the study into the usability of ECM (chapter 7) 98.5% of all problems were classified, and there were high levels of agreement between designers in terms of the classifications that they choose. However, certain problems with the ECM scheme emerged. Firstly, the scheme did not appear to be practically *exclusive*. That is to say, that in a few cases designers classified single mismatched elements in different ways and appeared to be able to justify their classifications. The possibility that the classification scheme was not properly explained was rejected as the discussions of the classifications revealed that designers had a very good understanding of the scheme. It appeared as though the scheme, although theoretically *exclusive* in its classification, was not practically *exclusive*. In other words, it was possible, in some circumstances, to classify one problem in a number of different ways. Although this is a failing of the ECM scheme, this need not be considered as a weakness in evaluative classifications of mismatch generally.

The same might be said of a further problem that emerged. It became apparent that mismatches, under the ECM scheme, could be treated in a piecemeal fashion. That is to say, that problems identified using ECM may be fixed, but that if a set of problems relate to a single structural underlying cause, then designers will be addressing symptoms rather than causes. Unfortunately, ECM presently has no scheme or framework for linking mismatches together to form a coherent whole. Clearly this is a problem that future research would need to address.

9. 4. 3 An investigation into the usefulness of ECM

During the investigation into the usefulness of ECM (chapter 8) some impressive improvements in system usability were produced using the scheme. Although the characteristics of the designers involved may have had some limited affect upon this improvement, it is, all the same, seen as an important demonstration of the potential worth of evaluative classification schemes, such as ECM.

Nevertheless, it seems unlikely that such dramatic results might be repeated with a larger system. In essence, small changes to a small system can make a large difference, whereas the same is not necessarily true of large systems. One reason for this is that a majordifficulty in building large systems is that they are frequently inconsistent in the way they present tasks to the user. Inconsistency, however, is less of a problem in small systems and this might certainly be said to be true of the system used in the final study. Unfortunately, as has been mentioned above, the ECM scheme has no method for detecting inconsistencies throughout a system, it only evaluates from the user's perspective. However, this need not be a problem for all evaluative classification schemes.

9. 5 Why does ECM work?

The answer to the question posed in the sub-title, that appears to arise out of both the usability and usefulness studies (chapters 7 & 8), is that ECM helped designers to think in a structured way that *did not* reflect the system's architecture. Although ECM encouraged discussion, and this may well have contributed towards a better understanding, the contribution the classification scheme made towards understanding the user's problems must have been greater than just this discussion.

It would seem that the scheme helped designers to think about the problem in a structured fashion, but in a way that was, to some extent, divorced from the structure of the system. Even those of us who consider ourselves to be sympathetic to the user's plight tend to think of the user's problems from a system perspective, using the system's structure as a guide to analyzing these problems. ECM appears to have allowed the designers to step outside of this structure and view the user's problems from a perspective where *right* and *wrong* were defined completely in terms of what the user wanted and expected. In essence, the evaluative classification scheme in question may have helped designers to appreciate the user's problems better by providing a structure for their problems where the user's views and opinions were paramount.

A further aspect of the way in which ECM works, that arose out of the study into the usability of ECM, was that the process of classifying problems and then meeting to classify the problems, both enabled and directed creative thinking. It provided an opportunity to think originally about how best to design the system. The importance of this should not be underestimated, given that much of a design team's time is taken up attempting to meet design goals that frequently have not been fully considered or developed, in their rush against time and budget.

Finally, ECM seems to aid the understanding of the user's problems by specifying the exact feature of the design that is incorrect from the user's perspective. In other words, and in the terminology of ECM, either an object or operation within the system is identified and either the concept or the symbol representing the concept are said to be mismatched with the user's model. This feature of ECM is probably one of its key elements, and it is similar, in this respect, to human error classifications. That is to say, that the error is broken down into its possible constituents. In the case of human error classifications, errors are broken down into lapses and slips, the first being an error in forming the intention and the second being an error in executing the intention.

In short then, ECM appears to work, in that it contributes to understanding the user and improving design, for four reasons:

1 It is user orientated
2 It provides a structure for thinking about the user's problems outside the system's architecture
3 It enables and directs creative thinking
4 It specifies the exact features of a system that are at odds with the user's model of the task and system

Nevertheless, this list does not seem to wholly account for the way in which ECM works, and so it may be useful to consider the levels at which it works in a little more detail. If we consider Norman's (1986) gulfs of interaction, discussed in chapter 2, we can see that ECM addresses problems at a number of the stages identified by Norman. It is difficult to say which stages in each of the bridges in figure 9.2 is addressed by ECM. Nevertheless, it is apparent that certain classifications within ECM indicate problems within the two bridges. For example, a concept mismatch may mean that the user will form an intention that is inappropriate when viewed from the designer's perspective. That is to say, that the mismatch will be exposed during execution (Norman's first bridge).

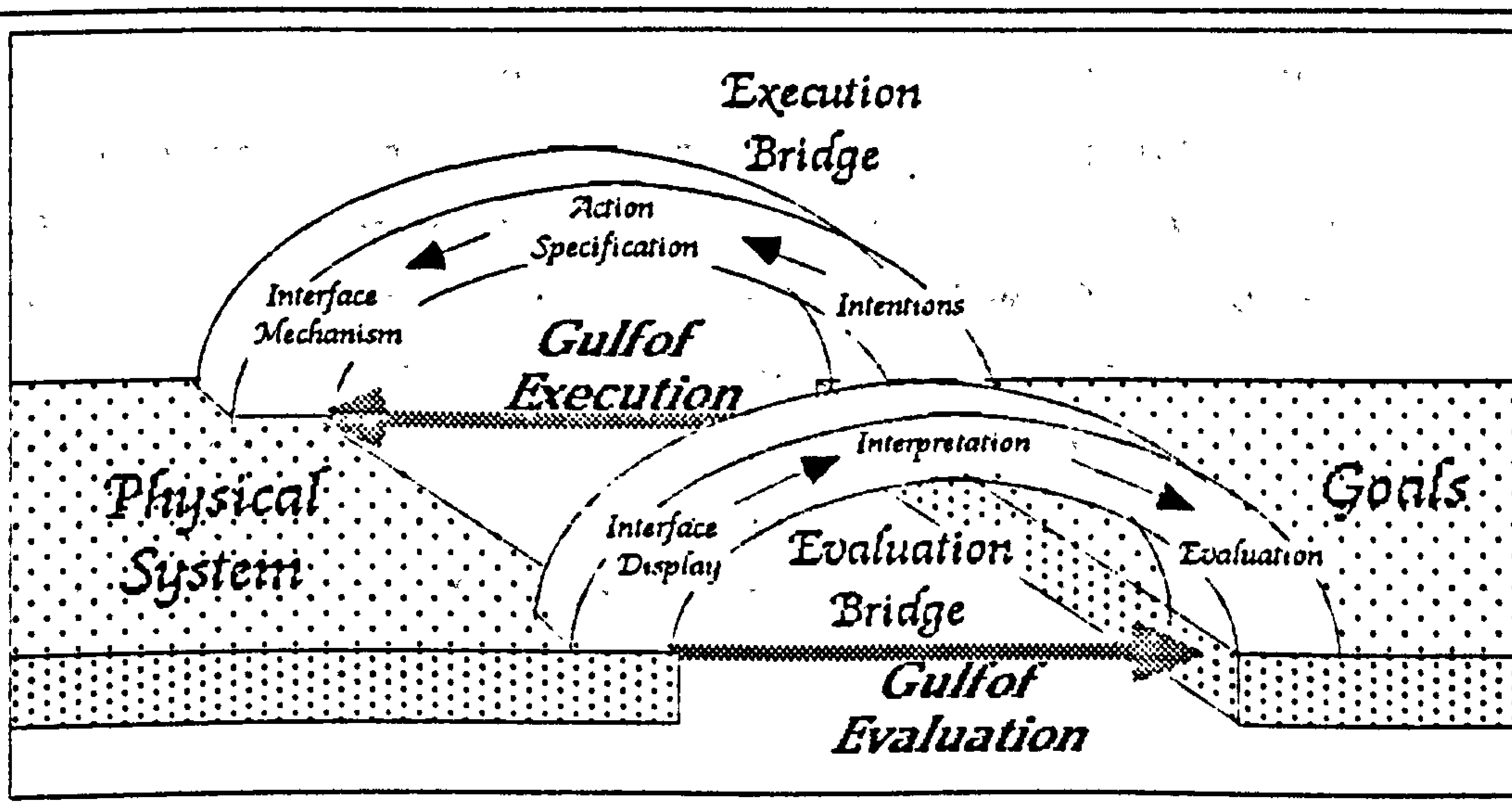


Figure 9.2: The gulfs of execution and evaluation, together with the bridges that span these gulfs. Adapted from Norman (1986).

A symbol mismatch will cause an error in evaluation (Norman's second bridge), although the problem may not show up until the user attempts to execute an inappropriate intention. For example, consider a problem from the pilot study of ECM.

TheMemomaker system showed text that was to be underlined when printed in italics on the screen. Consequently, although the concept was correct, the symbol representing the concept was not. This represents a problem of evaluation, although it does not usually show up until the user is observed trying to underline the text that has already beenunderlined.

It is more difficult, however, to specify exactly which stage during evaluation or execution where ECM identifies problems. However, this is probably because these stages are not easily distinguishable, as Norman (1986) acknowledges. An alternative way of considering ECM is to identify the levels at which it works within the framework laid down for Moran's (1981) Command Language Grammar (CLG; see figure 9.3).

Conceptualcomponent:	Tasklevel Semanticlevel
Communicationcomponent:	Syntacticlevel Interactionlevel
Physicalcomponent:	(Spatiallayoutlevel) (Devicelevel)

Figure 9.3: The framework for Moran's (1981) Command Language Grammar.

Once the object or operation that is at fault from the user's perspective, has been identified then the mismatch is classified as being due to an incorrect concept or symbol. It is the concept/symbol distinction that most readily maps onto the levels in Moran's (1981) CLG. A concept mismatch would appear to be a mismatch at the semantic level, whereas a symbol mismatch occurs at the syntactic level. This assumes that Moran's CLGis interpreted in such a way that the semantic level is considered to deal with the meanings and structure of objects and operations, while the syntactic level deals with the symbols that represent these objects or operations.

In truth, of course, ECM does not wholly and directly map onto either Norman's gulfs of evaluation and execution (1986; his approximate theory of action) or Moran's (1981) CLG. Indeed, if it did, then there would be little point in having such a scheme. In other words, if ECM mapped onto CLG precisely, then why not just use CLG? Yet, when trying to apply CLG in this way, it is often found that distinguishing between the semantic and syntactic levels is particularly difficult. This might be seen as evidence that, although similar, concepts and symbols are subtly different from semantics and syntactics, although this difference might emerge most pointedly when trying to classify actual user-system errors, rather than through a consideration of definitions.

ECM, then, identifies specific aspects of a system that are incorrect from the user's perspectiveand then further analyzes whether the problem is due to the concept or the symbol representing the concept. Moreover, this last distinctions seems to map onto Moran's distinction between the semantics and the syntactics of the interface, if somewhat crudely.

9. 6 The Problems with ECM

Although this approach, of classifying problems using ECM, has been shown to be worthwhile in the preceding chapters, ECM, nevertheless, has its problems. The first, and possibly most obvious, problem with an approach that concentrates upon user-system errors is that it relies upon usability testing. Without such testing evaluative classification schemes are unlikely to be of any use. On the other hand, many I.T. companies iteratively develop their products and test them in usability laboratories. However, this raises a central problem with usability testing in general.

The problem with usability testing is that wider issues may be overlooked. That is to say, that it is possible that all of the problems that users experience may be fixed, yet a system may remain substantially unusable because the use of the system cannot be explained using a consistent conceptual model. In other words, while looking at the detail of the design we may miss the larger issues that will ultimately determine whether a system is likely to be used outside the laboratory. Eason (1984) has suggested a broad definition of usability, where its ultimate measure is whether a system is used in a work environment, and the extent to which it is underutilized. Consequently, as usability laboratoriesare limited in the data they can provide, they cannot be considered as a suitable means for testing the ultimate usability of a system, which is influenced by many factors, from the task characteristics to the user's job role (Eason, 1984).

A consideration of user-system errors, however, may encourage some to believe that it is the only data they need to consider. Although a consideration of user-system errors might prove fruitful in a number of respects, it would be a mistake to consider such an approach as the only means by which useful information might be obtained regarding human-computer interaction. Yet there is a danger that, with the wealth of qualitative and quantitative information that errors provide, some design teams might regard the analysis of user-system errors as the only analysis they need bother with. Furthermore, the insights into human-computer interaction that a consideration of errors allows, provides a seductive and potentially misleading appeal that may lead some to neglect to consider such information within a wider evaluative context.

Overall, a focus upon user-system errors may distort an overall analysis of a system. This, however, is a general criticism that can be leveled at current usability testing within the design process; it is not peculiar to analyses of user-system errors. Present evaluative methods, because they involve users who have to use a system for the first time over a relatively short time period, tend to highlight learning problems at the expense of wider issues. For example, present usability-testing methods are unlikely to demonstrate whether or not a system is likely to support the user in fulfilling overall work goals, and whether the system fitted into the current working practices of an organization. Yet there is the possibility that a design team, when equipped with a positive set of results and recommendations, might regard the data they receive from usability laboratory as the sum total of the information they require.

Nevertheless, this problem might be viewed as an ever-present problem with any sort of evaluation; where later evaluation is incorrectly viewed as being unnecessary. And, although it is certainly a problem to be borne in mine, it cannot be considered to be a serious drawback that might prevent us from using and developing further evaluative classification schemes within HCI.

However, one further problem remains, although it does not appear to be intractable. ECM has been shown to have some promise as a means by which the user's problems can be analyzed. But if we return to an earlier argument, and one that was pivotal in the discussion, we will remember that a central reason for concentrating upon user-system errors was that it was difficult to consistently predict user behaviour without some recourse to the knowledge we possess about particular tasks and the world generally. However, although ECM may have been of some use in analyzing user-system misunderstandings, it does not provide any formal means by which we might backtrack to expose the knowledge that any particular user choose to apply in any one situation. This might be viewed as a current weakness. Moreover, if such a facility were available then it might greatly increase the amount and quality of information that employing the technique might provide.

9. 7 Future Development and Research

The research reported here began from an experimental perspective. Attention was concentrated upon cognitive style measures as a means of predicting user preference and performance. However, the first study demonstrated that cognitive style was likely to have a relatively low impact upon the user's behaviour when compared to other features such as the user's mental model of the task and system. Moreover, from a practical point of view, administering cognitive style tests can be time-consuming. Essentially, cognitive style measures were judged to have a *low-impact/high-effort* problem, and it was this discovery that led to the adoption of an action-research perspective. Such a decision, however, should not be taken to imply that cognitive style is of no psychological interest or importance. As a phenomenon, cognitive style represents a challenge for cognitive science to explain, even if the practical application of the tests to HCI or other IT-related fields may be in some doubt (Huber, 1983).

Research concentrated upon the development of a technique for understanding the user's misunderstandings of the system. Essentially, this was an attempt to understand practically, if only partially, the user's mental model of a system. Approaches such as metrication (i.e.Brook, 1986), often based upon usability taxonomies such as Shackel's (1986) are aimed at awarding scores to system along a variety of 'usability' scales. This is essentially quantative evaluation, where the aim is to establish whether a system has passed certain usability criteria. However, when designs are being re-formulated formative evaluaton is required, where qualitative data is used to inform re-design (Hewett, 1986). ECM, developed from an action-research perspective, was an attempt to develop a tool that would enable analysis of this qualitative information. It was an attempt to enrich the quality of the information produced when evaluating systems. In this case *betterquality* means information that more accurately reflects the user's needs.

The action-research perspective, within HCI, emphasizes the development and testing of tools, techniques and methods that are likely to be of practical use within the design process, as well as tools, techniques and methods that are theoretically sound. Consequently, while drawing ideas from theory and established approaches, attention is directedinitially towards practical application rather than theoretical development. To some degree this approach might be viewed as a reaction against some of the theoretically laden, but practically sterile approaches that have been advanced within human-computer interaction within recent years. While the development of theory, without any practical means of applying such theory, has always been a respectable path within science, given the numbers of theories that are taken up many years later for practical use when an application has become apparent, such a concentration within a subject that is undoubtedly practical in its orientation might be viewed as unhealthy (see Norman, 1988.)

This is not to imply, however, that theory is neglected. Within an action-research approach theory development coincides with practical development, and consequently, in this thesis, while concentrating upon the development of a new technique, there has been an understandable emphasis upon the practicalities of whether such an approach might be acceptable by a design team. This is because the work reported here can, at best, only be considered as a starting point for further research in this area. A great deal remains to be done before the approach epitomized by ECM is likely to have the theoreticaland practical support that is necessary for its acceptance within the main-stream of HCI research and practice.

9. 7. 1 Cognitive grammars

Human-computer interaction, as a discipline, has witnessed a great deal of research into cognitive grammars and the like. This approach has been characterized as one driven by theory rather than practical experience. As a result, one of the greatest concerns, regarding cognitive grammars, is whether they might by usefully applied in an everyday context, outside the academic laboratory. Paradoxically, it has been these cognitive grammars that have fed into ECM the ideas and central constructs for an approachbased upon evaluative classification. As was described in chapter five, ECM embodies within it a particular view of the world, as do cognitive grammars. Moreover, the constructs (or classifications), such as operations and objects, have been taken from cognitive grammars. As stated earlier in the thesis, ECM represents an attempt to use some of the concepts and definitions developed for the early evaluation of systems (i.e. using cognitive grammars) and apply them in the context of late evaluation.

9. 7. 2 The advantages and disadvantages of ECM

As such grammars are likely to be used for analysing smaller and smaller parts of systems then ECM might be viewed as a natural development from cognitive grammars, given this increasingly accepted trend (see Moran, 1986.) Moreover, an approach based upon ECM, or an ECM-like scheme has a number of significant advantages over the use of cognitive grammars in the design and development process. Firstly, while the usability of many grammars remains to be proved, there are already indications that ECM is usable. Secondly, ECM is notably less complex than many cognitive grammars, and this may account for the positive results regarding its usability. However, this also implies that learning the ECM scheme, as well as using it, might be considerably easier than learning some grammars. Thirdly, ECM produces qualitative information, and it is this sort of information that is required to refine design (Carroll & Rosson, 1985). Many cognitive grammars produce quantitative figures (numbers of rules) for the performance of tasks. This quantitative information, however, does not necessarily help to direct design towards eventual solutions.

Nevertheless, there are significant advantages that grammars have over ECM. Firstly, ECM requires a system to evaluate, it is difficult to see how it might be used without a prototype system of some sort. Cognitive grammars analyse the task, and this analysis can be independent of the system. Secondly, ECM requires users to test the system, cognitive grammars do not involve users. On the other hand, given the variabilityofhuman behaviour, this might not necessarily be a drawback. Thirdly, the process of using ECM is likely to be more time-consuming than employing a grammar. Although the actual classification process itself is likely to take less time than analysing a significant task using a grammar, the process of selecting tasks, creating task scenarios, organizing meetings, etc. is likely to take longer. Nevertheless, one might argue that, given the wealth of information that can potentially arise from classification meetings, such an effort is justified.

9. 7. 3 Competing techniques within HCI

Overall, it is clear that cognitive grammars and an approach such as that embodied within ECM are aimed at different stages within the design process. Essentially, ECM is a tool that might be of use during usability testing and the evaluation of prototypes. As a result it is possible to view cognitive grammars and ECM as being complimentary rather than in competition. Such a view might also be taken of other evaluation techniques, such as protocol analysis, direct observation, system logging, etc. (see Sutcliff, 1988, pp183). In practice, however, such approaches and methods may compete. Eason (1984) suggests that the user, in deciding whether or not to use a system, performs an implicit cost/benefit analysis. There is no reason to believe that engineers, in evaluating a human factors technique, do not perform a similar cost/benefit analysis. Such an analysis, however, is unlikely to be performed in isolation. That is to say, that engineers will not just consider whether a method might help them to improve a design, but also whether this help might be found in other approaches, such as techniques for involving users in the design and development process. Engineers are likely to evaluate the whole range of possible means for improving the usability of their designs. They may consider, not just what each technique offers, but the effort it requires and the overlaps between techniques. Whether this is desirable, of course, is another question (Booth, 1989.)

Consequently, future research would appear to require a taxonomy of HCI techniques that might aid designers in choosing the most appropriate technique for their needs. It appears as though the basis for such a taxonomy already exists, at least in part (Booth & Marshall, 1989; Simon, 1988; Sutcliff, 1989).

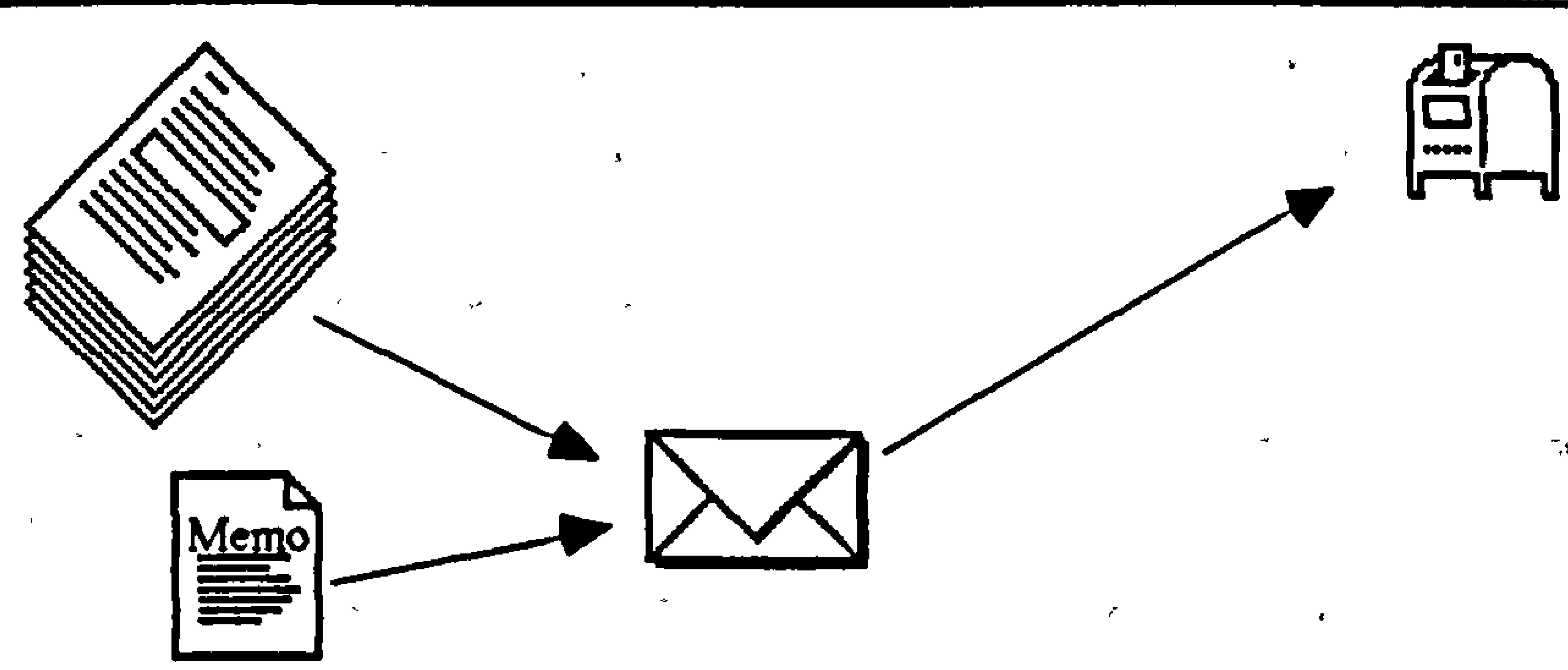
9. 7. 4 The development of ECM as an approach

Although there might be some eventual competition between grammars and ECM, if either are ever accepted within design and development, grammars might, nevertheless, feed into the development of classification schemes in the ECM vein. For example, one possibility is that ECM has fallen into the same trap the Moran's CLG (1981) has, in that it ignores the mappings between the different levels at which interaction takes place (see Young, 1981). For example, a scheme based upon simple tasks and rule schemata (Payne & Green, 1986) might be worthy of investigation.

9. 7. 5 The development of ECM as a classification scheme

Using grammars to provide the basis for new classification schemes, however, is to suggest how ECM and an approach might be developed. It may be helpful if the development of the ECM scheme itself is also addressed. From a practical standpoint a number of question seem likely to arise the mind of any potential user of ECM. For example, when should ECM be used? Can ECM be used on software products other than office automation systems? How might the user's views be best elicited? (To date this has only been assumed within the technique—it has never been explicitly addressed.) How might the classification discussion of the design team be best conducted? These and other questions require further studies; essentially more practical tests. We might expect the results of these tests to ultimately be embodied within guidelines as to how best to employ ECM, or any derivative of it.

More immediately, two studies would seem to present themselves as vehicles for further investigation. The first is essentially a replication of the study reported in chapter 8 of this thesis, where a small system was shown to be improved as a result of the applicationof ECM. Such a study always raises questions as to the influence of ECM on the eventual designs. For example, how might we know whether the differences were due to ECM or the capabilities of the designers, or to the focus on errors generally (personal communication, Clayton Lewis, 1989). A replication might also attempt to refine a larger, more complex system, as the system used in the study reported in chapter 8 was restricted in its functionality. Specifically, this study should attempt to investigate the way in which ECM might feed into re-design. An explanation and example of how ECM should feed into design is provided in figure 9.4.



A representation of the task a user was attempting on an pre-release version of Hewlett Packard's Newwave system. The icons in this figure at not the same as the icons on the Newwave system, although the concepts they represent are similar.

This example is taken direct from the usability trials at Hewlett Packard (reported in chapter 7). The user was attempting a series of tasks on a new icon-based electronic mail system. The user had just placed a memo and a report into an envelope (see picture above). The user complained that he would have liked to have been able to number the different items so that the receiver would know which to read first. This was classified by one team member as having two mismatched elements (hence two classifications). The first was an operation-symbol mismatch: the user wanted the items to be ordered, and indeed they were. Their ordering depends upon the order in which they are placed in the envelope. In other words, there was an operation-symbol mismatch as the user was not aware that it was possible to order the contents of an envelope, and clearly this needs to be communicated to the user in some way. The second element in the failure was classified as an operation-concept mismatch. Although it was possible to order the items in the envelope by placing the items in the envelope in a particular order, the user appeared to want to manipulate the ordering of items once they were in the envelope, and this operation was not possible in this system.

A less structured analysis of the user's problem might have returned the answer "he can order the items, he just hasn't read the manual." However, by classifying the different elements of the problem it is possible to identify the user's requirements of the system in more detail. The first classification (the operation-symbol mismatch) suggested that the user needed to be informed that items are already ordered in the envelope. The second classification (the operation-concept mismatch) suggested that there needs to be another operation where the user can re-order items once they are in the envelope. It is in this way that an ECM classification leads into re-design, by helping the design team to identify more accurately the user's requirements of the system. Of course, how the user might be informed of the ordering of items, or how the operation to re-order items might be implemented both require the technical and imaginative skills of the designer. ECM is not intended as a tool to replace the creative and technical skills of the designer, its aim is to augment and direct these skills by elucidating the user's requirements.

Figure 9.4: An explanation and example of how ECM might feed into design.

The second study might concentrate upon the potential value of the design recommendations produced by differing evaluative techniques. This would initially entail a qualitative analysis of the outputs of differing techiques (including ECM). Such a study might help to place various evaluative techniques in an overall perspective, where their respective benefits could be viewed in the context of a general taxonomy of HCI techniques suggested above.

Moreover, the classification scheme itself might need to be amended. This might not, however, necessarily entail adding categories to the scheme. One of the most attractive features of the ECM scheme was its fundamental simplicity. Consequently, it is difficult to see how ECM might be fruitfully extended as a classification scheme without making it less usable. It seems unlikely that a more accurate scheme with 30 categories would be used. Nevertheless, this does not prohibit the ECM scheme being developed extended in terms of the theory and reasoning that underlies it.

9. 7. 6 Knowledge sets and user's models

Clearly, any future work must address the deficits of the present scheme. The problem of the classification not being practically exclusive is one that might need to be considered further. The major area of concern, however, is the extent to which ECM encourages a piecemeal analysis. In other words, ECM does not support backtracking to expose the knowledge that the user choose to apply. If this were done, then it might be possible to see where, and in what situations, the user choose to apply inappropriate items of knowledge. This information might then be used to refine the system to ensure that the user was not encouraged to apply inappropriate knowledge and make unuseful deductions.

To do this, however, may entail a more explicit consideration of the user's model of the task and system. A theoretical framework for this already exists, at least in part. This framework is based around the idea of *knowledge sets* rather than models. The idea being that we can have a model of the user's *knowledge set,* but the user does not have a model of the task or system. In essence, the suggestion is that we replace the idea of a *model* with the idea of a *knowledge set*. To some extent this is a simple case of changing terminology. On the other hand, it provides a more accurate perspective on the user's representation of the task and system.

Atpresent user's are considered to have models of the task and system. This term *model,* however, implies something that is complete and coherent at some level of abstraction. Furthermore, it implies something which is uniform in its abstraction. None of these things can be said of user's *models*. As Norman (1981) and others have pointed out, user's models are inconsistent, incomplete, confusable, etc. Indeed, it is questionable whether the term *model* is really the most appropriate.

A further argument against the idea that users have models is that if we really did have models of everything we used in the world then this would be a very difficult and inefficient means of representing everything. In truth, we use what we understand about a cooker to understand a toaster and visa versa. If we did have separate models how might we generalize about artifacts of different models without some exterior mechanism that transcends the cognitive representation?

If we do not have models, then what do we have? It is suggested that we have items or chunks of knowledge that apply to whole varieties of situations and tasks. For any particular task we might have a *knowledge set*. This is a collection of the various pieces of knowledge. The important point is that knowledge chunks are re-usable. That is to say, that one chunk of knowledge might belong to a number of different knowledge sets.

When a user encounters a system for the first time he or she cannot possibly have a *model* of the system in the way that we usually use the term. The user has, instead, an incomplete and probably inappropriate knowledge set. This initial set is constructed from the user's expectations about the machine and its environment. As the user learns more about the system then the knowledge set is refined, in a process that is not dissimilar to natural selection. That is to say, that those chunks of knowledge that lead to inappropriate actions are selected out. At first chunks of knowledge that are not appropriate are discarded, while more useful chunks are recruited from memory. Gradually the become more consistent and complete. At some stage the knowledge chunks themselves are altered in order that they might better represent the system in question. This, of course, leads to confusion later when the same knowledge chunk is used in a different knowledge set, but has been modified.

In this way such a framework can account for learning by limited association, and for the errors we make when transferring between systems. Such a framework, it is suggested, might form a useful basis in considering what we would have formally called the *user's model*. In essence, this framework, it is hoped, might allow different user-system errors to be connected to expose underlying flaws and inconsistencies within the system, that may not be revealed by just one or two dialogue failures. The idea is that, by analysing the user-system errors that occur, we might backtrack to expose those items of knowledge that were inappropriately applied to the situation. This might show us whether particular items of knowledge are being consistently applied in an inappropriate way or at inappropriate times. The mechanism by which this might be achieved, however, remains unclear and must be the subject of future research.

9. 8 The Implications for Human-Computer Interaction

The major theme, throughout this thesis, has been one of shifting emphasis, where concentration upon user-system errors rather than upon predicting behaviour using cognitive grammars or tests has been viewed as being more profitable. It has been suggested that approaches that attempt to predict the user might have a number of drawbacks, not least of which being that the tests or grammars used do not have, embodied within them, the knowledge we use on an everyday basis to predict behaviour. The first experiment lent weight to the idea that behaviour at the interface is unlikely to be determined by single over-riding factors, such as cognitive style, but comes about as a result of a subtle and sometimes complex interaction of a number of factors.

An approach was suggested where analysis concentrated upon dialogue failures or user-system errors at the interface. It was proposed that the basis for this analysis, in line with approaches to human error, might lie with evaluative classification schemes. Such a scheme was developed as a means of investigating the potential worth of such an approach. Although the scheme had a number of drawbacks, the overall conclusion was that evaluative classification schemes for user-system errors have a potentially useful role to play in the development of usable computer systems. Furthermore, it is suggested that approaches to considering user-system errors, employing evaluative classification schemes, could form a valuable and productive area of research and application within human-computer interaction research and development.

Achenbach, T., M., (1969) Cue learning, associative responding, and school performance in children. *Developmental psychology, 1,* 717-725.

Adelson, B., (1985) Comparing natural and abstract categories: A case study from computer science. *Cognitive Science, 9,* 417-430.

Adelson, B., (1984) When novices surpass experts: The difficulty of a task may increase with expertise. *Journal of Experimental Psychology: Learning, Memory & Cognition, 10(3),* 483-495.

Adorno, T., W., Frenkel-Brunswick, E., Levinson, D., J., & Sanford, R., N., (1950) *The authoritarian personality.* Harper Row, New York.

Arnheim, R. (1969) *Visual Thinking.* Berkeley: University of California Press.

Bakan, P., & Strayer, F., F., (1973) On reliability of conjugate lateral eye movements. *Perceptual and motor skills, 36,* 429-430.

Baker, C. A. (1977) *An investigation of man-computer interaction in an on-line bibliographic retrieval system.* Loughborough University of Technology Library.

Bannon, L. J. (1986a) Helping users help each other. In: D. A. Norman & S. W. Draper (Eds.), *User centred system design: New perspectives on human-computer interaction.* Hillsdale, New Jersey, USA: Lawrence Erlbaum Associates.

Barnard, P. J. & Hammond, N. V. (1983) *Cognitive contexts and interactive communication.* IBM Hursley Human Factors Laboratory Report.

Barnard, P. J. & Hammond, N. V. (1982) Usability and its multiple determination for the occasional user of interactive systems. In: Williams (Ed.), *Pathways to the information society: proceedings of the 6th international conference on computer communication, London, UK.* Amsterdam,Netherlands:North-Holland.

Barnard, P. J., Hammond, N. V., Morton, J. Long, J. B. & Clark, I. A. (1981) Consistency and compatibility in human-computer dialogue. *International Journal of Man-Machine Studies, 15,* 87-134.

Bartlett, F., C., (1932) *Remembering: A study in experimental and social psychology.* Cambridge University Press, Cambridge.

Bellotti, V. (1988) Implications of current design practice for the use of HCI techniques. In: D. M. Jones & R. Winder (Eds.), *People and Computers IV, proceedings of the fourth conference on human-computer interaction.* Cambridge, UK: Cambridge University Press.

Benbasat, I., & Schroeder, R., G., (1977) An experimental investigation of some MIS design variables. *The Management Information Systems Quarterly, 1,* 37-49.

Bennett, J. L. (1984) Managing to meet usability requirements: establishing and meeting software development goals. In: J. L. Bennett, D. Case, J. Sandelin & M. Smith (Eds.), *Usability issues and health concerns.* Englewood cliffs, New Jersey, USA: Prentice-Hall.

Bertino, E. (1985) Design issues in interactive user interfaces. *Interfaces in Computing, 3,* 37-53.

Bechtel, W. (1988) *Philosophy of mind: An overview for cognitive science..* Lawrence Erlbaum Associates.

Bjørn-Andersen, N. (1986) Understanding the nature of the office for the design of third wave office systems. In: M. D. Harrison & A. F. Monk (Eds.), *People & Computers: Designing for*

Usability. Proceedings of the Second Conference of the BCS HCI specialist group. Cambridge, UK: Cambridge University Press.

Bjørn-Andersen, N. (1985) Are 'human factors' human? In: N. Bevan & D. Murray (Eds), *Man-Machine Interaction: State of the art report.* Maidenhead, UK: Pergamon Infotech.

Booth, P. A. (1989) *An Introduction to Human-Computer Interaction.* Lawrence Erlbaum Associates: Hove, Sussex.

Booth, P. A., & Marshall, C. J. (1989) Usability in human-computer interaction. In: P. A. Booth, *An Introduction to Human-Computer Interaction.* Lawrence Erlbaum Associates: Hove, Sussex.

Bradshaw, J., L., & Nettleton, N., C., (1981) The nature of hemispheric specialisation in man. *The behavioural and brain sciences, 4,* 51-91.

Branscomb, L. M. (1983) The computers debt to science. *Perspectives in Computing, 3,* 4-19.

Briggs, P. (1989) What we know and what we need to know: The user model versus the user's model in human-computer interaction. *Behaviour and Information Technology. 7 (4),* 431-442.

Brook, J. B. (1986) Usability engineering in office production development. In: M. D. Harrison & A. F. Monk (Eds.), *People & Computers: Designing for Usability. Proceedings of the Second Conference of the BCS HCI specialist group.* Cambridge, UK: Cambridge University Press.

Bruner, J., S., Goodnow, J., & Austin, G., A., (1956) *A study of thinking.* Wiley, New York.

Bury, K. F. (1984) The iterative development of usable computer interfaces. In: B. Shackel (Ed.), *Human-computer interaction - Interact '84: Proceedings of the first IFIP conference on human-computer interaction, London.* Amsterdam, Netherlands: North-Holland.

Card, S. K., Moran, T. P. & Newell, A. (1983) *The psychology of human-computer interaction.* Hillsdale, New Jersey, USA: Lawrence Erlbaum Associates.

Card, S. K., Moran, T. P. & Newell, A. (1980) Computer text-editing: An information-processing analysis of a routine cognitive skill. *Cognitive Psychology, 12,* 32-74.

Carroll, J. M. & Rosson, M. B. (1985) Usability specifications as a tool in iterative development. In: H. R. Hartson (Ed.), *Advances in Human-Computer Interaction.* Norwood, New Jersey, USA: Ablex.

Carroll, J. M., Thomas, J. C. & Malhotra, A. (1980) Presentation and representation in design problem solving. *American Journal of Psycholgy, 93,* 269-284.

Castell, A., (1986) MSc Project at Loughborough University, Department of Human Sciences.

Cheng, P. W., Holyoak, K. J., Nisbett, 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) Categorisation and representation of physics problems by experts and novices. *Cognitive Science, 5(2),* 121-152.

Clark, F., Drake, P., Kapp, M. & Wong, P. (1984) User acceptance of information technology through prototyping. In: B. Shackel (Ed.), *Human-computer interaction - Interact '84: Proceedings of the first IFIP conference on human-computer interaction, London.* Amsterdam, Netherlands:North-Holland.

Clarke, A. A. (1986) A three-level human-computer interface model. *International Journal of Man-Machine Studies, 24,* 503-547.

Clowes, I. (1987) User models. Talk given at Alvey Human Interface Club meeting, 11th May. Reported by K. Robinson in *Alvey Newsletter.*

Damodaran, L. (1983) User involvement in system design. *Data Processing, 25 (6),* 6-13.

Davis, R., (1983a) User error or computer error? Observations on a statistics package. *International Journal of man-machine studies, 19,* 359-376.

Davis, R., (1983b) Task analysis and user errors: a methodology for assessing interactions. *International Journal of man-machine studies, 19,* 561-574.

Drake, D., M., (1970) Perceptual correlates of impulsive and reflective behaviour. *Developmental Psychology,* 2.

Eason, K. D. (1985a) Emerging concepts in the implementation of I. T. systems. In: N. Bevan & D. Murray (Eds.), *Man-Machine Interaction: State of the art report.* Maidenhead, UK: Pergamon Infotech.

Eason (1985b) *User analysis, user participation and organizational change.* Notes distributed during workshop given at HCI '85 conference, University of East Anglia, Norwich.

Eason, K. D. (1984) Towards the Experimental Study of Usability. *Behaviour and Information Technology. 3 (2),* 133-143.

Eason, K. D. (1976) Understanding the naive computer user. *The Computer Journal, 19 (1),* 3-7.

Eason, K. D., Damodaran, L. & Stewart, T. F. M. (1975) Interface problems in Man-Computer Interaction. In: E. Mumford & Sackman (Eds.), *Human Choice and Computers.* Amsterdam, Netherlands:North-Holland.

Ericsson, K. A. & Simon, H. A. (1980) Verbal reports as data. *Psychological Review, 87 (3),* 215-251.

Fowler, C., J., H., Macauley, L., A., & Fowler, J., (1985) The Relationship between Cognitive Style and Dialogue Style: An Explorative Study. In: P. Johnson & S. Cook (Eds.), *People and Computers: Designing the interface.* Cambridge, UK: Cambridge University Press.

Freud, S. (1922) *Introductory lectures on psychoanalysis.* London, UK: George Allen & Unwin.

Galton, F., (1887) Notes on the prehension of idiots. *Mind, 12,* 79-82.

Golla, F., L., & Antonvitch, S., (1929) The respiratory rhythm in its relation to the mechanism of thought, *Brain, 52,* 491-509.

Gould, J. D. (in press) How to design usable systems. In: M. Helander (Ed.), *Handbook of Human-Computer Interaction.* Amsterdam,Netherlands:North-Holland.

Gould, J. D. & Lewis, C. H. (1985) Designing for usability - - key principles and what designers think. *Communications of the ACM, 28,* 300-311.

Graf, W., Elsinger, P. & Krueger, H. (1987) Methods for the ergonomic evaluation of alphanumeric computer-generated displays. In: H. J. Bullinger & B. Shackel (Eds.), *Human-computer interaction - Interact '87: Proceedings of the second IFIP conference on human-computer interaction, Stuttgart.* Amsterdam, Netherlands: North-Holland.

Green, T. R. G. (1987) Limited theories as a framework for human-computer interaction. Invited address to the Austrian Computer Society's 6th Annual Interdisciplinary workshop. *Mental models and human-computer interaction.* Schårding, Austria, 9th-12th June, 1987.

Green, T. R. G. (1986) Cognitive aspects of HCI. *Computer Bulletin, September.*

Griffitts, C., H., (1927) Individual differences in imagery, *Psychological monographs, 37* (172).

Habermas, J. (1981) *The Theory of Communicative Action: One, Reason and the Rationalization of Society.* English translation 1984. Massachusetts, USA: Beacon Press.

Hammond, N. V. & Barnard, P. J. (1985) Dialogue design: Characteristics of user knowledge. In: A. F. Monk (Ed.), *Fundamentals of human-computer interaction.* London, UK: Academic Press.

Hammond, N. V., Barnard, P. J., Clark, I. A., Morton, J. & Long, J. B. (1980) *Structure and content in interactive dialogue.* IBM Hursley Human Factors Report, HF034.

Hammond, N. V., Hinton, G., Barnard, P. J., MacLean, A., Long, J. & Whitefield, A. (1984) Evaluating the interface of a document processors: a comparison of expert judgement and user observation. In: B. Shackel (Ed.), *Human-computer interaction - Interact '84: Proceedings of the first IFIP conference on human-computer interaction, London.* Amsterdam, Netherlands: North-Holland.

Hammond, N. V., Jorgensen, A., MacLean, A., Barnard, P. J. & Long, J. (1983a) Design practice and interface usability: Evidence from interviews with designers. In: A. Janda (Ed.), *Human factors in computer systems: Proceedings of the CHI '83 conference, Boston.* Amsterdam, Netherlands: North-Holland.

Hammond, N. V., Morton, J., Barnard, P. J., Long, J. B. & Clark, I. A. (1987) Characterizing user performance in command-driven dialogue. *Behaviour & Information Technology, 6.*

Hammond, N. V., Morton, J., MacLean, A. & Barnard, P. J. (1983b) Fragments and signposts: Users models of systems, In: *Proceedings of the 10th International Conference on Human Factors in Telecommunications, Helsinki, June.* 81-88.

Heider, E., R., (1971) Information processing and the modifictation of an impulsive conceptual tempo. *Child Development, 42,* 1276-1281.

Hewett, T. T. (1986) The role of iterative evaluation in designing systems for usability. In: M. D. Harrison & A. F. Monk (Eds.), *People & Computers: Designing for Usability. Proceedings of the Second Conference of the BCS HCI specialist group.* Cambridge, UK: Cambridge University Press.

Hiltz, S., R,. (1980) Some results of the evaluation of the operational trials of the Electronic Information Exchange System. In: Benenfeld & Kazlauskas (eds) Communicating Information: *Proceedings of the 43rd ASIS Annual meeting 17. Published: Knowledge Ind Publications* Inc, White Plains NY USA. 1981.

Huber, G. P. (1983) Cognitive Style as a basis for MIS and DSS Designs: Much ado about nothing. *Management Science 29 (5),* 567-579.

Johnson-Laird, P. N. (1988) *The Computer and the Mind: An Introduction to Cognitive Science.* London: Fontana Press.

Johnson-Laird, P. N. (1983) *Mental Models: Towards a cognitive science of language, inference, and consciousness.* Cambridge, UK: Cambridge University Press.

Johnson-Laird, P. N. (1981) Mental models in cognitive science. In: D. A. Norman (Ed.), *Perspectives on Cognitive Science.* Norwood, New Jersey, USA: Ablex.

Kagan, J., (1966) Refection-Impulsivity: The Generability and Diagnosis of Conceptual Tempo. *Journal of Abnormal Psychology, 71,* 17-22.

Kagan, J., Pearson, L. & Welch, L., (1966) Conceptual Impulsivity and Inductive Reasoning. *Child Development, 37,* 583-594.

Kagan, J., (1965) Individual differences in the resoloution of response uncertainty. *Journal of Personality and Social Psychology. 2,* 154-160.

Kagan, J., Rosman, B., C., Day, D., Albert, J., & Phillips, W., (1964) Information processing in the child. *Psychological Monographs, 78.*

Kieras, D. E. & Polson, P. G. (1985) An approach to the formal analysis of user complexity.

International Journal of Man-Machine Studies, 22 (4), 365-394.

Kidd, A. (1982) Problems in man-machine dialogue design. *International Conference on Man-Machine Systems, 6-9th July 1982. UMIST, Manchester.*

Kocel, K., Galin, D., Ornstein, R., & Merrin, E., L., (1972) Lateral eyemovements and cognitive mode. *Psychonomic Science, 27,* 223-224.

Kogan, N., (1971) Educational implications of cognitive styles. In G. S. Lesser (ed) *Psychology and educationalpractice.* Glenview, Illinois: Scott, Foresman and Company.

Lakoff, G. (1987) *Women, fire and dangerous things: What caotgoeirs reveal about the mind.* Chicago, USA: Chicago University Press.

Lewis, B. N., (1976) Avoidance of aptitude treatment trivialities. In Messick, S. (ed) (1976) *Individuality in Learning.* Jossey-Bass, San Francisco.

Lewis, C. & Norman, D. A. (1986) Designing for error. In: D. A. Norman & S. W. Draper (Eds.), *User centred system design: New perspectives on human-computer interaction.* Hillsdale, New Jersey, USA: Lawrence Erlbaum Associates.

Lucas, H., C., (1979) An experimental investigation of the use of graphics in decision-making. 11-17, School of Business, New York University, cited in Powers *et al.* (1984).

Macaulay, L. A., Fowler, C. J. H. & Hutt, A. (1986) Human factors in the IT specification process. *Computer Bulletin, 2 (3),* 10-12.

MacLean, A., Barnard, P. J. & Wilson, M. (1986) Rapid prototyping of dialogue for human factors research: the EASIE approach. In: M. D. Harrison & A. F. Monk (Eds.), *People & Computers: Designing for Usability. Proceedings of the Second Conference of the BCS HCI specialist group.* Cambridge, UK: Cambridge University Press.

Marsden, P. H. (1989) *Modelling the retrieval of incomplete knowledge.* PhD Thesis, Department of Psychology, University of Manchester, UK.

Marshall, C. J., Christie, B. & Gardiner, M. M. (1987a) Assessment of trends in the technology and techniques of human-computer interaction. In: M. M. Gardiner & B. Christie (1987) *Applying cognitive psychology to user interface design.* Chichester, UK: Wiley.

Marshall, C. J., Nelson, C. & Gardiner, M. M. (1987b) Design guidelines. In: M. M. Gardiner & B. Christie (Eds.), *Applying cognitive psychology to user-interface design.* Chichester, UK: Wiley.

Martin, A. (1972) A new keyboard layout. *Applied Ergonomics, 3 (1).*

Mayer, R. E. (1981) *The promise of cognitive psychology.* San Francisco, USA: Freeman.

McClelland, J. L., Rumelhart, D. E. & the PDP Research Group, (1986a) *Parallel distributed processing: Explorations in the microstructure of cognition. 1: Foundations.* Cambridge, Massachusetts, USA: MIT Press.

McClelland, J. L., Rumelhart, D. E. & the PDP Research Group, (1986b) *Parallel distributed processing: Explorations in the microstructure of cognition. 2: Psychological and biological models.* Cambridge, Massachusetts, USA: MIT Press.

McKeithen, K., B., Reitman, J., S., Rueter, H., H., & Hirtle, S., C., (1981) Knowledge organisation and skill differences in computer programmers. *Cognitive Psychology, 13,* 307-325.

McKenna, F., P., (1984) Measures of field dependence: a cognitive style or cognitive ability? *Journal Personality and Social Psychology, 47,* 593-A.

Messick, S. (ed) (1976) *Individuality in Learning.* Jossey-Bass, San Francisco.

Mick, D., (1980) Human Factors in Information Work. In: Benenfeld & Kazlauskas (eds)

Communicating Information: Proceedings of the 43rd ASIS Annual meeting 17. Published: Knowledge Ind Publications Inc, White Plains NY USA. 1981.

Miller, G. A., & Johnson-Laird, P. N. (1976) *Language and Perception.* Cambridge University Press.

Miyata, Y. & Norman, D. A. (1986) Psychological issues in support of multiple activities. In: D. A. Norman & S. W. Draper (Eds.), *User centred system design: New perspectives on human-computer interaction.* Hillsdale, New Jersey, USA: Lawrence Erlbaum Associates.

Monk, A. F. (1986) Mode errors: a user-centred analysis and some preventative measures using key-contingent sound. *International Journal Man-Machine studies, 24,* 313-327.

Moran, T. P. (1986) Analytical performance models: A contribution to a panel discussion. In: M. Mantei & P. Orbeton (Eds.), *Human factors in computer systems - III: Proceedings of the CHI '86 Conference, Boston.* Amsterdam, Netherlands: Elsevier.

Moran, T. P. (1983) Getting into a system: External-internal task mapping analysis. In: A. Janda (Ed.), *Human factors in computer systems: Proceedings of the CHI '83 conference, Boston.* Amsterdam,Netherlands:North-Holland.

Moran, T. P. (1981) The command language grammar: A representation for the user interface of interactive computer systems. *International Journal of Man-Machine Systems, 15 (1),* 3-50.

Morgan, C. C. & Sufrin, B. A. (1984) Specification of the Unix filing system. *IEEE Transactions on Software Engineering, SE-10 (2),* 128-142.

Morton, J., Barnard, P. J., Hammond, N. V. & Long, J. B. (1979) Interacting with a computer: A framework. In: E. J. Boutmy & A. Danthine (Eds.), *Teleinformatics '79.* Amsterdam, Netherlands:North-Holland.

Newell, A. & Simon, H. A. (1972) Human problem solving. Englewood Cliffs, New Jersey, Prentice-Hall.

Nickerson, R. S. (1986) *Using computers: Human factors in information systems.* Cambridge, Massachusetts, USA: MIT Press.

Nielsen, J. (1986) A virtual protocol model for computer-human interaction. *International Journal for Man-Machine Studies, 24,* 301-312.

Norman, D. A. (1988) *The Psychology of Everyday Things.* New Yourk, USA: Basic Books.

Norman, D. A. (1987) Cognitive engineering-cognitive science. In: J. M. Carroll (Ed.), *Interfacing thought: Cognitive aspects of human-computer interaction.* Cambridge, Massachusetts, USA: MIT Press.

Norman, D. A. (1986) Cognitive Engineering. In: D. A. Norman & S. W. Draper (Eds.), *User centred system design: New perspectives on human-computer interaction.* Hillsdale, New Jersey, USA: Lawrence Erlbaum Associates.

Norman, D. A. (1983a) Design principles for human-computer interfaces. In: A. Janda (Ed.), *Human factors in computer systems: Proceedings of the CHI '83 conference, Boston.* Amsterdam, Netherlands:North-Holland.

Norman, D. A. (1983b) Design rules based on analyses of human error. *Communications of the ACM, 26 (4),* 254-258.

Norman, D. A. (1983c) Some observations on mental models. In: D. Gentner & A. L. Stevens (Eds.), *Mental Models.* Hillsdale, New Jersey, USA: Lawrence Erlbaum Associates.

Norman, D. A. (1981a) A psychologist views human processing: human errors and other phenomena suggest processing mechanisms. In: *Proceedings of the International Joint Conference on*

Artificial Intelligence, Vancouver.

Norman, D. A. (1981b) Twelve Issues for Cognitive Science. In: D. A. Norman (Ed.), *Perspectives on Cognitive Science.* Norwood, New Jersey, USA: Ablex.

Norman, D. A. (1981c) What is cognitive science?. In: D. A. Norman (Ed.), *Perspectives on Cognitive Science.* Norwood, New Jersey, USA: Ablex.

Norman, D. A. & Draper, S. W. (Eds.) (1986) *User centred system design: New perspectives on human-computer interaction.* Hillsdale, New Jersey, USA: Lawrence Erlbaum Associates.

Paivio, A., (1971) *Imagery and verbal processes.* New York, Holt, Rinehart & Winston.

Payne, S. J. (1984) Task-action grammars. In: B. Shackel (Ed.), *Human-computer interaction - Interact '84: Proceedings of the first IFIP conference on human-computer interaction, London.* Amsterdam,Netherlands:North-Holland.

Payne, S. J. & Green, T. R. G. (1986) Task-action grammars: A model of the mental representation of task languages. *Human-computer interaction, 2,* 93-133.

Polson, P. G. (1987) A quantitative theory of human-computer interaction. In: J. M. Carroll (Ed.), *Interfacing thought: Cognitive aspects of human-computer interaction.* Cambridge, Massachusetts: MIT Press.

Powers, M., Lashley, C., Sanchez, P., & Shneiderman, B., (1984) An experimental comparason of tabular and graphic data presentation. *International Journal of Man-machine Studies, 20,* 545-566.

Ragan, T. J., Ausburn, L., J., Huckebey, K., Black, K., T., Ausburn, F., B., Burkett, J., R., Stennell, V., & Burler, P., A., (1979) *Cognitive Styles: A Review of the Literature.* Brooks Air Force Base, Texas.

Rasmussen, J. (1986) *Information processing and human-machine interaction.* Amsterdam, The Netherlands: North Holland.

Rasmussen, J. (1980) What can be learned from human error reports? In: K. Duncan, M. Gruneberg & D. Wallis (Eds.), *Changes in working life.* London, UK: Wiley.

Rasmussen, J. (1976) Outlines of a hybrid model of the process operator. In: T. Sheridan G. Johannsen (Eds.), *Monitoring behaviour and supervisory control.* New York, USA: Plenum.

Reason, J. T. (1989) *Human Error.* Cambridge, UK: Cambridge University Press.

Reason, J. T. (1988) Cognitive underspecification: Its varieties and consequences. In: B. Baars (Ed.), *The psychology of error.* London, UK: Plenum.

Reason, J., (1983) Absent-mindedness and cognitive control. In: Morris (Ed) *Slips, lapses and memory in everyday life.* Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Reason, J. & Mycielska, K. (1982) *Absent-minded? The psychology of mental lapses and everyday errors.* Englewood Cliffs, New Jersey, USA: Prentice-Hall.

Rector, A. L., Newton, P., Marsden, P. H. (1985) What kind of system does an expert need? In: P. Johnson & S. Cook (Eds.), *People and Computers: Designing the interface.* Cambridge, UK: Cambridge University Press.

Reisner, P. (1987) Discussion: HCI, what is it and what research is needed? In: J. M. Carroll (Ed.), *Interfacing thought: Cognitive aspects of human-computer interaction.* Cambridge, Massachusetts, USA: MIT Press.

Richards, J. T., Boies, S. J. & Gould, J. D. (1986) Rapid prototyping and system development: Examination of an interface toolkit for voice and telephony applications. In: M. Mantei & P.

Orbeton (Eds.), *Human factors in computer systems - III: Proceedings of the CHI '86 Conference, Boston.* Amsterdam, Netherlands: Elsevier.

Richardson, A. (1977) Verbalisers-Visualisers: A Cognitive Style dimension. *Journal of Mental Imagery, 1,* 109-125.

Riesbeck, C., K., (1984) Knowledge reorganisation and reasoning style. *InternationalJournal of Man-Machine Studies, 20,* 45-61.

Riley, M. & O'Malley, C. (1984) Planning nets: A framework for analyzing user-computer interactions. In: B. Shackel (Ed.), *Human-computer interaction - Interact '84: Proceedings of the first IFIP conference on human-computer interaction, London.* Amsterdam, Netherlands: North-Holland.

Robertson, I. T. (1982) Individual Differences in Information Processing Strategy and Style. *International Conference on Man-Machine Systems. 6-9 July1982.* UMIST, Manchester.

Robertson, I. T., (1985) Human Information Processing Strategies and Style, *Behaviour and Information Technology, 4(1),* 19-29.

Roe, A., (1951) A study of imagery in research scientists. *Journal of Personality, 19,* 459-470.

Rosch, E. (1975) Classification of real-world objects: Origins and representations in cognition.

Rubinstein, R. & Hersh, H. (1984) *The human factor: Designing computer systems for people.* Burlington, MA, USA: Digital Press.

Sato, K., (1983) Reflection-impulsivity as a differentiation of decision criterion under decision-maing situations. *Japanese Journal of Educational Psychology, 31(3),* 186-194.

Searle, J. R. (1980) Minds, brains and programs. *Behavioural and Brain Sciences, 3,* 417—424.

Searle, J. R. (1979) *Expression and meaning: Studies in the theory of speech acts.* Cambridge: Cambridge University Press.

Shackel, B. (1986) Ergonomics in design for usability. In: M. D. Harrison & A. F. Monk (Eds.), *People & Computers: Designing for Usability. Proceedings of the Second Conference of the BCS HCI specialist group.* Cambridge, UK: Cambridge University Press.

Shackel, B. (1985) Ergonomics in information technology in Europe-a review. *Behaviour & Information Technology, 4 (4),* 263-287.

Shackel, B. (1984) Information technology-a challenge to ergonomics and design! *Behaviour & Information Technology, 3 (4),* 263-275.

Shackel, B. (1981) The concept of usability. In: *Proceedings of the IBM software and information usability symposium.* 1-30. Ploughkeepsie, New York, USA.

Sheehy, N. P. (1987) Nonverbal behaviour in dialogue. In: R. G. Reilly (Ed.), *Communication failure in dialogue and discourse.* Amsterdam, Netherlands: Elsevier.

Short, P., L., (1953) The objective study of mental imagery. *British Journal of Psychology, 44,* 38-51.

Simon, T., & Young, R. M. (1988) GOMS meets STRIPS: The integration of planning with skilled procedure execution in human-computer interaction. In: D. M. Jones & R. Winder (Eds.), *People and Computers IV, proceedings of the fourth conference on human-computer interaction.* Cambridge, UK: Cambridge University Press.

Spiegler, I., (1983) Modeling Man-Machine Interface in a Database Enviroment. *International Journal of Man-Machine Studies. 18,* 55-70.

Storrs, G., Rivers, R. & Canter, D. (1984) The future on man-machine interface research: a discussion and a framework for research. *Applied Ergonomics, 15 (1),* 61-63.

Sufrin, B. A. (1986) Formal methods and the design of effective user interfaces. In: M. D. Harrison & A. F. Monk (Eds.), *People & Computers: Designing for Usability. Proceedings of the Second Conference of the BCS HCI specialist group.* Cambridge, UK: Cambridge University Press.

Sutcliff, A. G. (1989) Task analysis, systems analysis and design: Symbiosis or synthesis? *Interacting with Computers, 1(1),* 6—12.

Sutcliff, A. G. (1988) *Human-Computer Interface Design.* London, UK: MacMillan.

Twigger, D. (1986) *Usability: definition, operationalization, measurement and impact on software design.* Internal report, HCI Research Unit, Huddersfield Polytechnic, UK.

Weiser, M., & Shertz, J., (1983) Programming problem representation in novice and expert programmers. *International Journal of Man-Machine Studies, 19,* 391-398.

Welty, C., (1985) Correcting user errors in SQL. *International Journal, of Man-Machine Studies, 22,* 463-477.

Whiteside, J. & Wixon, D. (1987) Discussion: Improving human-computer interaction - a quest for cognitive science. In: J. M. Carroll (Ed.), *Interfacing thought: Cognitive aspects of human-computer interaction.* Cambridge, Massachusetts, USA: MIT Press.

Wickens, C., D., & Kramer, A., (1985) Engineering Psychology. *Annual Review of Psychology, . 36,* 307-348.

Witkin, H., A., & Goodenough, D., R., (1981) *Cognitive Styles: Essence and origins.* Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Witkin, H., A., Goodenough, D., R., & Oltman, P., K., (1979) Psychological Differentiation: Current Status. *Journal of Personality & Social Psychology, 37(7),* 1127-1145.

Witkin, H., A., Moore, C., A., Owen, D., R., Raskin, E., Oltman, P., K., Goodenough, D., R., & Friedman, F., (1977) Role of the Field-Dependent and Field-Independent Cognitive Styles in Academic Evoloution: A Longitudinal Study. *Journal of Educational Psychology, 69(3),* 197-211.

Witkin, H., A., (1976) Cognitive styles in learning and teaching. In Messick, S. (ed) (1976) *Individuality in Learning.* Jossey-Bass, San Francisco.

Witkin, H. A., (1950) Individual differences in ease of perception of embedded figures. *Journal of Personality, 19,* 1-15.

Wixon, D., Whiteside, J., Good, M. & Jones, S. (1983) Building a user defined interface. In: A. Janda (Ed.), *Human factors in computer systems: Proceedings of the CHI'83 conference, Boston.* Amsterdam,Netherlands:North-Holland.

Wundt, W. (1907) *Outlines of Psychology.* Third edition. Leipzig: Engelmann.

Yando, R., M., & Kagan, J., (1968) The effect of teacher tempo on the child. *Child Development, 39,* 27-34.

Young, R. M. (1985) User models as design tools for software engineers. Paper given at Alvey workshop, September 1985, Loughborough, UK.

Young, R. M. (1981) The machine inside the machine: Users models of pocket calculators. *International Journal of Man-Machine Studies, 15,* 87-134.

Young, R. M. & Harris, J. E. (1986) A viewdata-structure editor designed around task/action mapping. In: M. D. Harrison & A. F. Monk (Eds.), *People & Computers: Designing for Usability. Proceedings of the Second Conference of the BCS HCI specialist group.* Cambridge, UK: Cambridge University Press.

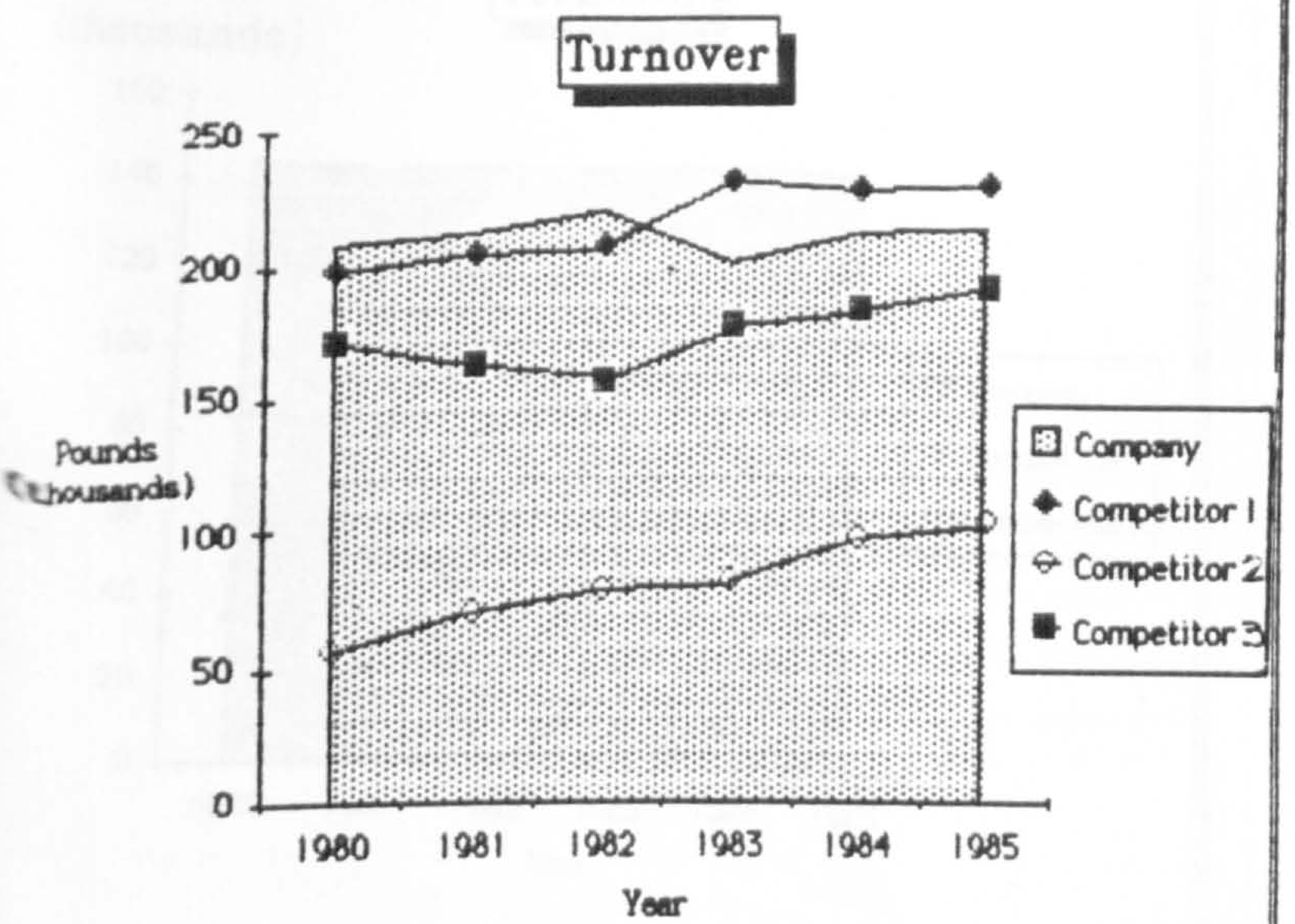
Appendix One:

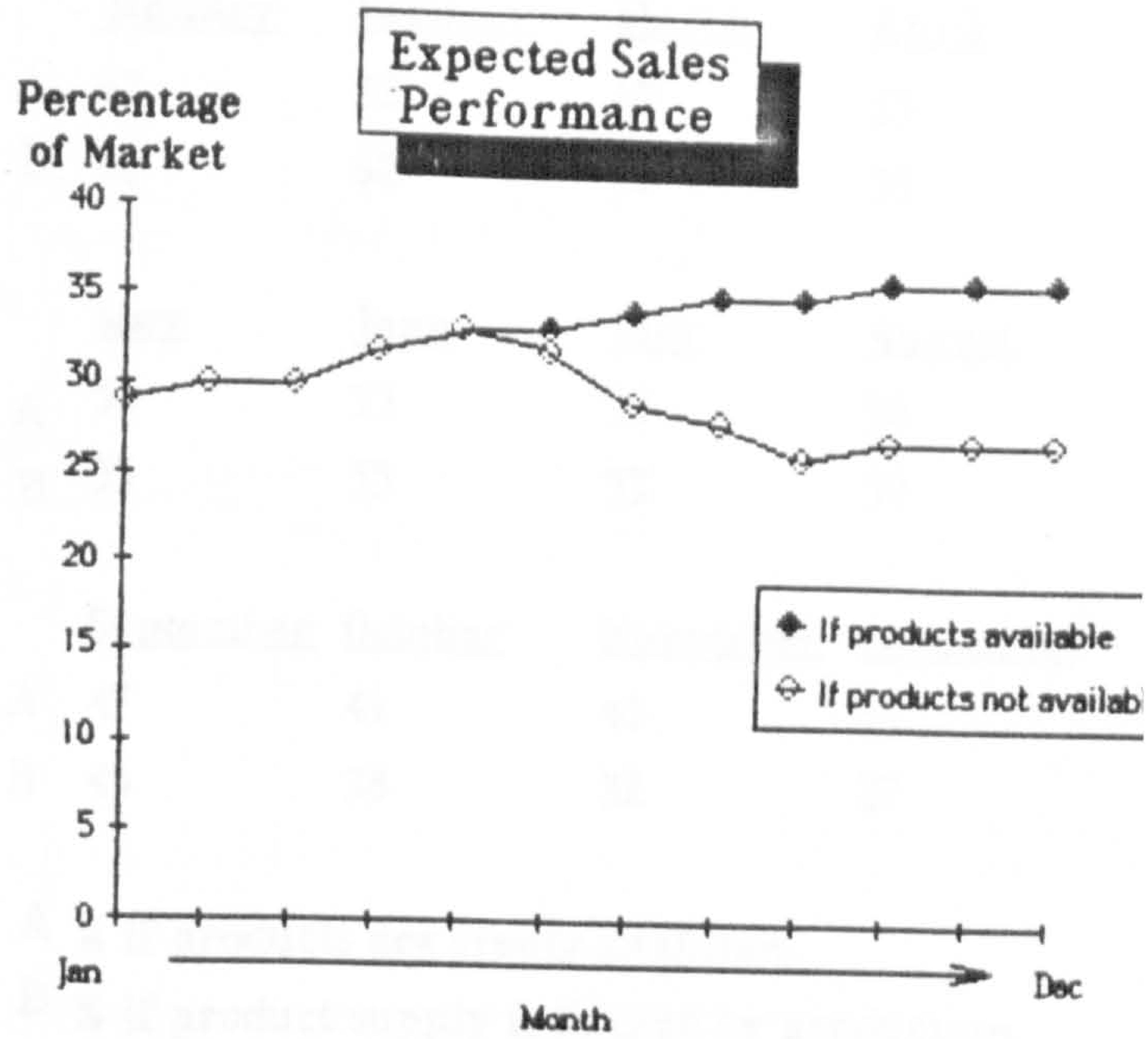
The twelve presentations used for the first experiment.

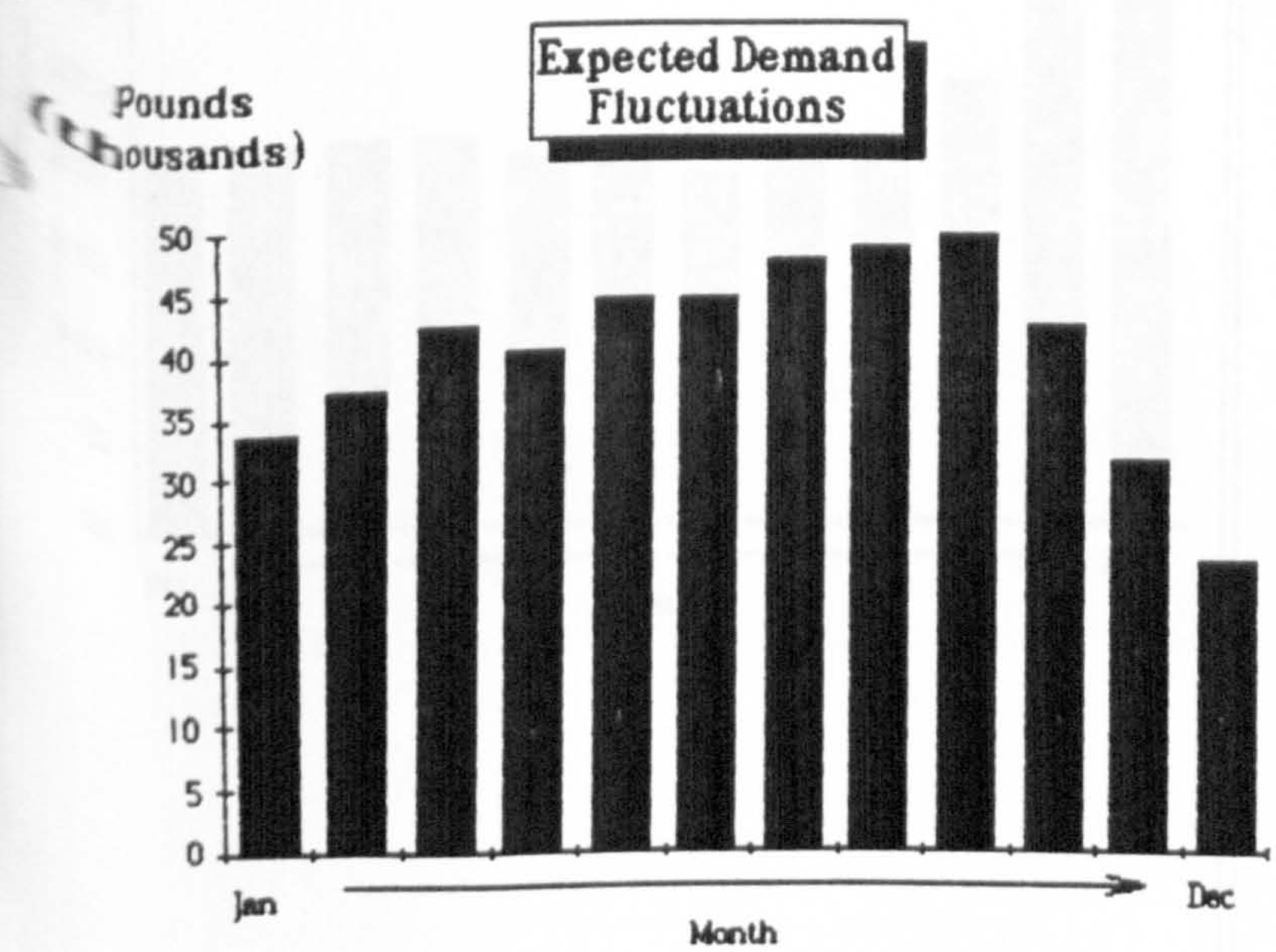
TEXT BOUND INTO

THE SPINE

Product 1





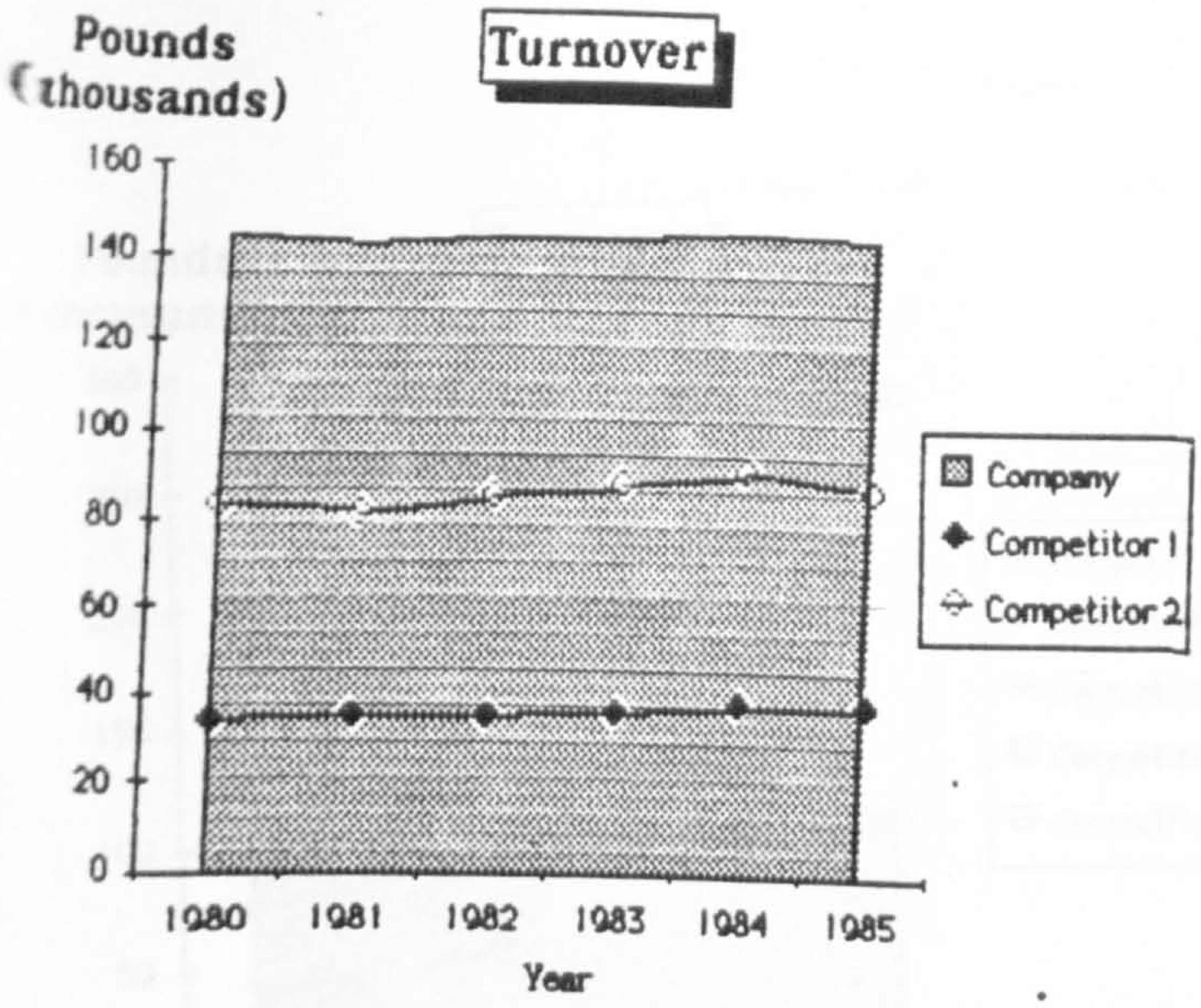


Expected Production Costs

Stable production	£350	per unit
+/- 10% fluctuation	£370	per unit
+/- 20% fluctuation	£420	per unit
+/- 30% fluctuation	£390	per unit
+/- 40% fluctuation	£385	per unit

Selling price-	£450
Break even point-	£400
Monthly storage cost per unit-	£5
Number of units per year-	480,000

Product 2



Expected Sales Performance

Percentage of Market

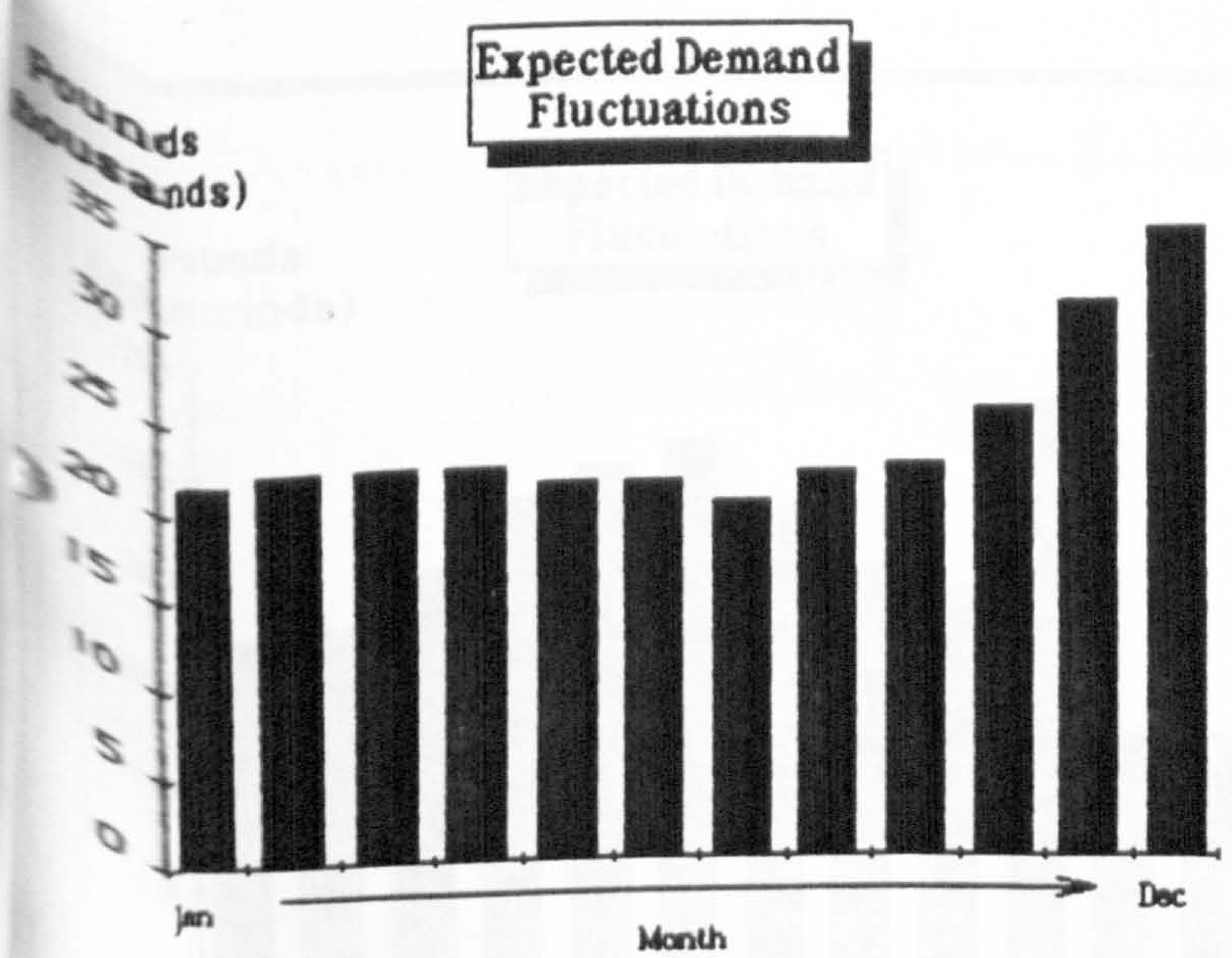
	January	Febuary	March	April
A	62	62	62	55
B	62	62	62	55

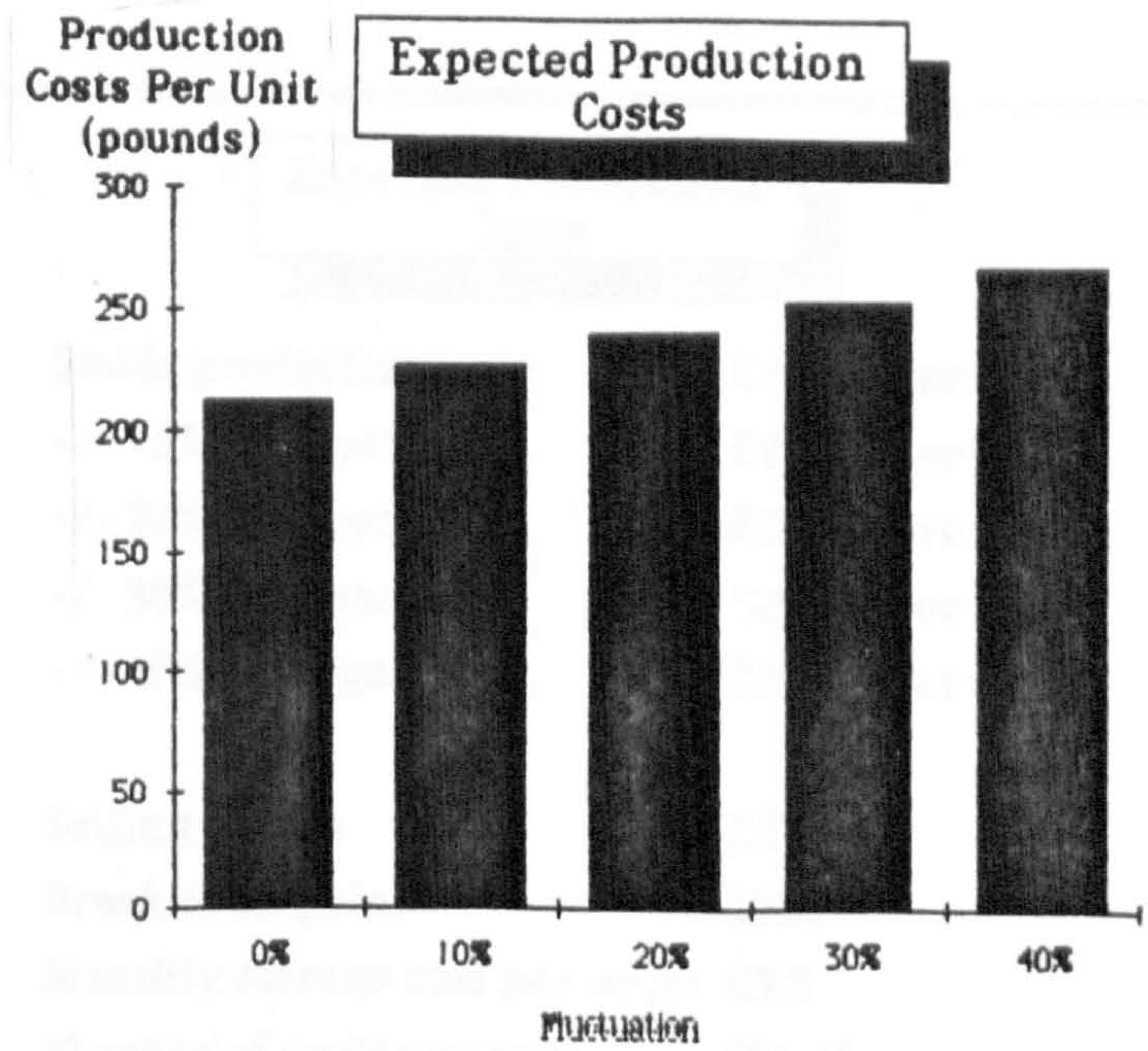
	May	June	July	August
A	55	55	55	50
B	55	55	55	50

	September	October	November	December
A	47	41	41	41
B	45	38	32	27

A % if products are freely availible.

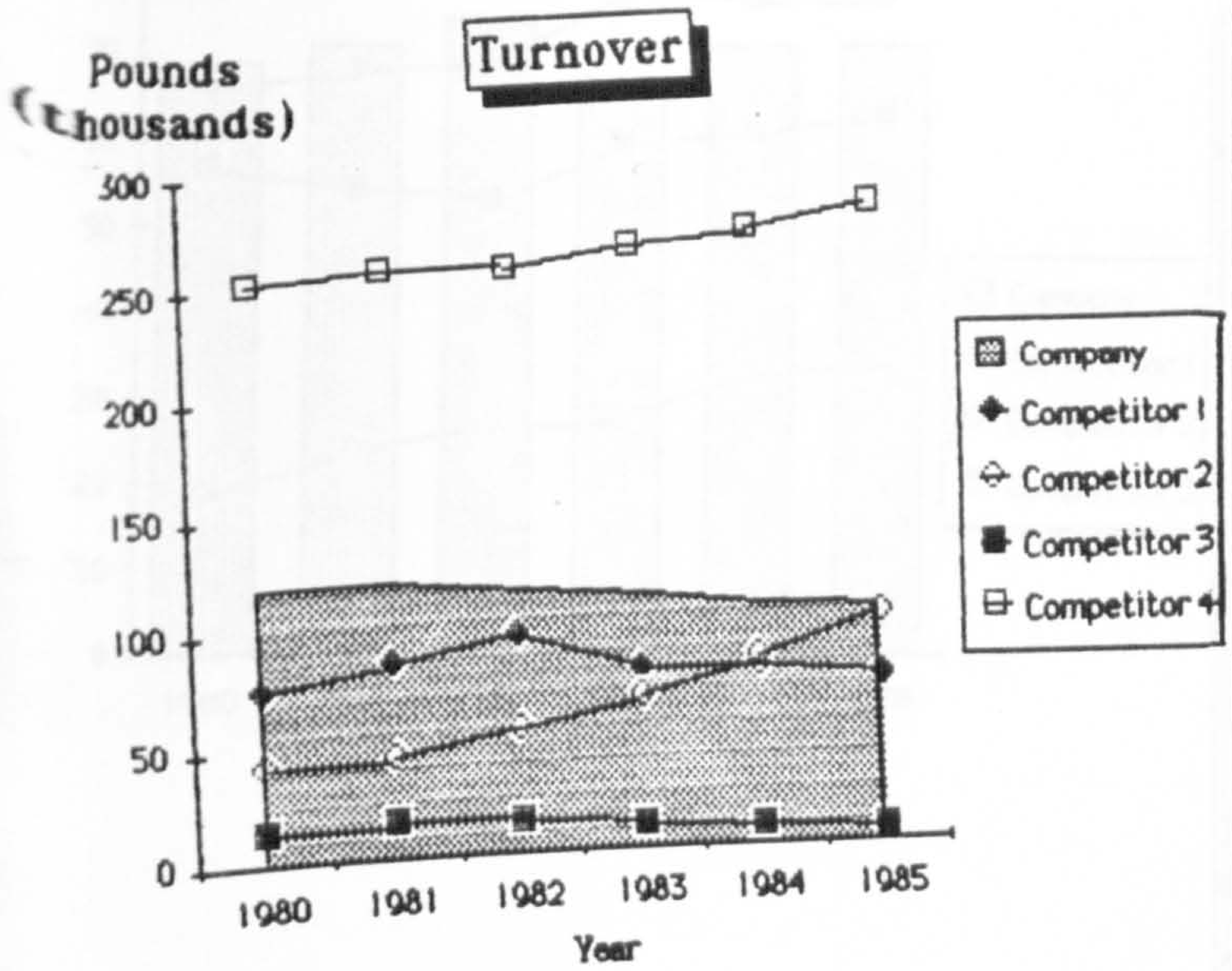
B % if product supply is limited by production.

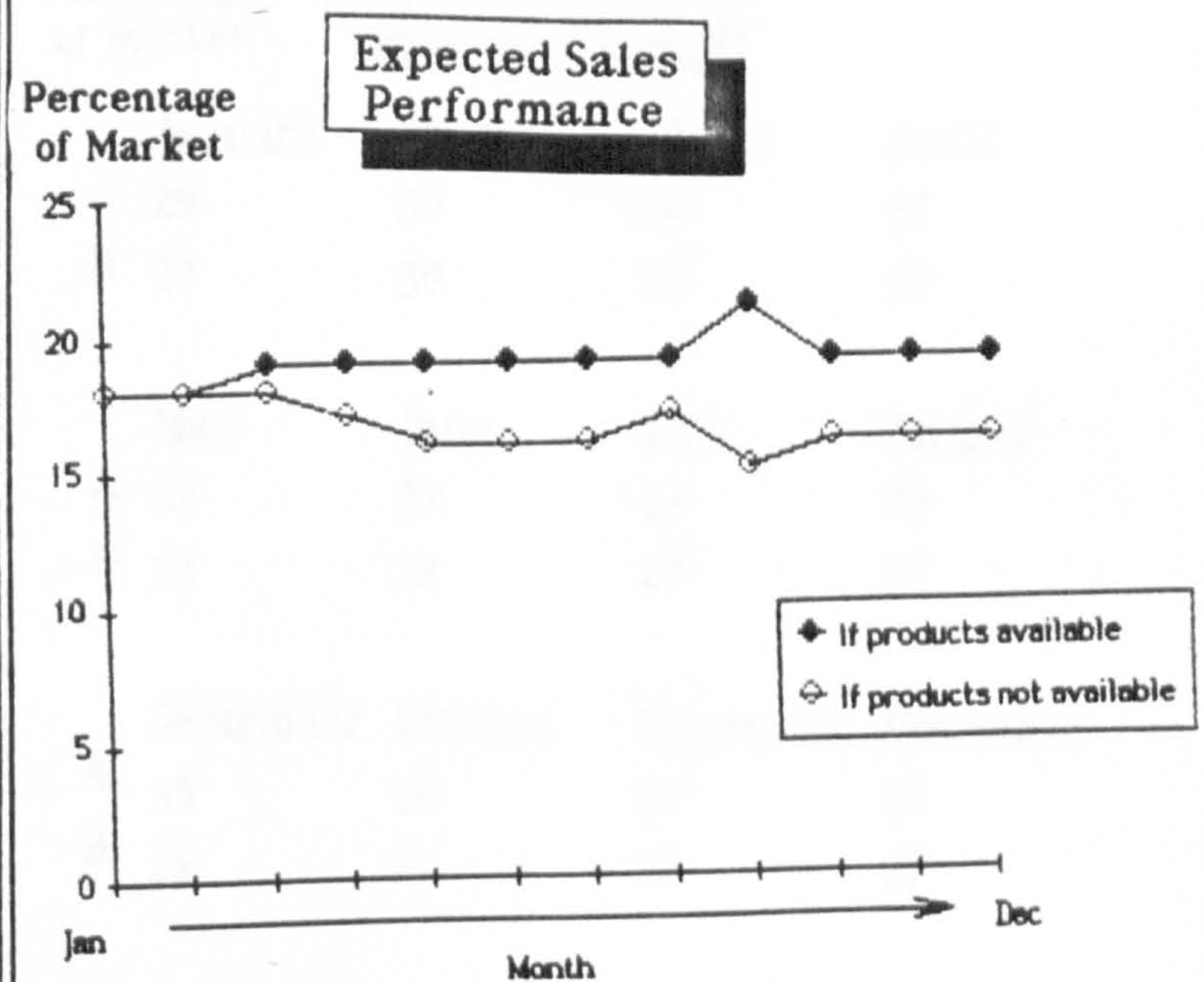


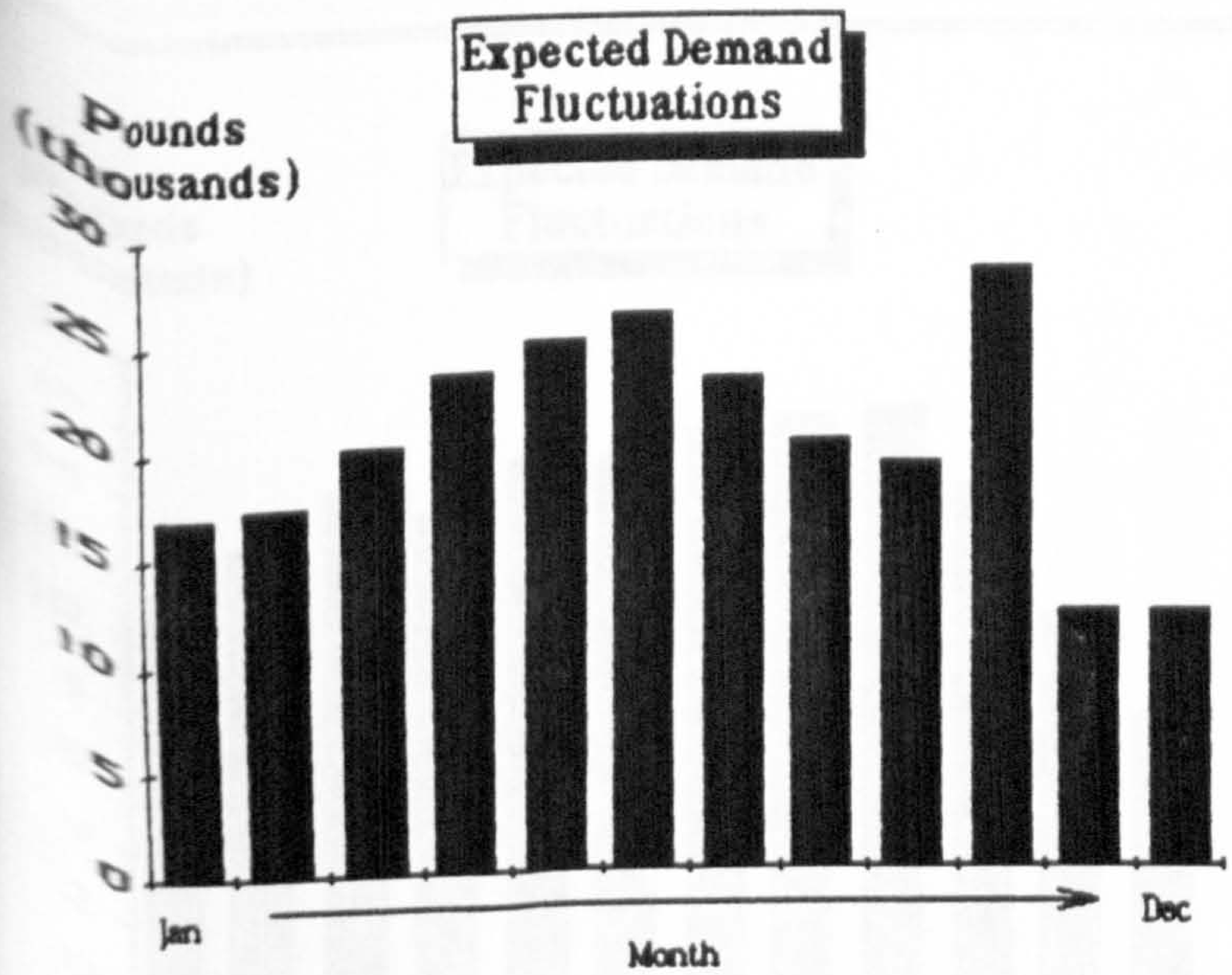


Selling price=	£334
Break even point=	£280
Monthly storage cost per unit=	£1.2
Number of units per year=	283,000

Product 3







Expected Production Costs

Stable production	£200	per unit
+/- 10% fluctuation	£270	per unit
+/- 20% fluctuation	£263.5	per unit
+/- 30% fluctuation	£257	per unit
+/- 40% fluctuation	£263.5	per unit

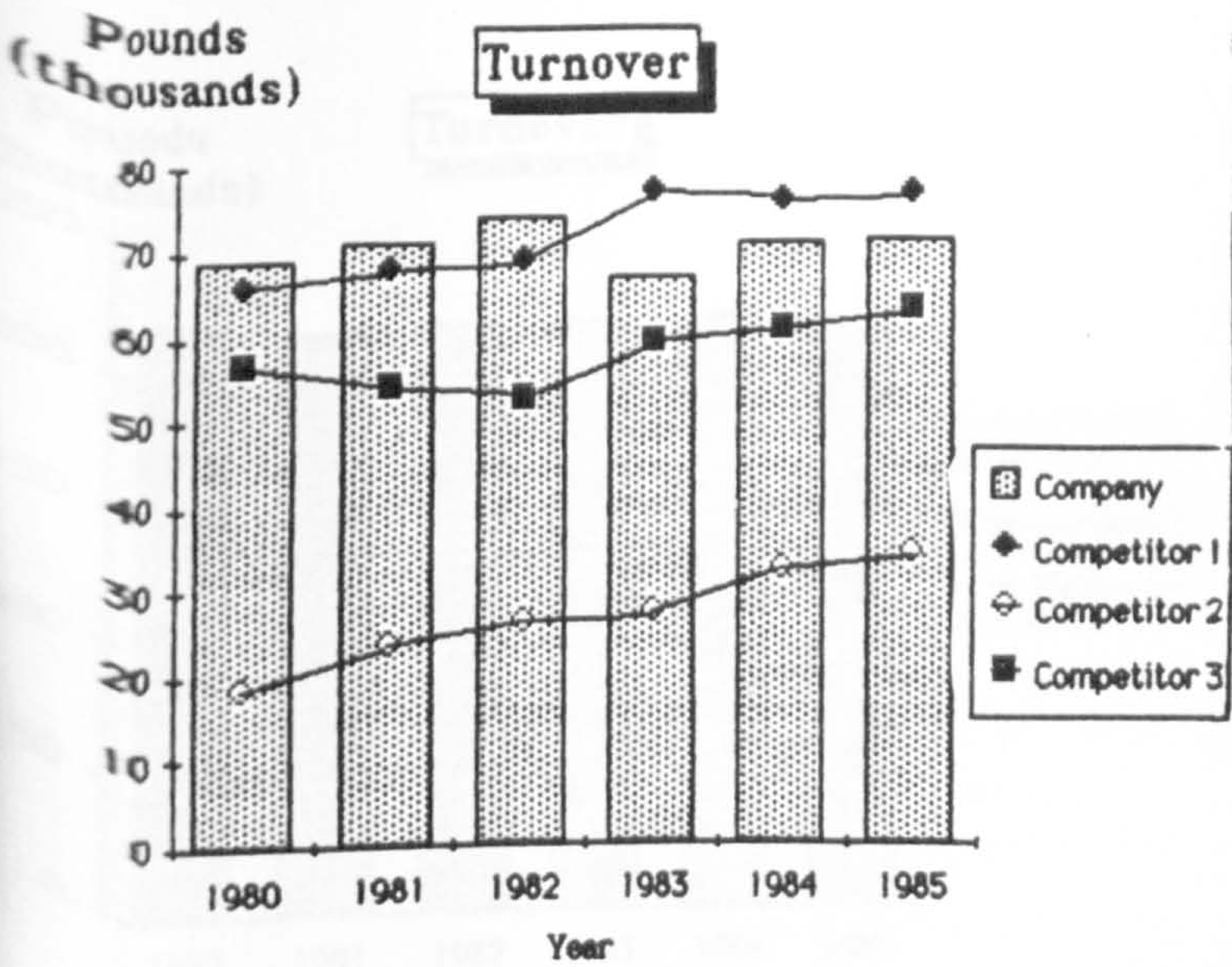
Selling price= £277

Break even point= £260

Monthly storage cost per unit= £5.5

Number of units per year= 288,000

Product 4



Expected Sales Performance

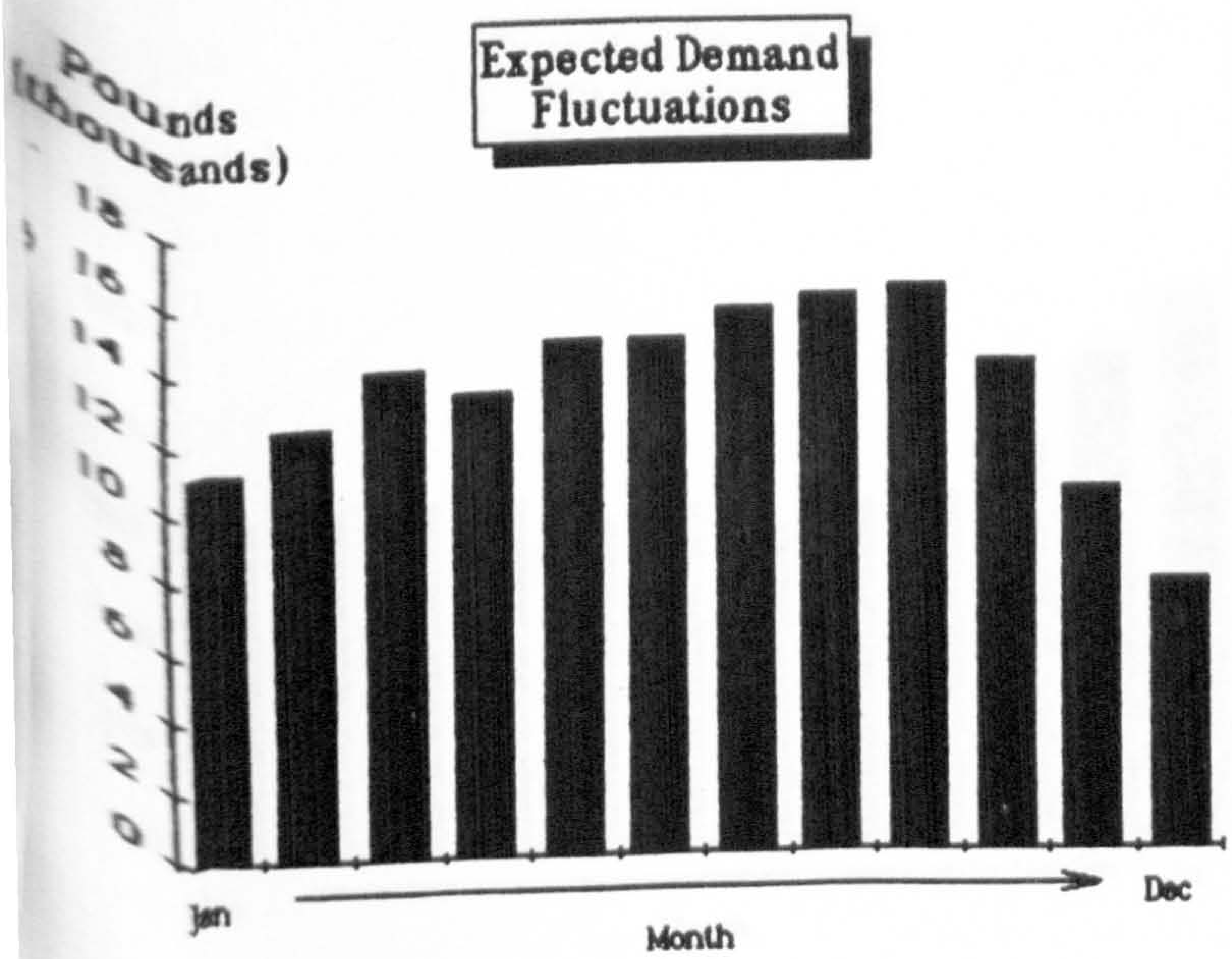
Percentage of Market

	January	Febuary	March	April
A	29	30	30	32
B	29	30	30	32

	May	June	July	August
A	33	33	34	35
B	33	32	29	28

	September	October	November	December
A	35	36	36	36
B	26	27	27	27

A % if products are freely availible.
B % if product supply is limited by production.

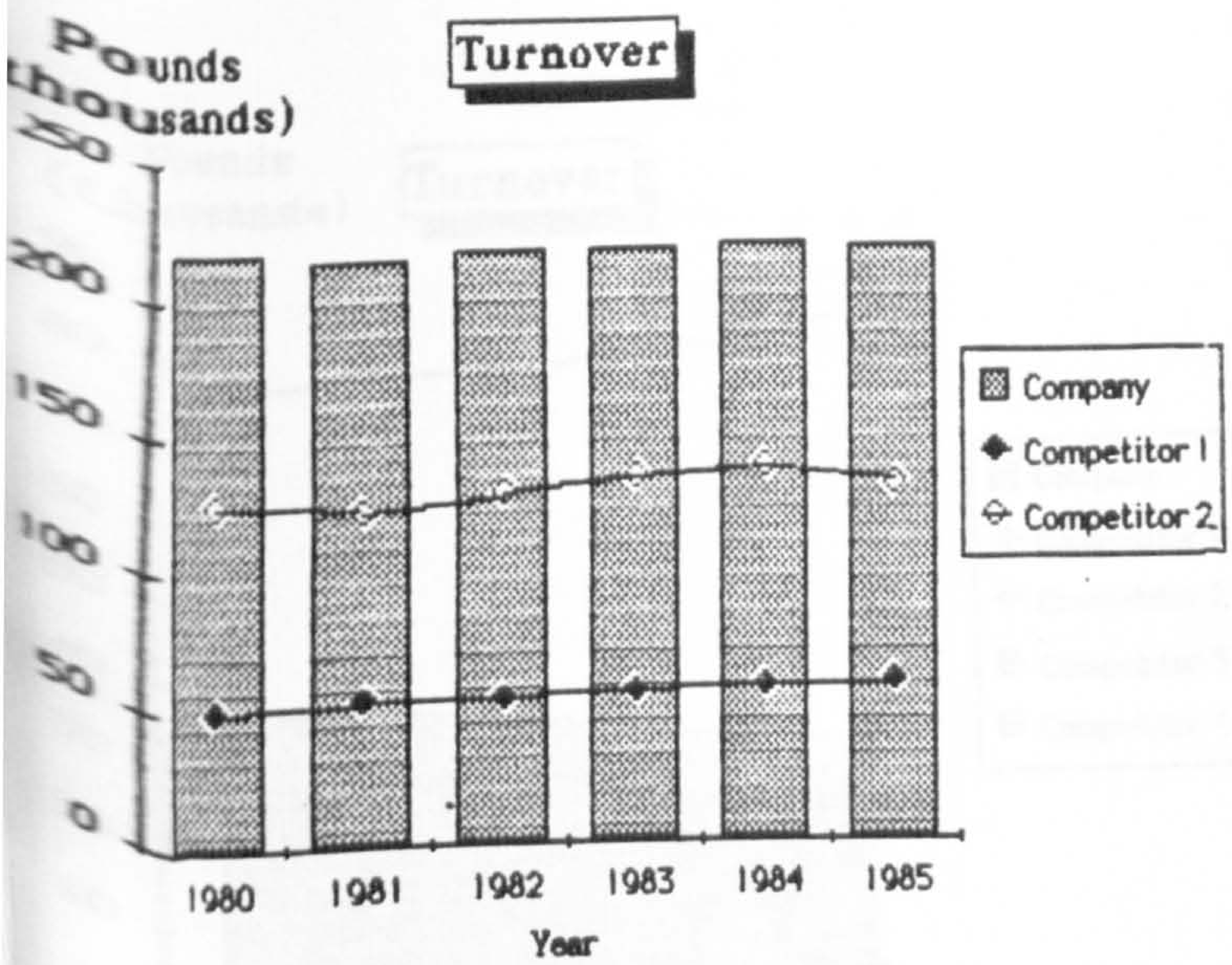


Expected Production Costs

Stable production	£117	per unit
+/- 10% fluctuation	£123	per unit
+/- 20% fluctuation	£140	per unit
+/- 30% fluctuation	£130	per unit
+/- 40% fluctuation	£128	per unit

Selling price= £150
Break even point= £133
Monthly storage cost per unit= £1.7
Number of units per year= 160,000

Product 5



Expected Sales Performance

Percentage of Market

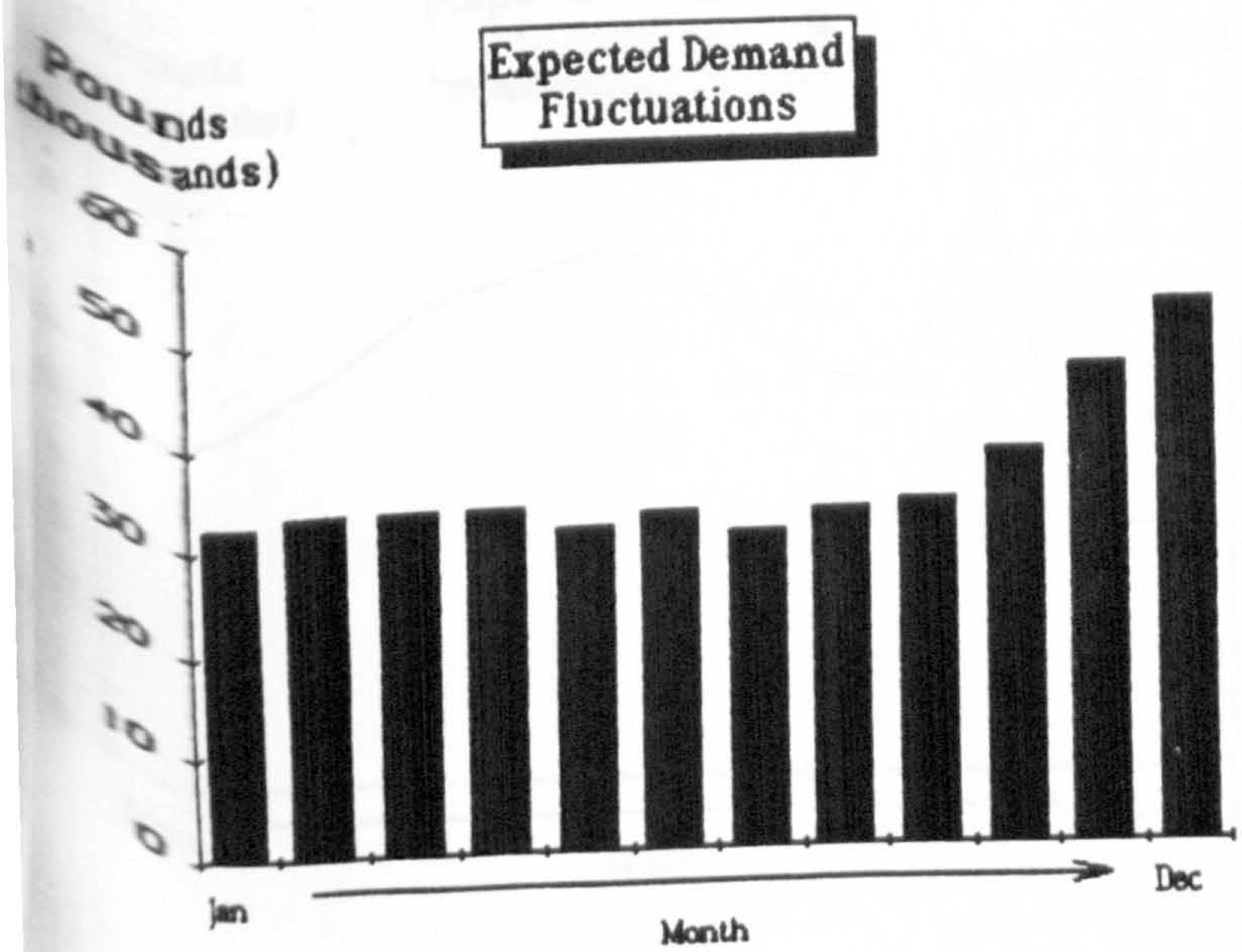
	January	Febuary	March	April
A	29	30	30	32
B	29	30	30	32

	May	June	July	August
A	33	33	34	35
B	33	32	29	28

	September	October	November	December
A	35	36	36	36
B	26	27	27	27

A % if products are freely availible.

B % if product supply is limited by production.

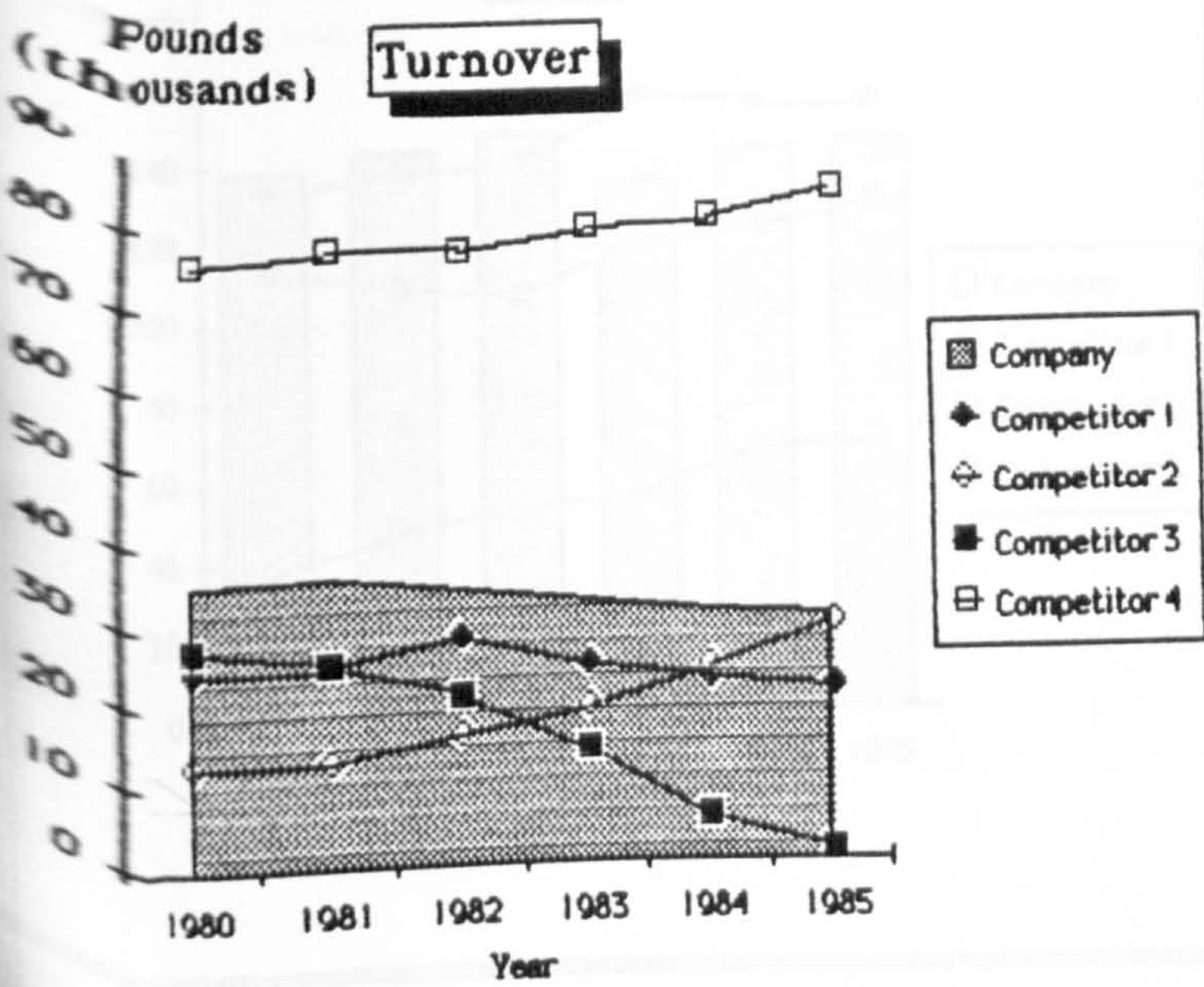


Expected Production Costs

Stable production	£117	per unit
+/- 10% fluctuation	£123	per unit
+/- 20% fluctuation	£140	per unit
+/- 30% fluctuation	£130	per unit
+/- 40% fluctuation	£128	per unit

Selling price=	£150
Break even point=	£133
Monthly storage cost per unit=	£1.7
Number of units per year=	160,000

Product 6



Expected Sales Performance

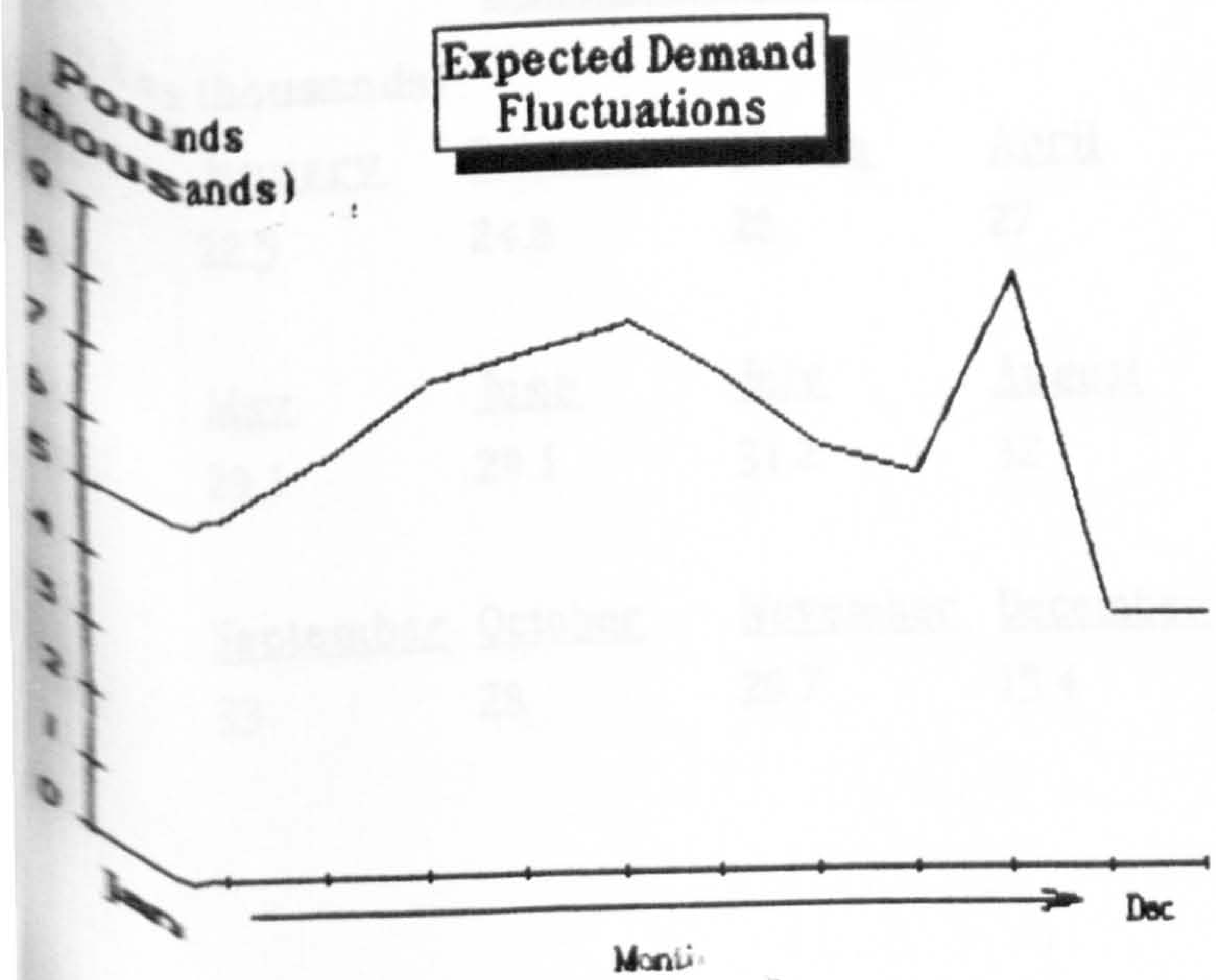
Percentage of Market

	January	Febuary	March	April
A	18	18	19	19
B	18	18	18	17

	May	June	July	August
A	19	19	19	19
B	16	16	16	17

	September	October	November	December
A	21	19	19	19
B	15	16	16	16

A % if products are freely availible.
B % if product supply is limited by production.



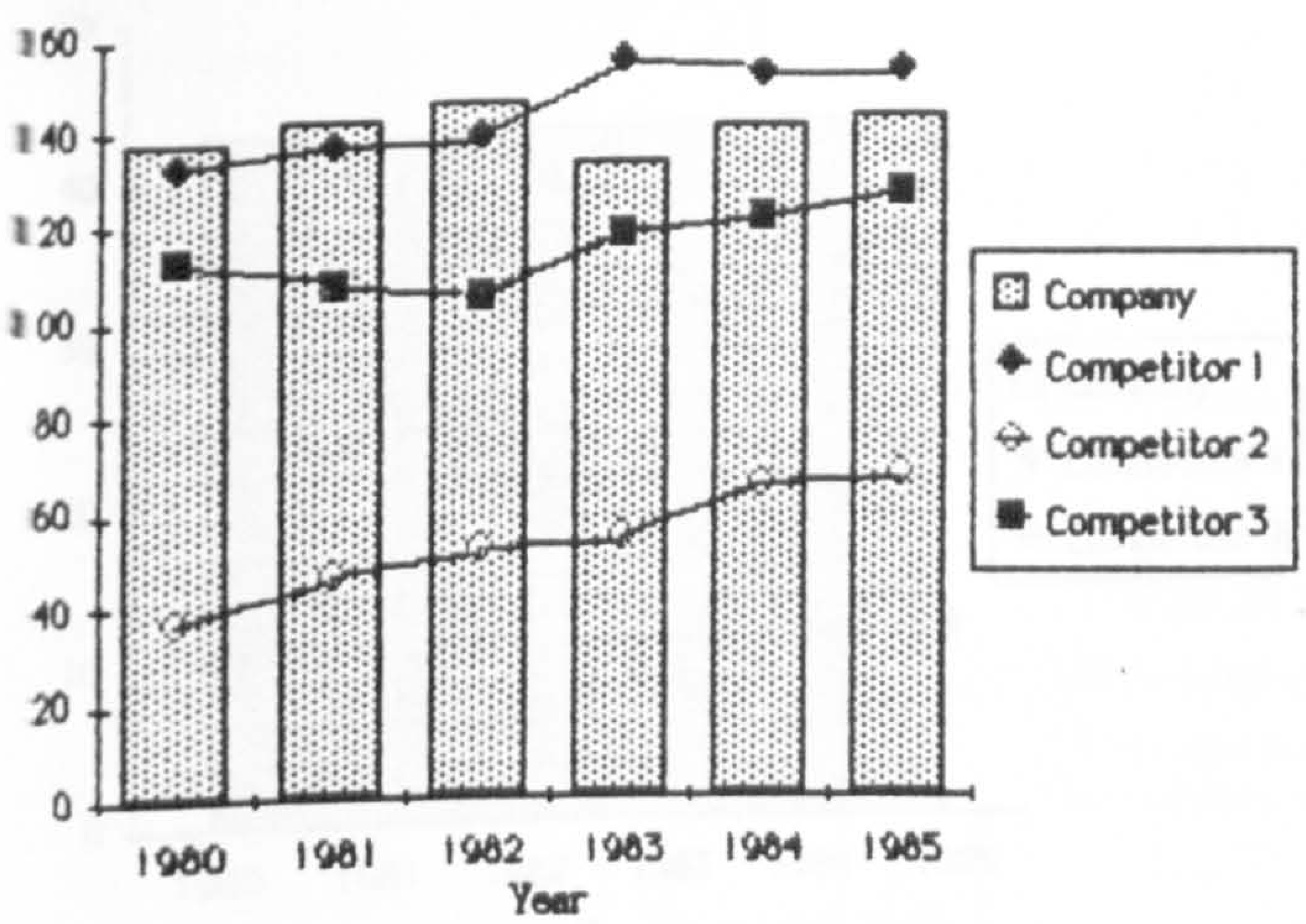
Expected Production Costs

Stable production	£60	per unit
+/- 10% fluctuation	£81	per unit
+/- 20% fluctuation	£79	per unit
+/- 30% fluctuation	£77	per unit
+/- 40% fluctuation	£79	per unit

Selling price= £83
Break even point= £78
Monthly storage cost per unit= £1.6
Number of units per year= 686,000

Product 7

Turnover



Expected Sales Performance

Percentage of Market

	January	Febuary	March	April
A	29	30	30	32
B	29	30	30	32

	May	June	July	August
A	33	33	34	35
B	33	32	29	28

	September	October	November	December
A	35	36	36	36
B	26	27	27	27

A % if products are freely availible.

B % if product supply is limited by production.

Expected Demand Fluctuations

Pounds (thousands)

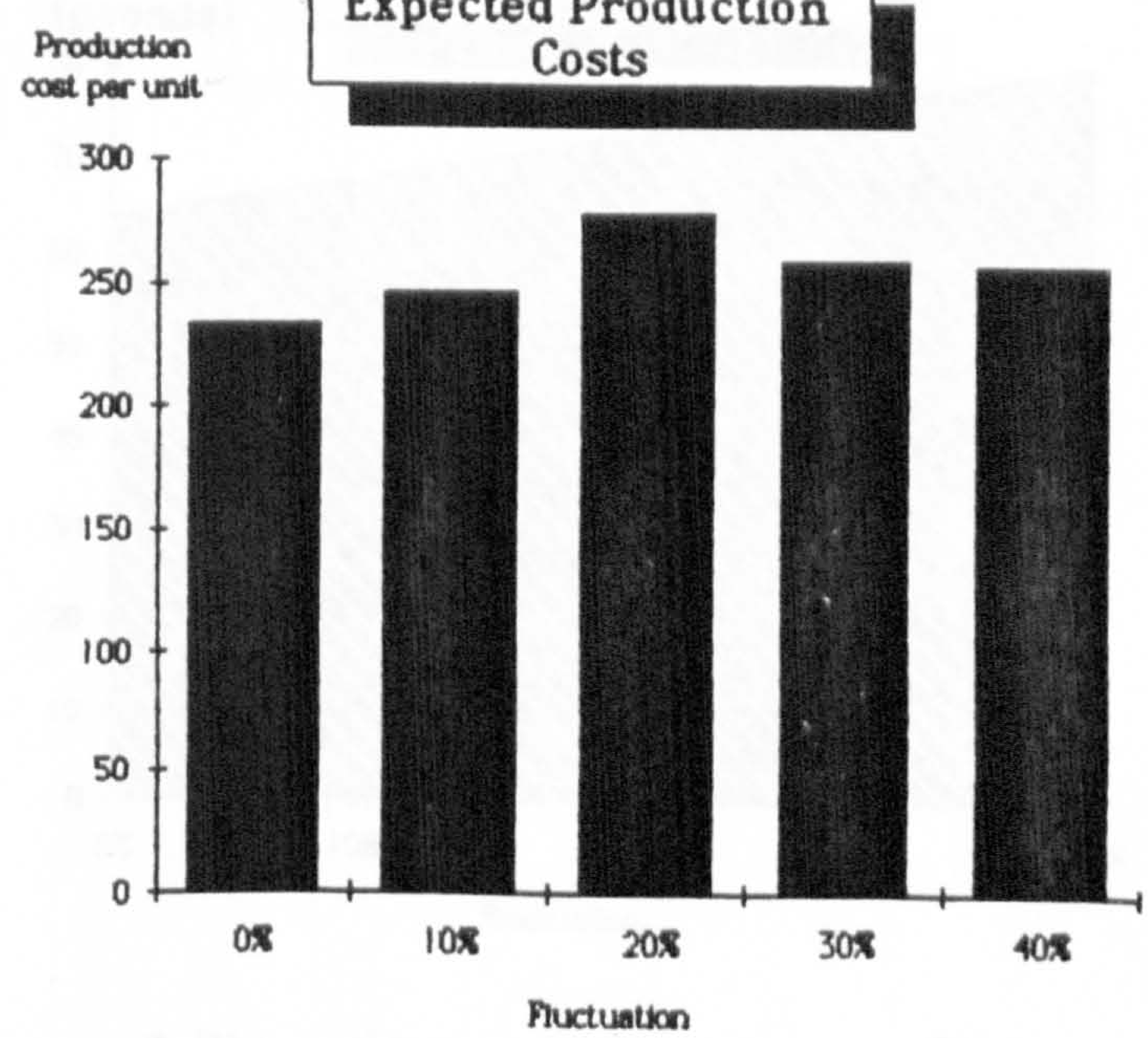
(In thousands)

January	Febuary	March	April
22.5	24.8	28	27

May	June	July	August
29.1	29.1	31.2	32

September	October	November	December
33	28	20.7	15.4

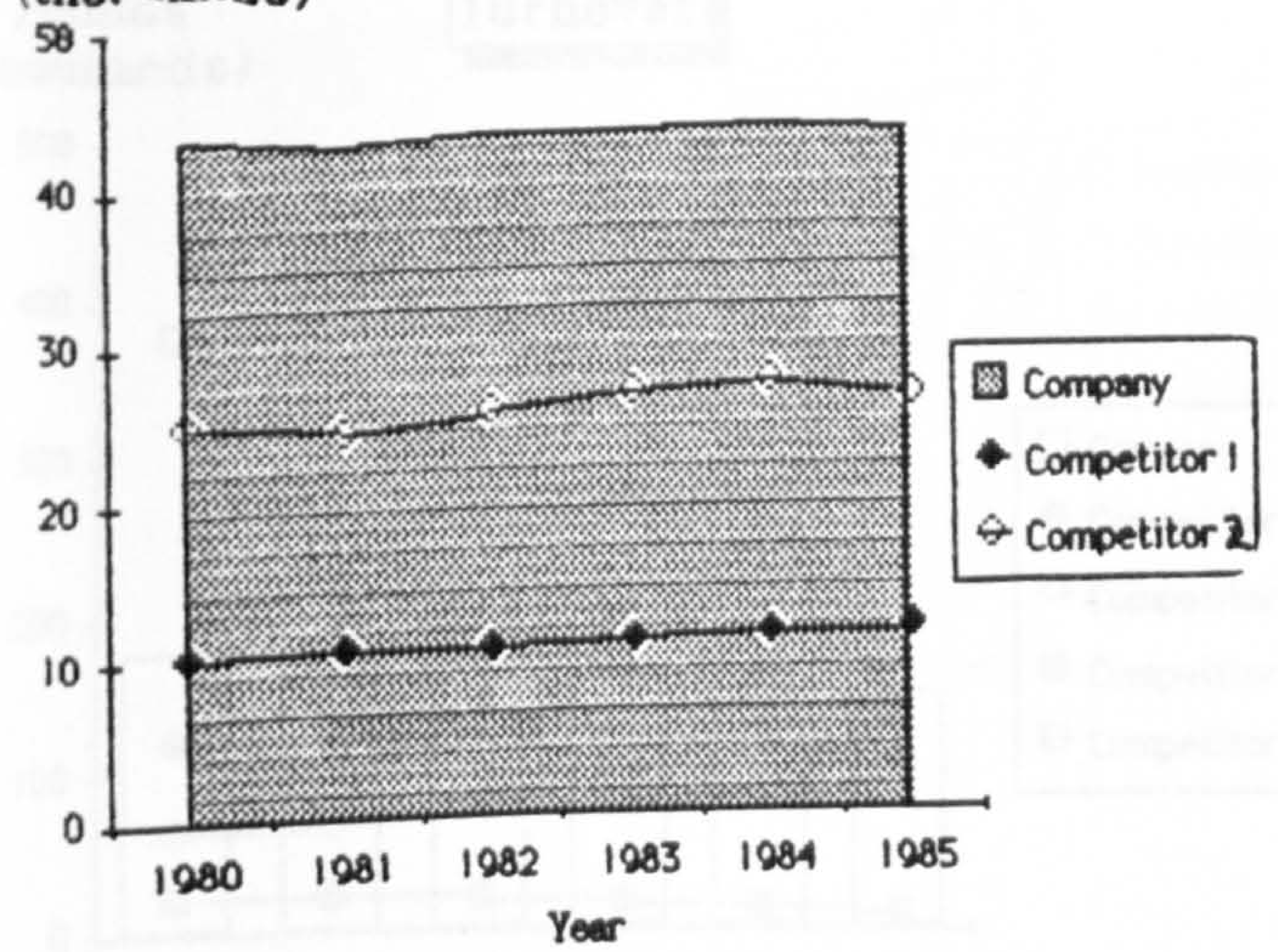
Expected Production Costs



Selling price-	£300
Break even point-	£267
Monthly storage cost per unit-	£3
Number of units per year-	320,000

Product 8

Turnover



Expected Sales Performance

Percentage of Market

	January	Febuary	March	April
A	62	62	62	55
B	62	62	62	55

	May	June	July	August
A	55	55	55	50
B	55	55	55	50

	September	October	November	December
A	47	41	41	41
B	45	38	32	27

A % if products are freely availible.

B % if product supply is limited by production.

Expected Demand Fluctuations

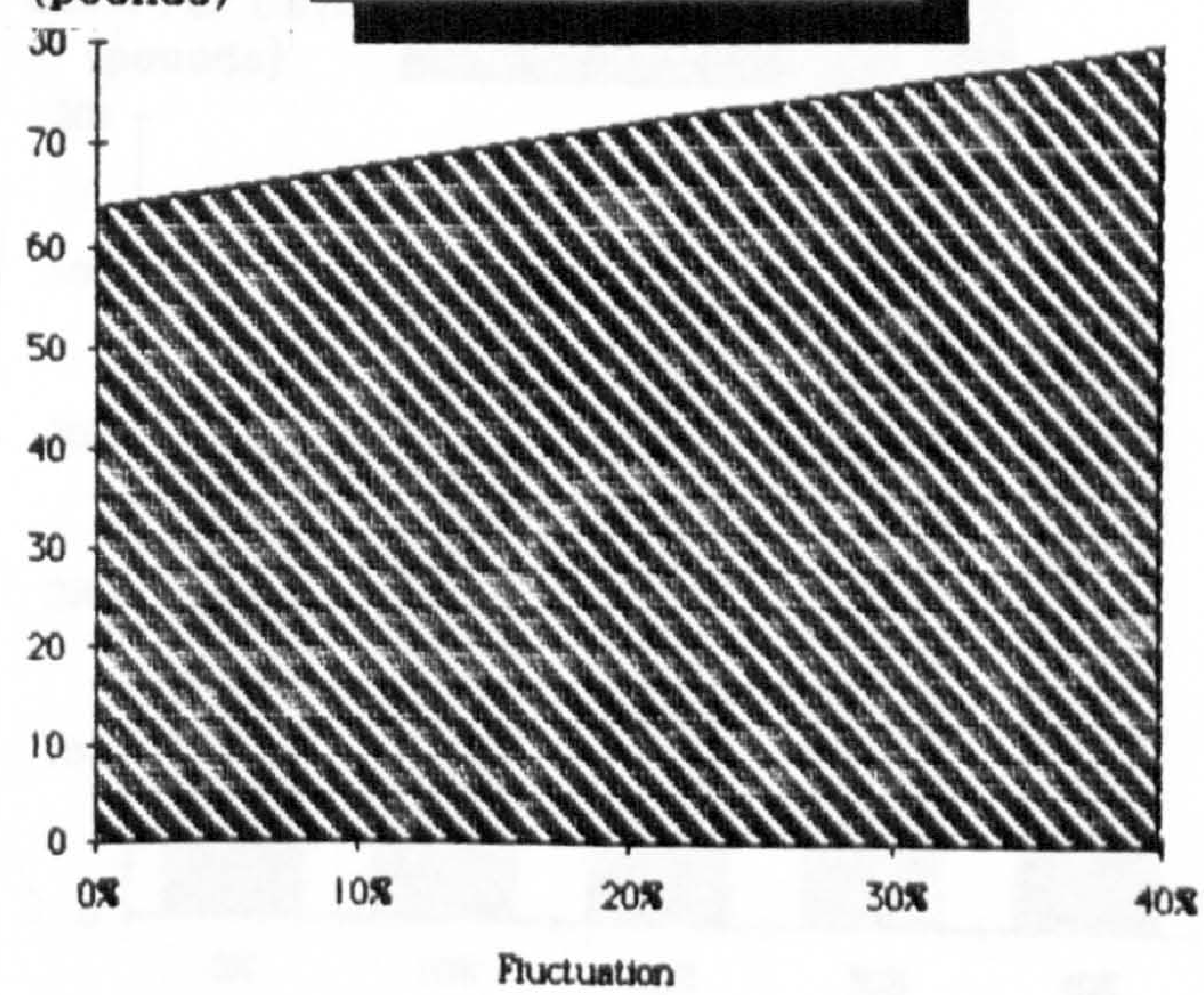
Pounds (thousands)

January	Febuary	March	April
6.4	6.6	6.6	6.6

May	June	July	August
6.2	6.4	6	6.4

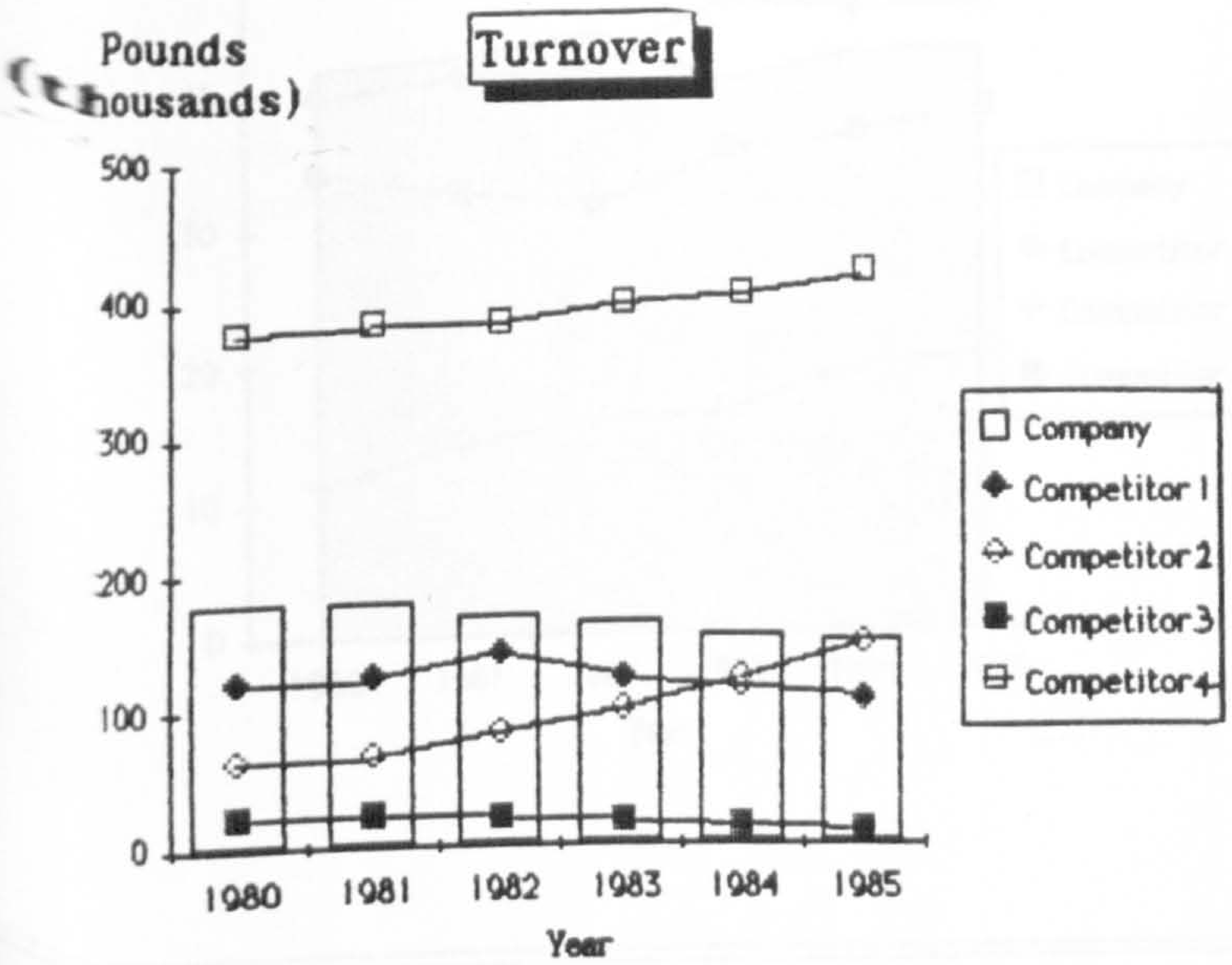
September	October	November	December
6.6	7.6	9.2	10.4

Expected Production Costs



Selling price=	£100
Break even point=	£84
Monthly storage cost per unit=	£0.4
Number of units per year=	85,000

Product 9



Expected Sales Performance

Percentage of Market

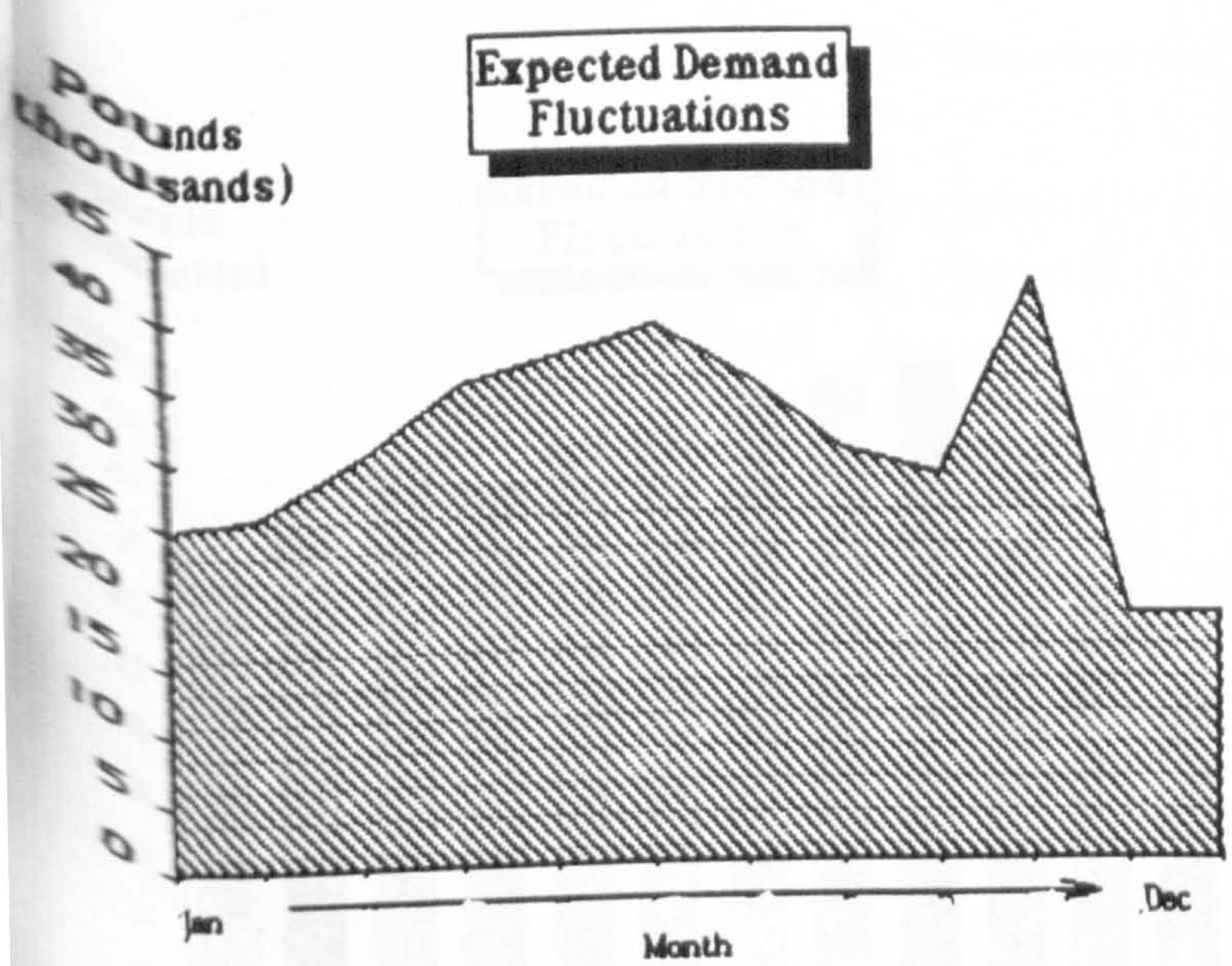
	January	Febuary	March	April
A	18	18	19	19
B	18	18	18	17

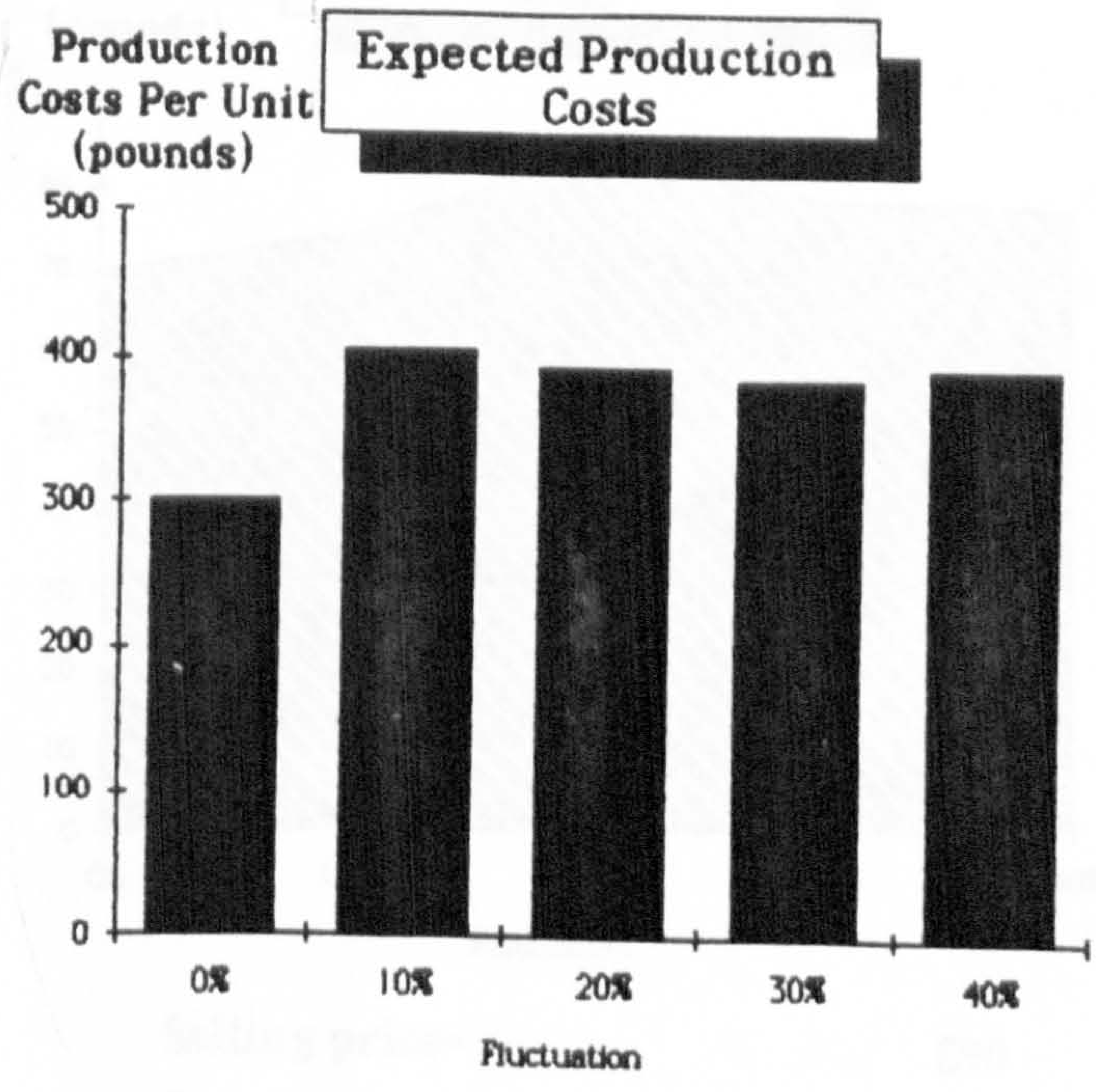
	May	June	July	August
A	19	19	19	19
B	16	16	16	17

	September	October	November	December
A	21	19	19	19
B	15	16	16	16

A % if products are freely availible.

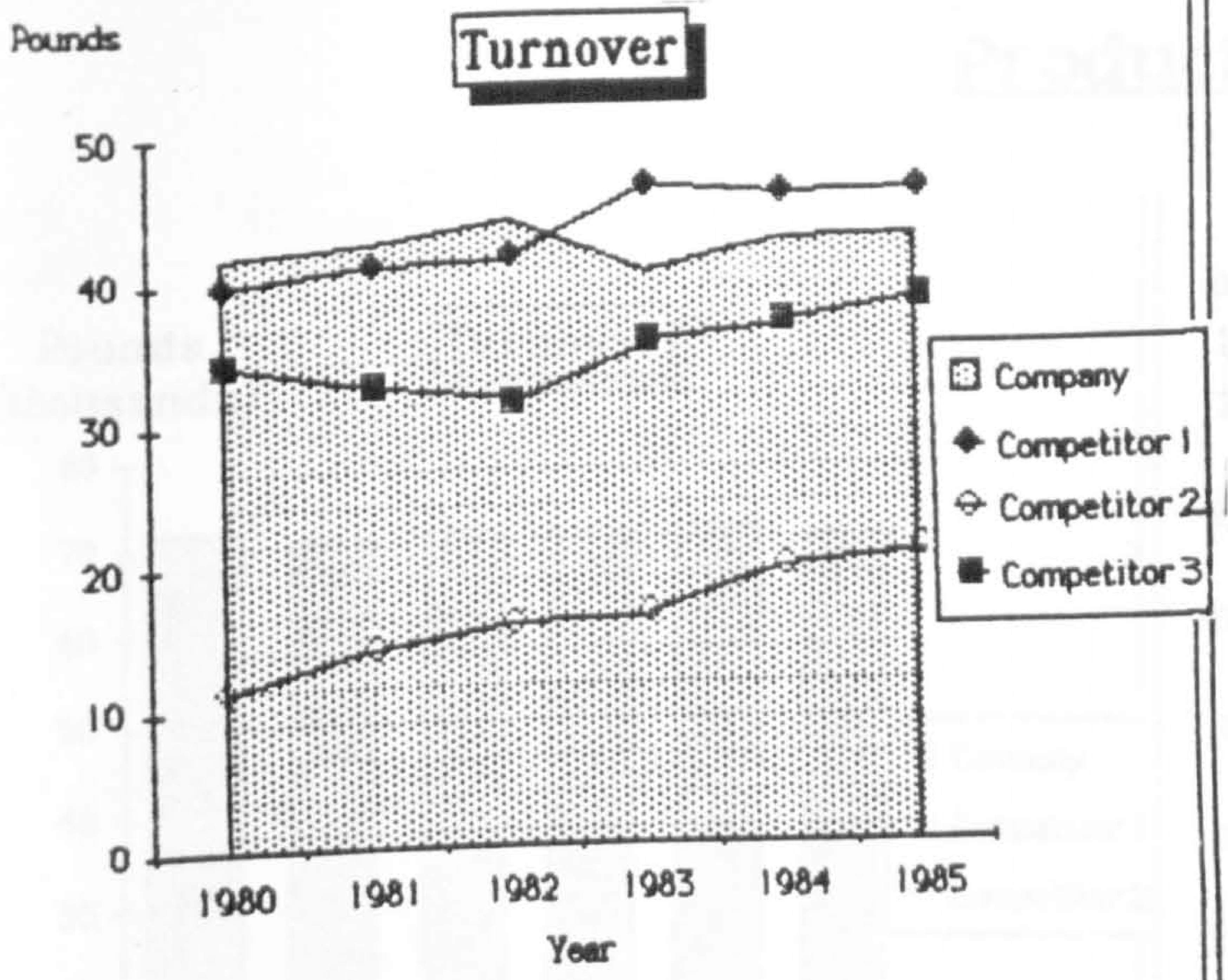
B % if product supply is limited by production.





Selling price=	£415
Break even point=	£390
Monthly storage cost per unit=	£8
Number of units per year=	343,000

Product 10



Expected Sales Performance

Percentage of Market

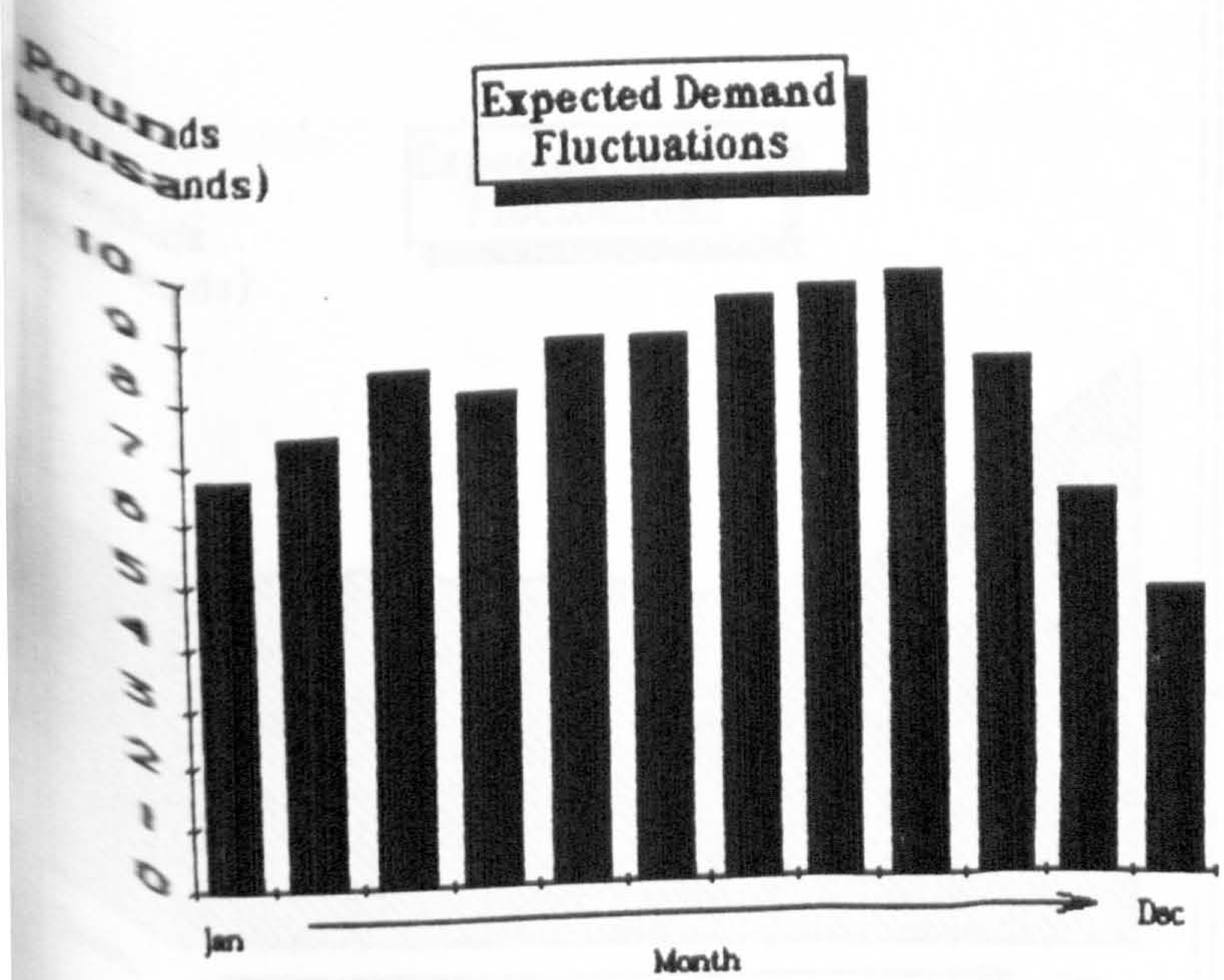
	January	Febuary	March	April
A	29	30	30	32
B	29	30	30	32

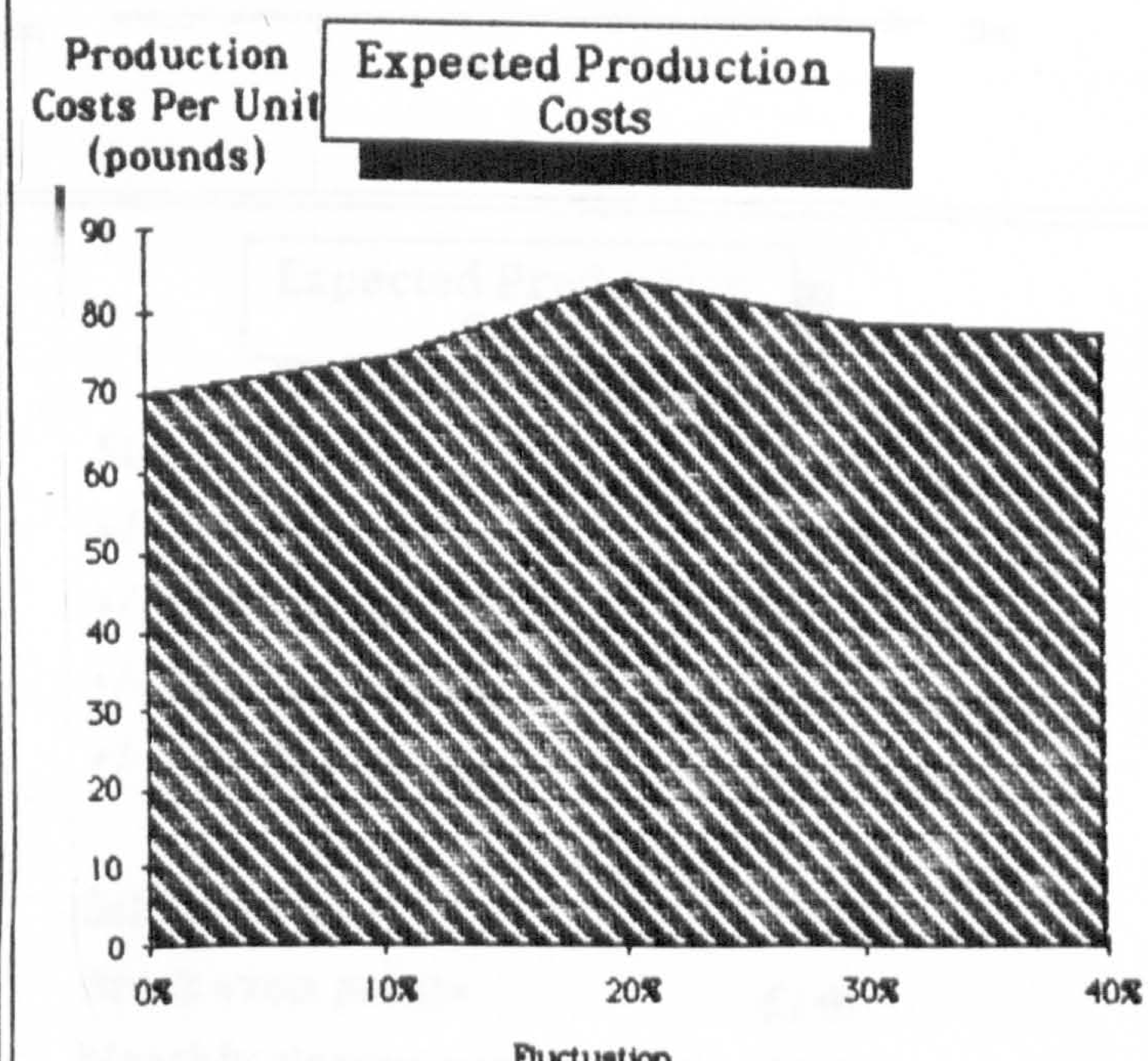
	May	June	July	August
A	33	33	34	35
B	33	32	29	28

	September	October	November	December
A	35	36	36	36
B	26	27	27	27

A % if products are freely availible.

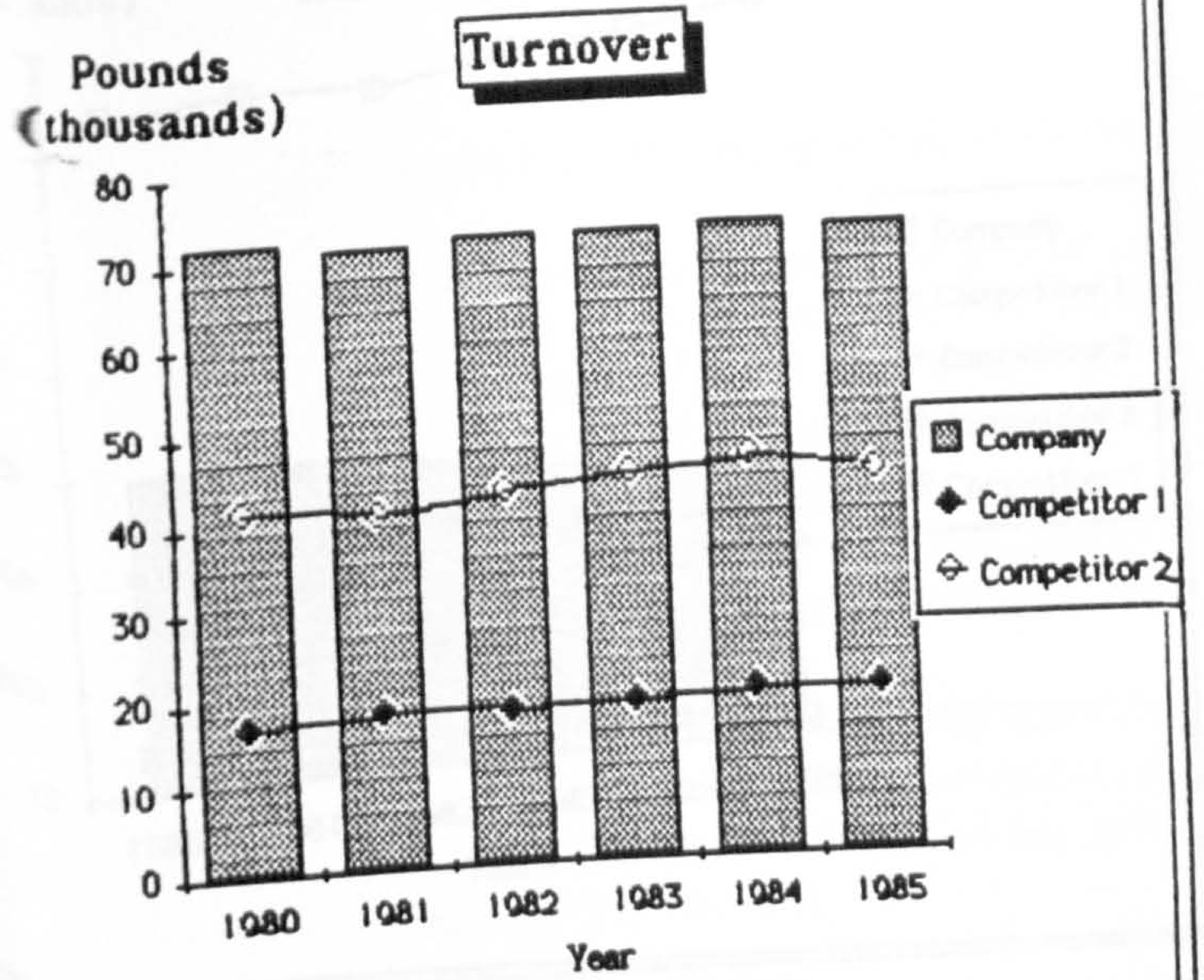
B % if product supply is limited by production.

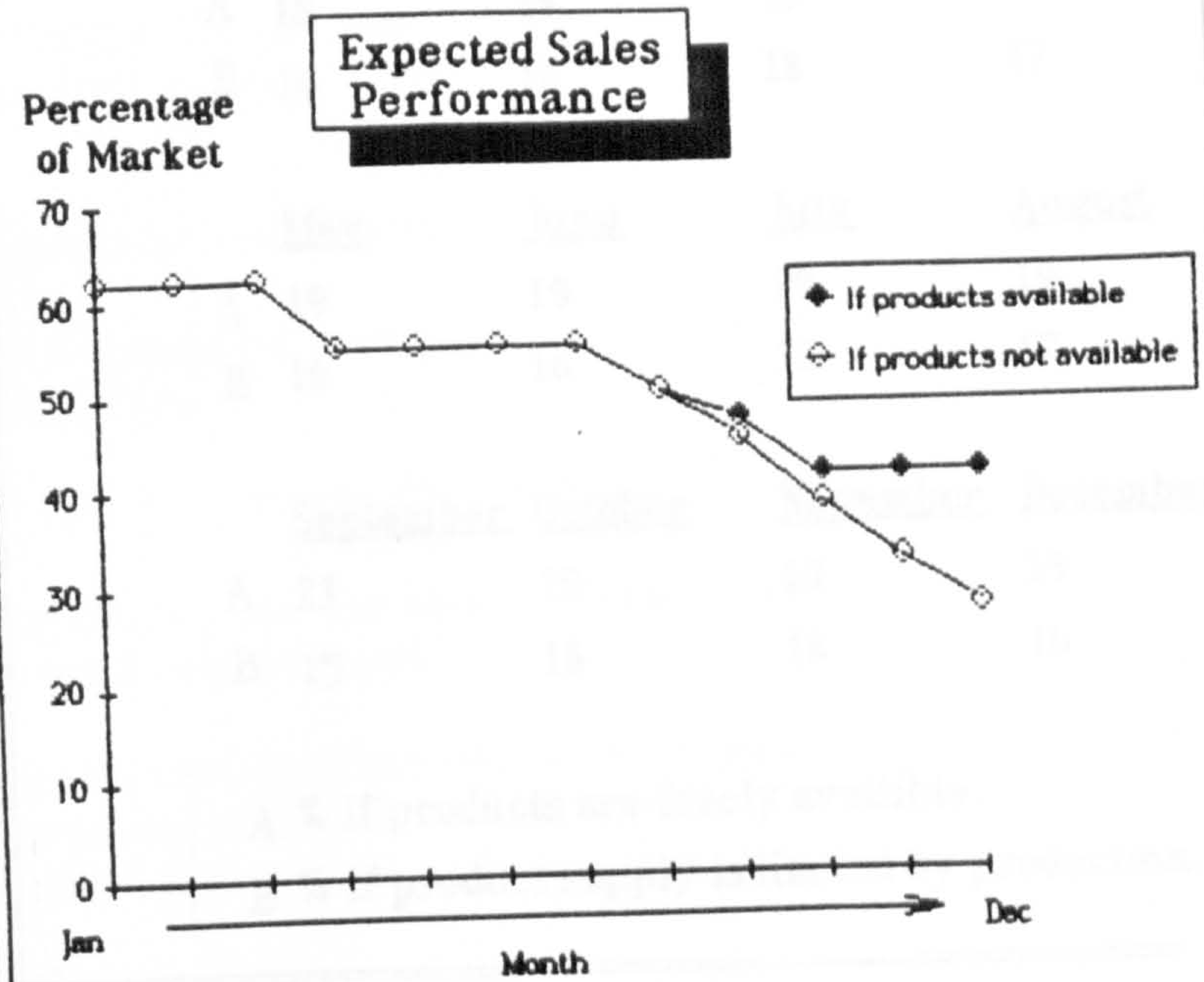


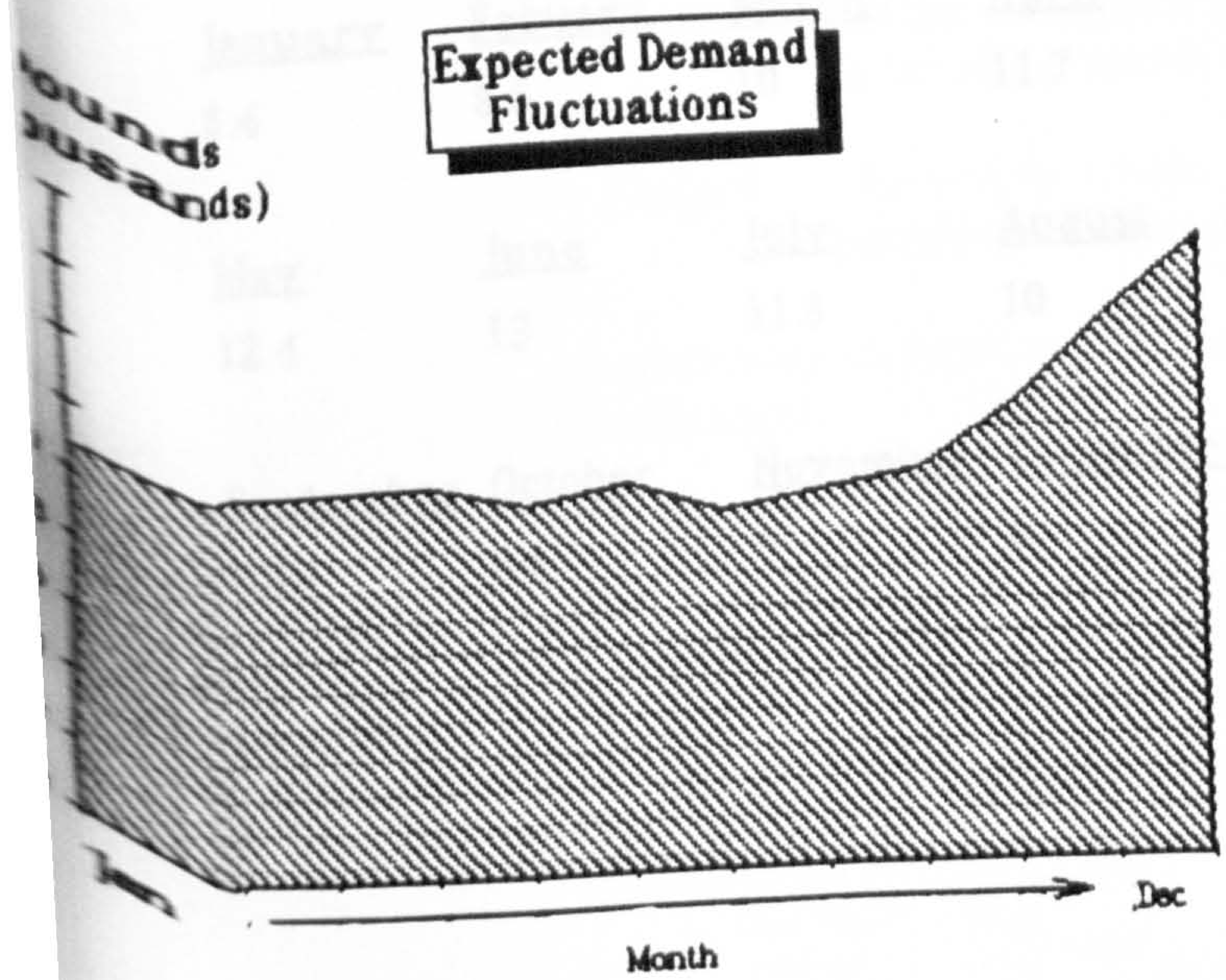


Selling price=	£90
Break even point=	£80
Monthly storage cost per unit=	£1
Number of units per year=	96,000

Product 11







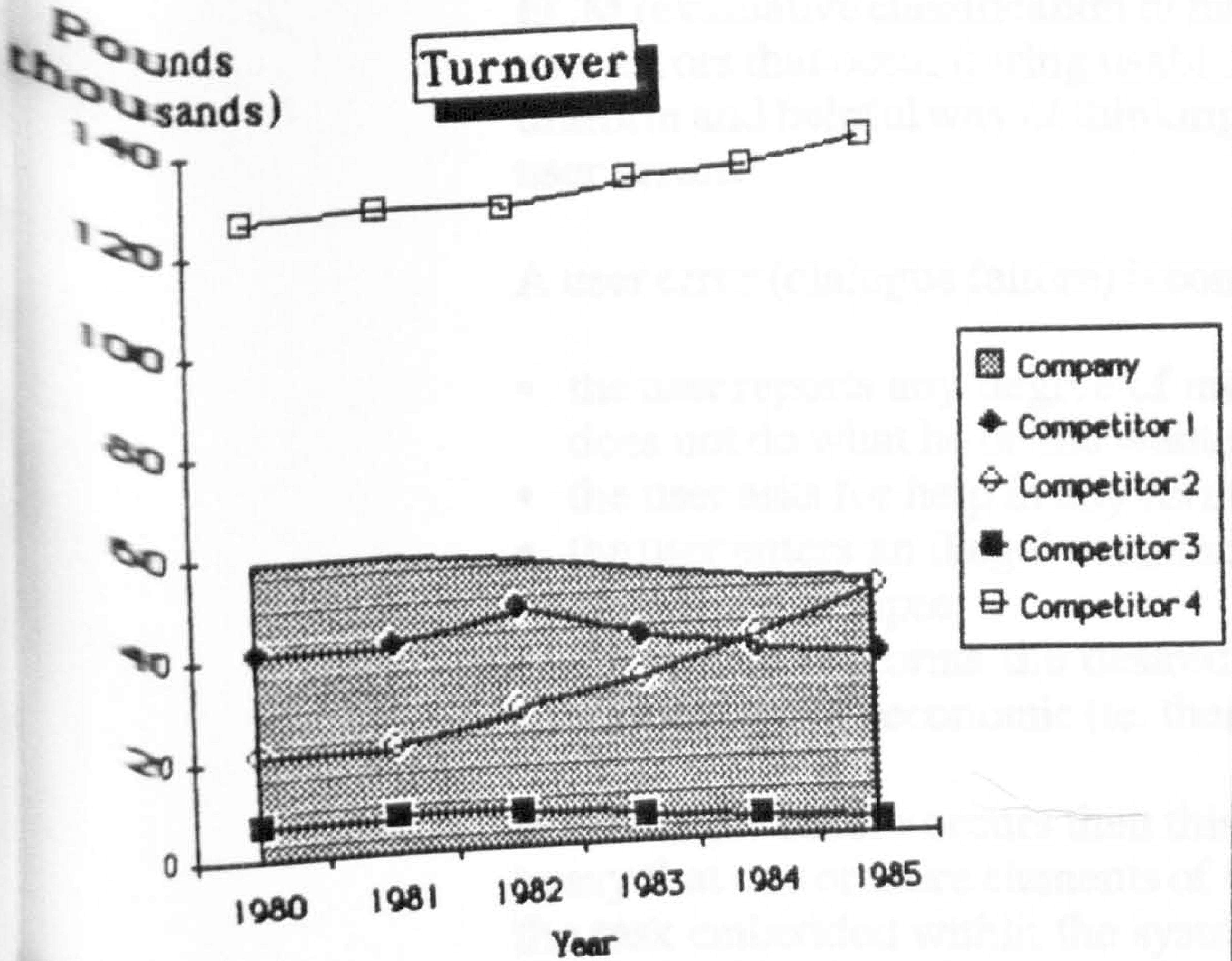
Expected Production Costs

Stable production	**£107**	**per unit**
+/- 10% fluctuation	**£113**	**per unit**
+/- 20% fluctuation	**£120**	**per unit**
+/- 30% fluctuation	**£127**	**per unit**
+/- 40% fluctuation	**£133**	**per unit**

Selling price= **£167**
Break even point= **£140**
Monthly storage cost per unit= **£0.7**
Number of units per year= **142,000**

Product 12

Turnover



Expected Sales Performance

Percentage of Market

	January	Febuary	March	April
A	18	18	19	19
B	18	18	18	17

	May	June	July	August
A	19	19	19	19
B	16	16	16	17

	September	October	November	December
A	21	19	19	19
B	15	16	16	16

A % if products are freely availible.

B % if product supply is limited by production.

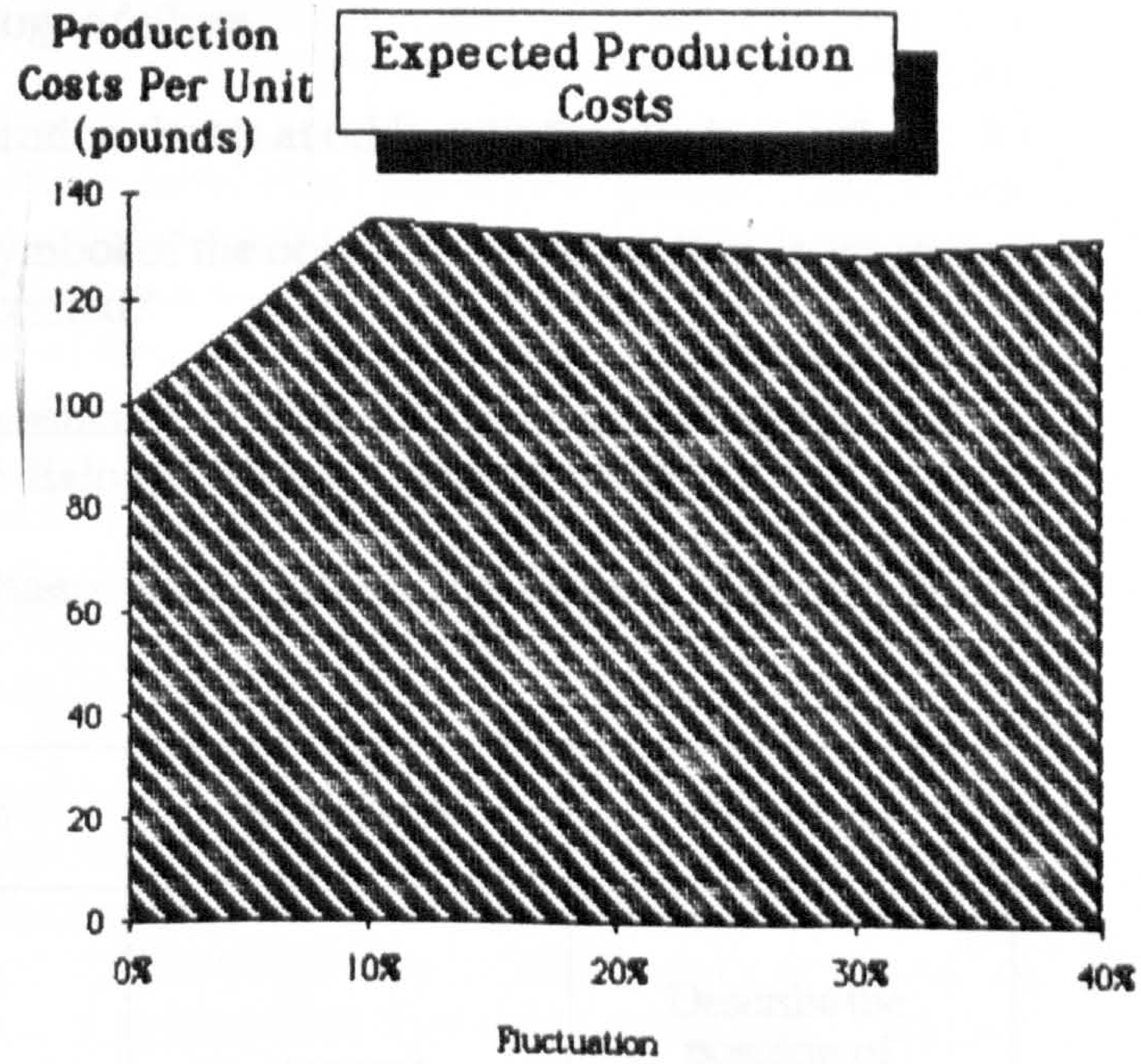
Expected Demand Fluctuations

January	Febuary	March	April
8.4	8.6	10	11.7

May	June	July	August
12.4	13	11.8	10

September	October	November	December
9.4	14	6	6

Expected Production Costs



Selling price=	£138
Break even point=	£130
Monthly storage cost per unit=	£2.7
Number of units per year=	114,000

Appendix 2:

A Short Summary of ECM.

ECM (evaluative classification of mismatch) is a technique for classifying the causes of user errors that occur during usability testing. The purpose of the scheme is to provide a uniform and helpful way of thinking and talking about the different elements (causes) of user errors.

A user error (dialogue failure) is considered to have occurred if:

- the user reports any degree of misunderstanding during the dialogue (ie. the system does not do what he or she wanted it to do),
- the user asks for help in any form,
- the user enters an illegal command that is not purely the result of a keystroke error, mental slip or lapse,
- the system performs the desired operation, but the series of commands the user adopted was uneconomic (ie. there is a shorter method).

If a dialogue failure occurs then this is said to be the result of a model mismatch. That is to say that one or more elements of the model of the task the user has and the model of the task embedded within the system do not match. The classification process works something like this;

Stage 1 Identification of the dialogue failure

Stage 2 Is it an object or an operation that is at odds with the user's view?

Stage 3 Is it the concept or the symbol of the object or operation that is wrong as far as the user is concerned?

Stage 4 The way in which the mismatched element (the concept or symbol) contributed towards the dialogue failure is then described.

Another way of representing the scheme is to consider the choices available during classification:

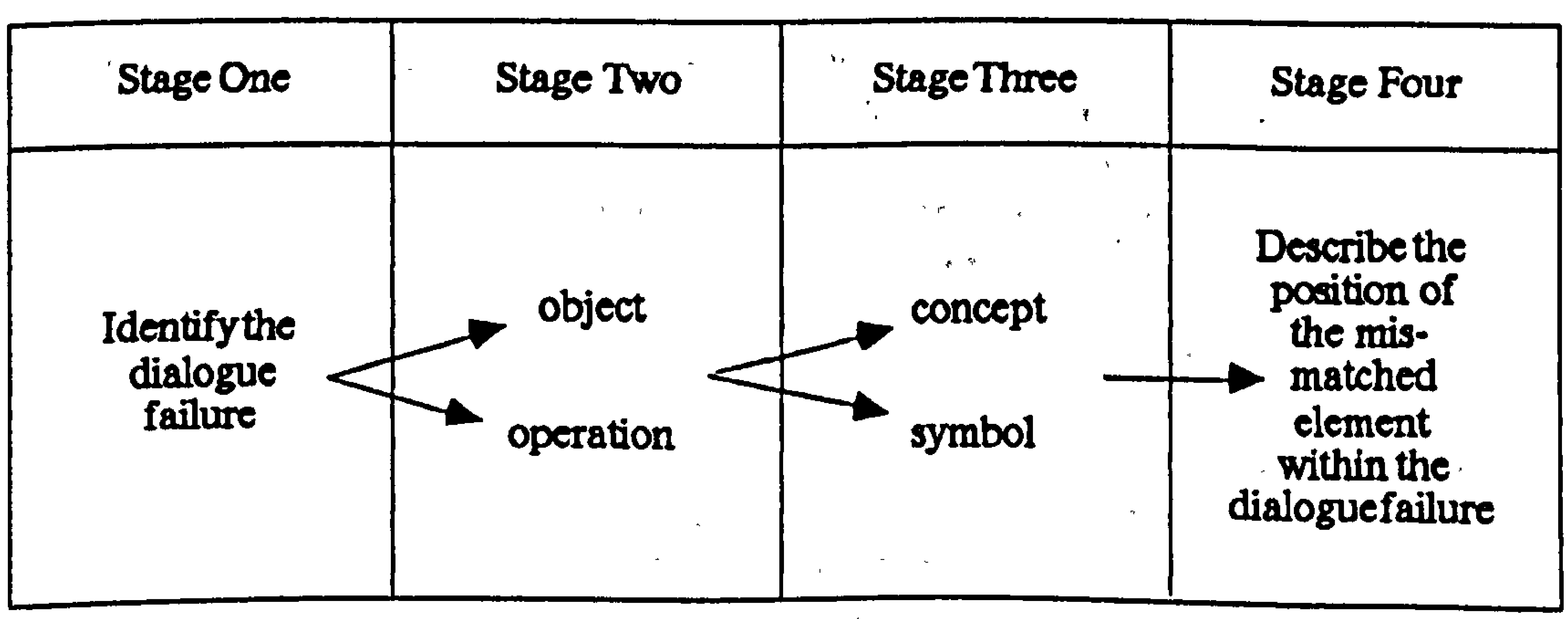


Figure 1; showing the choices made during the classification process.

More than one element may contribute towards a dialogue failure. In other words; a problem may have been caused by several contributing factors. As more problems are considered and their elements (causes) are classified this should enable a picture of mismatched elements to be built up.

The four classes of mismatch are as follows:

	Object	Operation
Concept	object-concept mismatch	operation-concept mismatch
Symbol	object-symbol mismatch	operation-symbol mismatch

Figure 2; showing the four classes into which a mismatched element should fit.

In more detail; these four classes of mismatch are as follows:

An object-concept mismatch: This is where an object may be in the wrong form as far as the user is concerned. The user may wish to use an object that the system does not possess or the object may be in the wrong context from the user's perspective. All of these are object-concept mismatches. ***For example;*** a user of a desktop system may want a filing cabinet as well as folders for different files. Yet if there is no filing cabinet within the desktop system then one of the elements (causes) of the problem would be an object-concept mismatch. Alternatively, the object may be unnecessary and unwanted. ***For example;*** an object the user did not wish to know about such as a print-file in a unix-like system (a configured file) might also involve an object-concept mismatch.

An object-symbol mismatch: This is where the symbol representing an object is not what the user either wants or expects. ***For example;*** in a library system users may look for a catalogue number. If however, the system refers to these as accession numbers then there may be an object-symbol mismatch; the object is correct, but the symbol (the term "accession" rather than "catalogue") is not what the user expects. Another example of an object-symbol mismatch can be found on the Hewlett Packard 'Memomaker' word processing package, where text which will be underlined when printed is displayed on screen in italics with no underline creating confusion amongst first-time users.

An operation-concept mismatch: This is where an operation is not what the user either wants or expects. In other words; the operation does not perform a series of actions in the way the user requires. ***For example;*** a user may want to merge two files. If no such operation exists then this may create an operation-concept mismatch. Alternatively, the operation might be acceptable to the user, but may occur in the wrong context. ***For example;*** there are systems where the line delete operation deletes the line below the cursor rather than the line the cursor is actually on. The user might consider the operation to have been correct, but in the wrong context. Again this is an example of an operation-concept mismatch.

An operation-symbol mismatch: This is where the symbol representing an operation is not what the user would expect or want. ***For example;*** a user might want to print a file. If the command for printing a file is "nroff -ms *filename* > *printfilename*" followed by "lpc *printfilename*" rather than "print" then this may create, amongst other things, an operation-symbol mismatch. Alternatively, a symbol associated with an operation may be in the wrong context as far as the user is concerned. ***For example;*** a command name in the wrong menu would be classed as an operation-symbol mismatch.

The process of classification need not be complicated. Here is an example of the classification process:

Stage 1 Identification of the dialogue failure: Fred reports problems while trying to print a file.

Stage 2 Is it an object or an operation that is at odds with the user's view? Fred reports that he does not know the command to print and that he cannot see a command on the screen that might possibly cause his file to be printed. As we can see that Fred's file looks OK then there appears to be a problem with the ***operation*** to print a file.

Stage 3 Is it the concept or the symbol of the object or operation that is wrong as far as the user is concerned? As we know that the command "spool" will cause a file to be printed then it looks as if the concept of the operation fits in with what Fred wants. Yet the symbol "spool" does not appear to be what Fred expected. Therefore, this is a ***symbol*** mismatch.

Stage 4 The way in which the mismatched element (the concept or symbol) contributed towards the dialogue failure is then described. Fred did not appear to associate the symbol "spool" with the concept of printing his file. In this example there appears to be only one mismatched element (the operation-symbol mismatch) which has contributed towards the dialogue failure.

The example above shows the questions that might be asked during the classification process. These questions were taken from figure one shown on page two. The mismatch in this example is of the operation-symbol type; where an operation (the operation to print) has the wrong or unexpected symbol as far as the user is concerned.

Appendix 3:

The First Study of ECM: The instructions.

Part A.

1 Type a brief letter to inform Mr R. P. Jones of Holbeck House, 14 Able Walk, Branshome, York, that you will not be able to attend the proposed meeting which was provisionally set for 9 am, 21st April 1987. Give your own address as: Department of Mathematics and Computer Studies, The Polytechnic, Huddersfield, HD1 3DH.

Part B.

2 Replace the words "spending plan" in the second paragraph with the word "budget".

3 Insert the following paragraph at the beginning of the text:

"An ebullient President Reagan was last night said to be sitting up in bed reading briefing papers at the Bethesda Naval Hospital after a painful but apparently successful prostate operation early yesterday. The initial verdict from the doctors was that no cancer had been found in the prostate tissue or the four small growths removed from his colon on Sunday".

4 Adjust the text so that the third and fourth paragraphs are joined to form one large paragraph.

5 Underline the words "Guardian News Report".

6 Change the margins on the text so that the text is only four inches wide.

7 Rejustify the whole text.

8 Replace the word "traditional" in the third paragraph with "ritual".

9 Centre the heading "Guardian News Report".

10 Remove the words "and demoralised" in the third paragraph.

11 Move the heading "Guardian News Report" back to the left.

12 Move the second paragraph to the end of the text.

Appendix 4:

The First Study of ECM: The Results.

A list of all of the mismatches that occured is presented. Following this, some tables are presented which show the different mismatches, grouped around system concepts, and the number of subjects that experienced these problems.

1 User expects backspace to delete, not just move the cursor backwards (operation-concept).

2 Position of the margin is not clearly marked (object-symbol).

3 Position of right margin too far to the left immediately after loading Memomaker and beginning session (object-concept).

4 User expects text to wrap-back after deletion of text (operation-concept).

5 User could not change the text outside of the margin (operation-concept).

6 In help; User expected more help when command name was selected. Instead the system only highlighted that part of the text on the screen relating to the command and the user had already read this (operation-concept).

7 User expected the text to automatically re-justify (operation-concept).

8 User tried to use 'delete character' to join the paragraphs together (operation-concept).

9 Underline operation causes text to appear in italics on screen rather than underlined (object-symbol).

10 Not clear what 'define block' means when the system means select (operation-symbol).

11 The re-justifying operation joins all of the paragraphs together (operation-concept).

12 User though align command would move the heading back to the left (operation-symbol).

13 User thought paste would move heading back to the left (operation-symbol).

14 User thought 'cut' operation would destroy text rather than save it to be pasted later (operation-symbol).

15 User didn't expect wrap-round facility (operation-symbol).

16 The 'line insert' command not inserting the line in the position the user expected (operation-concept).

17 User expected return to create a new line and move text below down (operation-concept).

18 User unaware of underline facility (operation-symbol).

19 The term 'define block' not understood by the user (operation-symbol).

20 User expected cursor to wrap round when being moved (operation-concept).

21 User expected to find a 'move text to the left' command (operation-concept).

22 User thought 'DEL' button would delete text (operation-symbol).

23 User expected the 'del char' to delete text behind cursor not in front of it (operation-concept).

24 User expected return to cause indentation in text (operation-concept).

25 User did not expect margin keys to be in the format menu (operation-symbol).

26 User did not expect the underline operation to be under the block keys menu (operation-symbol).

27 User unaware of 'overwrite' facility (operation-symbol).

28 User unaware of 'center line' facility (operation-symbol).

29 User expected, when selecting a block, to move the cursor, mark the block and then select the block - the system however begins it's selection straight away. (operation-concept).

30 User thought that 'paste' was a mode and not a command (operation-symbol).

31 User did not expect new line after wrap-round as return does not create a line and move text down the page (operation-concept).

32 User did not expect the 'align' command to be in the 'block' menu (operation-symbol).

33 User thought 'center line' command should be in one of the menus (operation-symbol).

34 User did not expect a blank line to be left after cutting a paragraph (object-symbol).

35 The 'beep' the system creates when the cursor approaches the end of the line means nothing and caused the user to believe that something was wrong (operation-concept).

36 User unaware that 'insert char' was a mode and not a command for a single character or word (operation-symbol).

37 User did not expect 'underline' command to be part of the 'enhance block' sub-menu (operation-symbol).

38 User thought that the 'set tab' operation would re-set the margin (operation-symbol).

39 User thought that 'justify' was a command not an option (operation-symbol).

40 User did not realise that text would be pasted from where the cursor was within the text - the relationship between the cursor and pasting is unclear to the user (operation-symbol).

41 User expressed a wish for a 'spelling checker' facility (operation-concept).

42 User expressed a wish for a 'search' facility (operation-concept).

43 User expected the 'insert' mode selection to be under one of the menus (operation-symbol).

44 The 'delete line' operation did not operate on the line the user expected (operation-concept).

45 User expected the 'backspace' key to join the paragraphs together (operation-concept).

46 User attempted to use the 'copy' operation to paste text onto the end of a paragraph as a way of joining them (operation-concept).

47 User unaware of delete facility (operation-symbol).

48 User did not realise that blank areas were counted as text when using the 'align' operation(operation-concept).

49 User thought that, before beginning typing, a file must be created (operation-concept).

50 User did not realise that 'CAPS' key is an on and off mode key and does not have to be pressed for each letter wanted in capitals (operation-symbol).

51 The 'caps' mode indicator not obvious (operation-symbol).

52 User unaware that it was possible to alter the margins (operation-concept).

53 The system, in accessing one of the discs, caused the user to believe that he had made a mistake (operation-concept).

54 User did not know how to scroll the text to find the paragraph they wanted (operation-symbol).

55 User did not realise that the cursor had to be placed on the line that was to be centered when using the 'center line' command (operation-concept).

56 When trying to move a heading back to the left of the page the user selected the 'align' command, but did not realise that the blank spaces between the margin and the heading needed to be selected as well as the heading itself (operation-concept).

57 User did not expect the 'align' command to move the heading away from the center (operation-concept).

58 The relationship between setting the margin and the cursor not clear to the user (operation-concept).

59 User wanted a 'join' command to join paragraphs (operation-concept).

60 User confused by the 'Memomaker main' command. User wanted to return to the main menu but did not realise that this was the command (operation-symbol).

61 User expected the 'cut' text operation to join paragraphs together (operation-concept).

62 User expected the text, when re-aligned, to line up with both margins. The user did not want a 'ragged' effect at the right margin (operation-symbol).

63 User expected the word set at the tab to move as the tab was moved (operation-concept).

64 User expected the 'delete line' operation to delete the text from the cursor to the end of the line, not all of the text on the line (operation-concept).

65 User expected an 'underline' key (operation-concept).

66 User thought 'center line' was a mode not an operation (operation-concept).

67 User expected 'center line' to move text back to the left as well as to the center (operation-concept).

68 User expressed a wish to be able to move directly from one menu to the next without having to go through the main menu every time (operation-concept).

69 User thought align was the underline command (operation-symbol).

70 User expected to have to apply the underline operation to the text under the title rather than to the text itself (operation-concept).

71 User thought the 'set tab' command would re-justify the text (operation-symbol).

72 User thought all text would insert and was not aware that insert was an on-off mode(operation-concept).

73 User thought 'insert line' was a mode rather than a command (operation-symbol).

74 The user tried to use 'delete line' to join paragraphs together (operation-concept).

75 User tried to use 'copy' and 'paste' to move text to the center of the line (operation-concept).

List of mismatches grouped in concepts

		Subject												Total
		1st Year Students					Staff					Others		
	mismatch	1	2	3	4	5	6	7	8	9	10	11	12	
Insert	36					2	1		1	1				4
	16		1											1
	72												1	1
	27			1										1
	73										1			1
Cut, copy & paste	30			1										1
	14		1				1	1			1	1		5
	40					1		1						2
	75										1			1
Facilities	68											1		1
	41						1							1
	42									1	2		1	3
	Help 6	1										1		1
	Tab 63									1				1
	Bleep 35					1	1	1			1			4
	Starting 49							1	1					2
	Disc Assess 53							1						1
Caps	50							1				1		2
	51							1						1

		Subject												Total
		1st Year Students					Staff					Others		
	mismatch	1	2	3	4	5	6	7	8	9	10	11	12	
Deletion	47						1							1
	1	1	2	1	1	1	1	1		1	1		1	10
	22			1										1
	23			2			1			1	2			4
	44						1							1
	64									1				1
Wrap	15		1	1	1	1		1	1			1	1	8
	20		1	1			2		1					4
	4	1	2	1	2	2			1	3			1	8
	31				1				1					2
Underline	18		1			1								2
	65									1				1
	70											1		1
	69											1		1
	9	1	1	1	1	1	1	1	1	1	1	1	1	12
Return	17		1	1	2				1	1			1	6
	24			1										1
	Scrolling 54							1						1

		Subject												
		1st Year Students					Staff					Others		Total
	mismatch	1	2	3	4	5	6	7	8	9	10	11	12	
Margins	52							1						1
	58							1			1			2
	2	2	1	1	1	1	1	2	1	1	2	1	2	12
	3	1	1	1	1	1	1	1	1	1	1	1	1	12
	38					1					1			2
	5		2	1	2	1		1	1	1	1	1		9
Center line	28			1		1								2
	55							1			1			2
	66									1				1
	67									1				1
	13	1												1
	12	1												1
	21		1							1		1	1	4
Joining paragraphs	59								1					1
	8	1				1	1		1		1		1	6
	45						1							1
	46						1							1
	61								1					1
	74										1			1

		Subject												Total
		1st Year Students					Staff					Others		
	mismatch	1	2	3	4	5	6	7	8	9	10	11	12	
Selecting text	10	1	1	1				1						4
	19	1	1	1										3
	29			1							1			2
	34				1		1		1					3
Justify/Align	7	2	2	1		2		4	1	2	2	1	5	10
	48 & 56						1	1						2
	62								1					1
	71											1		1
	39					1	1	1			1			4
	11	1	1	1	1	1	1	1	1	1	1	1	1	10
	57							1			1			2
Menu	26			1		1	1	1	1					5
	32				1	1		1						3
	33				1	1	1				1	1	1	6
	37					1								1
	43						1						1	2
	25			1				1		1		1	1	5
	60								1					1

Appendix 5:

The Second Study of ECM: The Classification Questionnaire.

ECM: USABILITY STUDY.

The purpose of this document is to allow you (the subject) to try out a new technique for classifying different elements of the problems users experience with different systems. Your responses and comments will allow us to assess the classification scheme. Furthermore, your opinions will allow the Human Factors people at Hewlett Packard to decide whether or not the scheme is likely to be of use within the company.

Some of the problems that users experienced during the Newwave mail usability testing have been described and you are being asked to try to classify the various elements of these problems using the scheme. ***This is not a test of your ability, but a test of the usability and usefulness of the classification scheme.***

There are three parts to this assessment of ECM (evaluative classification of mismatch):

1 Firstly, you are asked to read the summary of ECM (the classification scheme) and you should try to classify all of the problems described using the scheme.

2 Secondly, once you have classified all of the problems you should get together with some of your colleagues who have also classified the problems and discuss any of the classifications you have not agreed upon. This will be explained in more detail later.

3 Thirdly, you will be asked to take part in a short interview where you will be asked about the classification technique, particularly about how easy or difficult it is to use and whether you believe it is likely to be of any use.

Before you start could you please write your name and your position within the company in the space provided.

NAME ____________________________________

POSITION ____________________________________

A short summary of ECM.

ECM (evaluative classification of mismatch) is a technique for classifying the causes of user errors that occur during usability testing. The purpose of the scheme is to provide a uniform and helpful way of thinking and talking about the different elements (causes) of user errors.

A user error (dialogue failure) is considered to have occurred if:

- the user reports any degree of misunderstanding during the dialogue (ie. the system does not do what he or she wanted it to do),
- the user asks for help in any form,
- the user enters an illegal command that is not purely the result of a keystroke error, mental slip or lapse,
- the system performs the desired operation, but the series of commands the user adopted was uneconomic (ie. there is a shorter method).

If a dialogue failure occurs then this is said to be the result of a model mismatch. That is to say that one or more elements of the model of the task the user has and the model of the task embedded within the system do not match. The classification process works something like this;

Stage 1 Identification of the dialogue failure

Stage 2 Is it an object or an operation that is at odds with the user's view?

Stage 3 Is it the concept or the symbol of the object or operation that is wrong as far as the user is concerned?

Stage 4 The way in which the mismatched element (the concept or symbol) contributed towards the dialogue failure is then described.

Another way of representing the scheme is to consider the choices available during classification:

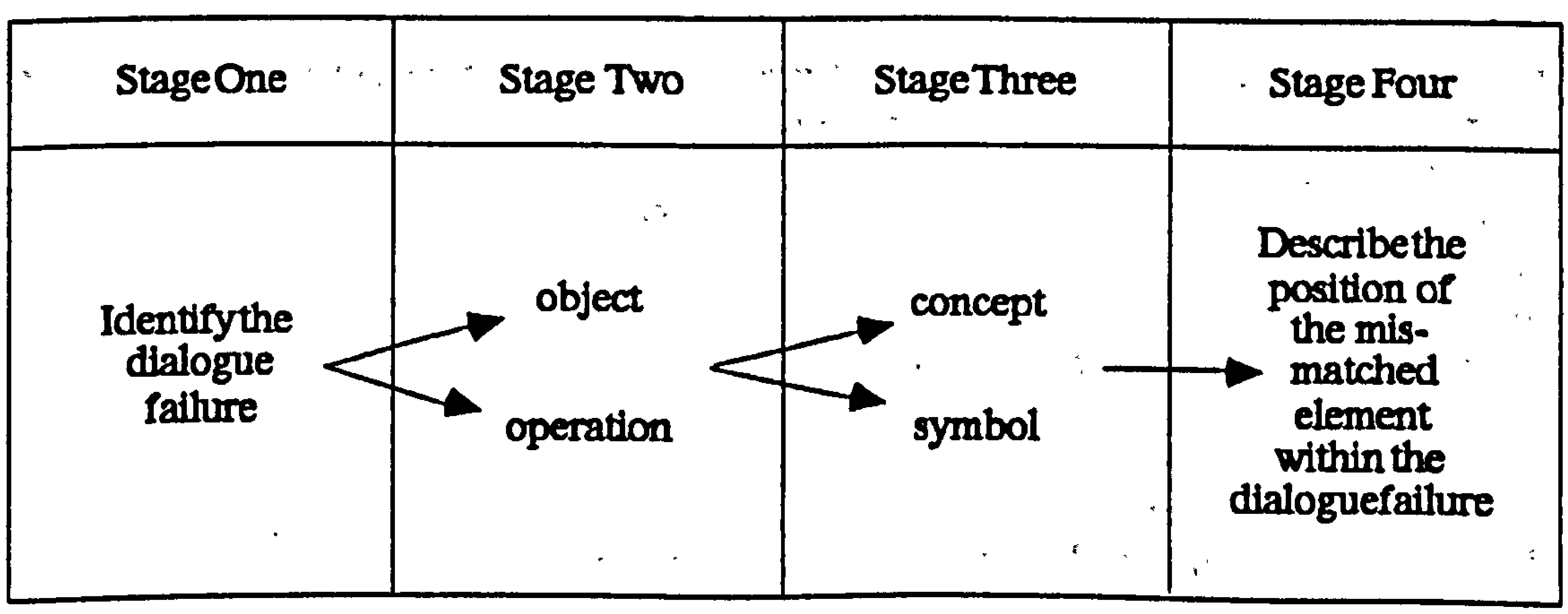


Figure 1; showing the choices made during the classification process.

More than one element may contribute towards a dialogue failure. In other words; a problem may have been caused by several contributing factors. As more problems are considered and their elements (causes) are classified this should enable a picture of mismatched elements to be built up.

The four classes of mismatch are as follows:

	Object	Operation
Concept	object-concept mismatch	operation-concept mismatch
Symbol	object-symbol mismatch	operation-symbol mismatch

Figure 2; showing the four classes into which a mismatched element should fit.

In more detail; these four classes of mismatch are as follows:

An object-concept mismatch: This is where an object may be in the wrong form as far as the user is concerned. The user may wish to use an object that the system does not possess or the object may be in the wrong context from the user's perspective. All of these are object-concept mismatches. ***For example;*** a user of a desktop system may want a filing cabinet as well as folders for different files. Yet if there is no filing cabinet within the desktop system then one of the elements (causes) of the problem would be an object-concept mismatch. Alternatively, the object may be unnecessary and unwanted. ***For example;*** an object the user did not wish to know about such as a print-file in a unix-like system (a configured file) might also involve an object-concept mismatch.

An object-symbol mismatch: This is where the symbol representing an object is not what the user either wants or expects. ***For example;*** in a library system users may look for a catalogue number. If however, the system refers to these as accession numbers then there may be an object-symbol mismatch; the object is correct, but the symbol (the term "accession" rather than "catalogue") is not what the user expects. Another example of an object-symbol mismatch can be found on the Hewlett Packard 'Memomaker' word processing package, where text which will be underlined when printed is displayed on screen in italics with no underline creating confusion amongst first-time users.

An operation-concept mismatch: This is where an operation is not what the user either wants or expects. In other words; the operation does not perform a series of actions in the way the user requires.

For example; a user may want to merge two files. If no such operation exists then this may create an operation-concept mismatch. Alternatively, the operation might be acceptable to the user, but may occur in the wrong context. ***For example;*** there are systems where the line delete operation deletes the line below the cursor rather than the line the cursor is actually on. The user might consider the operation to have been correct, but in the wrong context. Again this is an example of an operation-concept mismatch.

An operation-symbol mismatch: This is where the symbol representing an operation is not what the user would expect or want. ***For example;*** a user might want to print a file. If the command for printing a file is "nroff -ms *filename* > *printfilename*" followed by "lpc *printfilename*" rather than "print" then this may create, amongst other things, an operation-symbol mismatch. Alternatively, a symbol associated with an operation may be in the wrong context as far as the user is concerned. ***For example;*** a command name in the wrong menu would be classed as an operation-symbol mismatch.

The process of classification need not be complicated. Here is an example of the classification process:

Stage 1 Identification of the dialogue failure: Fred reports problems while trying to print a file.

Stage 2 Is it an object or an operation that is at odds with the user's view? Fred reports that he does not know the command to print and that he cannot see a command on the screen that might possibly cause his file to be printed. As we can see that Fred's file looks OK then there appears to be a problem with the ***operation*** to print a file.

Stage 3 Is it the concept or the symbol of the object or operation that is wrong as far as the user is concerned? As we know that the command "spool" will cause a file to be printed then it looks as if the concept of the operation fits in with what Fred wants. Yet the symbol "spool" does not appear to be what Fred expected. Therefore, this is a ***symbol*** mismatch.

Stage 4 The way in which the mismatched element (the concept or symbol) contributed towards the dialogue failure is then described. Fred did not appear to associate the symbol "spool" with the concept of printing his file. In this example there appears to be only one mismatched element (the operation-symbol mismatch) which has contributed towards the dialogue failure.

The example above shows the questions that might be asked during the classification process. These questions were taken from figure one shown on page two. The mismatch in this example is of the operation-symbol type; where an operation (the operation to print) has the wrong or unexpected symbol as far as the user is concerned.

USERS' PROBLEMS.

In this section a number of problems that users experienced will be described. Your task is to classify these problems using the classification scheme.

It is important to remember that you are classifying each problem from the user's point of view. Whether the user was sensible or the critisims reasonable does not matter - the point of this exercise is only to classify the problem. Moreover, what can be done to change the system or the training, or whether it is worthwhile changing anything at all is a design question that comes after this classification.

First, three examples:

Example 1 **Newwave**

SITUATION: User had selected "create a new" from the "file" menu and had found the "Post-it" object in the "create a new" window.

PROBLEM: User commented that he did not like the name of "Post-it". He thought this was confusing and he stated that he would prefer it if the object in question was called "memo" or something similar.

CLASSIFICATION:	object/ operation	concept/ symbol
Element1:	Object	Symbol
Description:	The user did not like the symbol "post-it".	
Element2:	Object	Symbol
Description:	The user reported that he would prefer the symbol "memo".	
Element3:		
Description:		

COMMENTS: Whether or not there are two elements to this problem, or whether there is only one element is a possible subject for discussion.

While you are working through the problems the category under which each problem comes may not seem immediately obvious. Indeed, the idea of the scheme is that it should structure the way we think about the user's problems. Consequently, thinking in terms of the classification may seem a little difficult at first. As you get used to the classification scheme, please try to think about whether it is useful as a way of analysing the problem and note these down in the comments section if you can.

Example 2 **Newwave**

SITUATION: The user had a memo (Post-it) in an envelope and had completed the distribution list.

PROBLEM: The user placed the envelope with memo in the mailroom and expected this operation to send the memo. He did not expect to have to do anything else. In other words; he did not expect to have to use the "transfer mail" command or any of the commands available in the mailroom window.

CLASSIFICATION:	object/ operation	concept/ symbol
Element 1:	Operation	Concept

Description: The user did not appear to view the further operation of "transfer mail" as being necessary.

Element 2: ______ ______

Description: ______

Element 3: ______ ______

Description: ______

COMMENTS: This problem appears to have only one element although you may disagree!

SITUATION: The user had entered the distribution list window with the intention of adding some names to the distribution list.

PROBLEM: The user looked under the "edit" menu to attempt to find a way of getting into the edit mode. It was explained to the user that to edit the list she needed to change the mode under the "mode" menu. The user complained that she thought it more logical for this command to be under the edit menu.

CLASSIFICATION:	object/ operation	concept/ symbol
Element 1:	Operation	Symbol

Description: The symbol for evoking a change in the mode appears to be in the wrong context (menu)

Element 2: ________ ________

Description: ________

Element 3: ________ ________

Description: ________

COMMENTS: The symbol of the operation to change the mode to edit appears not to have been mismatched with the user's model of the task. But the symbol does seem to be under the wrong menu.

As you work through the problems below, please try to explain your reasons for your classifications in the comments. You may find that some of these problems have more than one possible element (classification). Please put all of your classifications down and try to justify each one if you can.

If you have any thoughts about the classification scheme itself then please jot these down in any of the comments sections. Remember, it is the classification scheme that is being tested, not you!

Please ignore the "new class" lines (under the comments section) for now. If you have any problems then please refer back to the summary of ECM provided earlier.

GOOD LUCK!

SITUATION: The user was attempting to select the "create a new" command from the "file" menu.

PROBLEM: The user complained about the menu selection, stating that he did not like having to hold down the mouse button. He said he would prefer to click on a menu and for it to stay open until he clicked on the command he wanted.

CLASSIFICATION: object/ operation | concept/ symbol

Element 1: ______ ______

Description: ______

Element 2: ______ ______

Description: ______

Element 3: ______ ______

Description: ______

COMMENTS: ______

NEW CLASS: ______

SITUATION: The user was looking at the incoming mail.

PROBLEM: The user complained that he wanted envelopes opened for him and that he was not altogether keen on things arriving in envelopes at all.

CLASSIFICATION: object/operation concept/symbol

Element 1: ______ ______

Description: ______

Element 2: ______ ______

Description: ______

Element 3: ______ ______

Description: ______

COMMENTS: ______

NEW CLASS: ______

SITUATION: The user was trying to create a new Post-it.

PROBLEM: The user looked under the "items" menu and then explained that he was looking for the "create a new" command.

CLASSIFICATION: object/operation concept/symbol

Element 1: ________ ________

Description: ________

Element 2: ________ ________

Description: ________

Element 3: ________ ________

Description: ________

COMMENTS: ________

NEW CLASS: ________

SITUATION: The user opened a Post-it with the intention of adding text.

PROBLEM: The user began looking under the menus and explained that he expected to have to select an edit mode from the "edit" menu before he would be able to add text.

CLASSIFICATION: object/ operation | concept/ symbol

Element 1: ________ ________

Description: ________

Element 2: ________ ________

Description: ________

Element 3: ________ ________

Description: ________

COMMENTS: ________

NEW CLASS: ________

SITUATION: The user was attempting to send a CV and a post-it together.

PROBLEM: The user had to be told that the way to send the two together was to put them both in an envelope.

CLASSIFICATION: object/operation concept/symbol

Element 1: ______ ______

Description: ______

Element 2: ______ ______

Description: ______

Element 3: ______ ______

Description: ______

COMMENTS: ______

NEW CLASS: ______

SITUATION: The user had placed a memo and a CV in an envelope.

PROBLEM: The user stated that he would like to be able to number the different items so that the receiver would know which to read first.

CLASSIFICATION: object/operation concept/symbol

Element 1: ________ ________

Description: ________

Element 2: ________ ________

Description: ________

Element 3: ________ ________

Description: ________

COMMENTS: ________

NEW CLASS: ________

PAGE NUMBERING AS ORIGINAL

SITUATION: The user was attempting to select a command from a menu.

PROBLEM: The user double clicked the menu head several times in an attempt to open it. She did not remember, until told, that she should hold the mouse button down to kept the menu open.

CLASSIFICATION: object/ operation concept/ symbol

Element 1: ____________ ____________

Description: ______________________________

____

Element 2: ____________ ____________

Description: ______________________________

____

Element 3: ____________ ____________

Description: ______________________________

____

COMMENTS: ______________________________

____

____

____

NEW CLASS: ______________________________

____

____

____

SITUATION: The user needed to add names to the distribution list.

PROBLEM: The user had the envelope window open, but reported that she did not know how to open the distribution list. In other words; she did not know that it could be opened in the same way that other objects could.

CLASSIFICATION: object/ operation concept/ symbol

Element 1: ______ ______

Description: ______

Element 2: ______ ______

Description: ______

Element 3: ______ ______

Description: ______

COMMENTS: ______

NEW CLASS: ______

SITUATION: The user had just created an envelope and needed to enter the names of those to whom the memo was to be sent.

PROBLEM: The user stated that she wanted to address the envelope by writing on its front - not by creating a distribution list.

CLASSIFICATION:	object/ operation	concept/ symbol
Element 1:	______	______
Description:	______	
Element 2:	______	______
Description:	______	
Element 3:	______	______
Description:	______	

COMMENTS: ______

NEW CLASS: ______

SITUATION: The user had added text to a post-it and was closing the window of the post-it.

PROBLEM: When the "save changes?" window appeared the user commented that she thought the changes would be saved automatically and that the window was unnecessary.

CLASSIFICATION: object/ operation concept/ symbol

Element 1: ____________ ____________

Description: ______________________________

____

Element 2: ____________ ____________

Description: ______________________________

____

Element 3: ____________ ____________

Description: ______________________________

____

COMMENTS: ______________________________

____

____

____

NEW CLASS: ______________________________

____

____

____

SITUATION: The user had the mailroom window open.

PROBLEM: After the user was told how to recover items in the mailroom (by dumping the contents of the mailroom to the wastepaper basket) she commented that she would prefer a method which was "easier" and did not entail all of the mailroom contents being removed.

CLASSIFICATION: object/operation concept/symbol

Element 1: ______ ______

Description: ______

Element 2: ______ ______

Description: ______

Element 3: ______ ______

Description: ______

COMMENTS: ______

NEW CLASS: ______

SITUATION: The user needed to copy the contents of a distribution list to a distribution list of an envelope.

PROBLEM: The user stated that she had no idea of how to do this.

CLASSIFICATION: object/ operation concept/ symbol

Element 1:

Description:

Element 2:

Description:

Element 3:

Description:

COMMENTS:

NEW CLASS:

SITUATION: The user had the mailroom window open.

PROBLEM: The user complained that although it was shown how many messages were waiting to go, she would have liked to know which messages were waiting to be transferred.

CLASSIFICATION: object/ operation concept/ symbol

Element 1: ________ ________

Description: ________

Element 2: ________ ________

Description: ________

Element 3: ________ ________

Description: ________

COMMENTS: ________

NEW CLASS: ________

SITUATION: The user was editing within the distribution list windows.

PROBLEM: The user stated that she would like to be able to edit the list without having to move the cursor from one box to another.

CLASSIFICATION: object/ operation concept/ symbol

Element 1: ____________ ____________

Description: ____________

Element 2: ____________ ____________

Description: ____________

Element 3: ____________ ____________

Description: ____________

COMMENTS: ____________

NEW CLASS: ____________

SITUATION: The user had created a post-it and already had a distribution list.

PROBLEM: The user tried to put the distribution list on top of the post-it. She explained that she thought that it might be a short-cut, rather than creating a new envelope or placing the memo in the mailroom.

CLASSIFICATION: object/ operation concept/ symbol

Element 1: ____________ ____________

Description: ____________

Element 2: ____________ ____________

Description: ____________

Element 3: ____________ ____________

Description: ____________

COMMENTS: ____________

NEW CLASS: ____________

SITUATION: A mail transfer failed and the user was informed that it was because one of the names on the distribution list was incorrect.

PROBLEM: It was explained to the user that when the transfer failed then the distribution list and memos would be returned. The user commented that she would prefer address/name checking to take place before the transfer.

CLASSIFICATION: object/ operation concept/ symbol

Element 1: ______ ______

Description: ______

______

Element 2: ______ ______

Description: ______

______

Element 3: ______ ______

Description: ______

______

COMMENTS: ______

______

______

______

NEW CLASS: ______

______

______

______

SITUATION: A mail transfer failed and the user was informed that it was because one of the names on the distribution list was incorrect.

PROBLEM: The user stated that she did not like the idea that the whole list would be returned to her and she suggested that only those names/addresses which had failed should be returned.

CLASSIFICATION: object/ operation concept/ symbol

Element 1: ______ ______

Description: ______

______

Element2: ______ ______

Description: ______

______

Element3: ______ ______

Description: ______

______

COMMENTS: ______

______

______

______

NEW CLASS: ______

______

______

______

Thank you for working your way through these problems.

Could you now get together with some of your colleagues and go through each problem again, looking for any classifications that do not agree. In other words; if you have classified a particular problem as having three elements and your colleagues have classified only two elements for the same problem then you should discuss the problem and try to agree on the numbers and types (classifications) of elements for each problem.

1) Any **new classifications** should be marked on the "NEW CLASS" line (eg the elements of the problem are now classified as an operation-concept mismatch and an object-symbol mismatch).

2) If you already agree with all of your colleagues on the classifications for a particular problem then just mark the "NEW CLASS" line with a ***cross***.

3) If your colleagues have chosen different classifications to you, but they change their minds and agree with your classification then write their ***names*** on the "NEW CLASS" line.

4) If you cannot agree with all of your colleagues then please mark the "NEW CLASS" line with ***"cannot agree"***.

Appendix 6:

Usability Questionnaire

ECMQuestionnaire.

Please answer all questions by ticking the appropriate scale under each of the questions. If you feel that you do not know enough to provide an answer then please tick the 'don't know' box. If you have any comments relating to a question then please add these on the lines provided below the question. All comments and observations will be very much appreciated.

1 How well do you understand the classification scheme?

I understand nothing of it	I understand some of it	Familiar with it	Quite well	Very well	Don't know
☐	☐	☐	☐	☐	☐

Comments:
....
....
....
....

2 Was the classification scheme easy to learn?

Hard to learn	Not particularly easy	Moderately easy	Quite easy	Very easy	Don't know
☐	☐	☐	☐	☐	☐

Comments:
....
....
....
....

3 Did you find that the classificatory scheme was a useful way of thinking about user errors?

Of no help at all	Not particularly useful	Moderately useful	Quite useful	Very useful	Don't know
☐	☐	☐	☐	☐	☐

Comments:
....
....
....
....

4 Have you used the concepts of the classificatory scheme during discussions about user errors?

Not at all	Occasionally	Some of the time	Quite often	Very often	Don't know
☐	☐	☐	☐	☐	☐

Comments:

....

....

....

....

5 Are you likely to use this scheme in future discussions about user errors?

Very unlikely	Unlikely	Possibly	Quite likely	Very likely	Don't know
☐	☐	☐	☐	☐	☐

Comments:

....

....

....

....

6 Were the concepts of the classificatory scheme used during discussions about user errors?

Never	Occasion-ally	Some-times	Quite often	Very often	Don't know
☐	☐	☐	☐	☐	☐

Comments:

....

....

....

....

7 Was it possible to classify all user errors under the scheme?

No, none at all	A few errors	Most errors	Almost all errors	Yes, all errors	Don't know
☐	☐	☐	☐	☐	☐

Could you give details of either the types of errors that were difficult to classify or specific examples that you can remember?

....
....
....
....
....
....
....
....
....

8 Do you think that the classificatory scheme could be used to classify user errors on other systems?

Definitely not	Unlikely	Possibly	Quite likely	Most certainly	Don't know
☐	☐	☐	☐	☐	☐

Comments:

....
....
....
....

9 Does the classificatory scheme provide enough detail for you to suggest design changes?

No detail at all	Not enough detail	Adequate detail	Yes, enough detail	Yes, more than enough detail	Don't know
☐	☐	☐	☐	☐	☐

Could you suggest ways in which the detail the scheme provides might be improved?

....
....
....
....
....
....
....
....
....

10 Does the classificatory scheme adequately suggest ***which parts*** of the design need to be changed?

Definitely not of use	Unlikely to be of use	Possibly helpful	Quite helpful	Yes, very helpful	Don't know
☐	☐	☐	☐	☐	☐

Could you give reasons for your answer?

....
....
....
....
....
....

11 Does the classificatory scheme help you to decide ***how*** different parts of the design might be changed?

Definitely not of use	Unlikely to be of use	Possibly helpful	Quite helpful	Yes, very helpful	Don't know
☐	☐	☐	☐	☐	☐

Could you give reasons for your answer?

....
....
....
....
....
....
....
....

12 Do you believe that this classificatory scheme is generally usable?

Hard to use	Not particularly usable	Moderately usable	Quite usable	Very usable	Don't know
☐	☐	☐	☐	☐	☐

Comments:

....
....
....
....

13 Do you think other people in Hewlett Packard are likely to consider this scheme as a useful way of thinking and talking about user errors?

No one at all	Only a few people	Some people	Quite a few people	Most people	Don't know
☐	☐	☐	☐	☐	☐

Comments:
....
....
....
....

14 Would you want to see this scheme used during future usability testing?

Never	Occasion-ally	Some-times	For some of the testing	During all testing	Don't know
☐	☐	☐	☐	☐	☐

Comments:
....
....
....
....

Could you suggest ways in which the classificatory scheme might be improved?
....
....
....
....
....
....
....
....

If you have any further comments on the ECM classificatory scheme then these would be very much appreciated:
....
....
....
....
....
....
....

Appendix 7

ECM Meeting and Discussion: Verbal Protocol.

There were eight people present at the meeting, which took Place at Hewlett Packard, Pinewood, Wokingham, and lasted approximately two hours on a morning in September 1987. The people present included the author (Paul Booth), three software engineers, two human factors engineers, a technical writer, and a Senior Learning Product Manager. With the exception of the author, all of people present at the meeting were working on the product that was being discussed. All of the people present had classified the problems with the product, shown in the previous appendix, using ECM prior to the meeting. Where it was not possible to identify words speakers used, because they were too muffled or quietly spoken a "xxxx" has been placed for each word. When a string of words could not be identified they have been marked as "...(not discernable)...".

PROBLEM 1

Human Factors Engineer 1	So what's everyone got?
Software Engineer 2	I classified it as an operation concept.
Human Factors Engineer 1	What's the description?
Software Engineer 2	Errm.....I've got that the user seems to feel that the way in which he selects a menu item is unexpected or unwanted.
Software Engineer 1	I got that, I also thought there was another operation concept problem in that the user would have preferred a single click.
Technical Writer	But is that any different I mean.......
Software Engineer 1	Well, no I was wondering about that (general laughter).
Human Factors Engineer 1	In actual fact I think it's an operation symbol problem....
Human Factors Engineer 2	Yes, I thought it was a symbol mismatch and.....
Human Factors Engineer 1	It's just that the behaviour of the pull down menu is not what the user would like. I think that......(not discernable).
Software Engineer 2	Surely, if it's the behaviour then it's a....errm...concept thing.
Software Engineer 3	I got symbol.......
Human Factors Engineer 1	Well no because the menu's OK, it's just the symbol of the menu....errm....that's not right as far as the user's concerned.
Software Engineer 3	The reason I had for the symbol mismatch was that the concept of selecting commands via menus is OK, but the physical mechanism is unergonomic.
Software Engineer 2	But that must be a concept problem.
Human Factors Engineer 1	What's the difference between a symbol and a concept problem? What's the answer? (general laughter).

Paul Booth Well there isn't really. I don't want to say anything because I might influence you in your decisions and in how you think about both the classification scheme and the system. The right answer is the one you agree on, you're the test of what's right.

Learning Products Engineer That makes me feel confident (general laughter). I decided upon an operation concept....errr....because....errr...the user preferred a different behaviour from the system than what it gave.

Software Engineer 3 So really, it's an operation concept because the system's behaviour wasn't right.

Learning Products Engineer Yes.

Human Factors Engineer 1 Yeh, I agree with that.

PROBLEM 2

Human Factors Engineer 1	Any Votes?
Learning Products Engineer	I got an operation concept and an object concept problem. My..
Human Factors Engineer 1	So did I.
Learning Products Engineer	My classification...my description was that, for the operation concept, that....errr.... the user wanted envelops to be opened automatically, but I'm not entirely sure what we might mean by automatically. My object concept description was that the user didn't like the concept of envelops.
Software Engineer 1	Well I got those but I don't think the user didn't like envelops, he just didn't like things arriving in them (general laughter).
Learning Products Engineer	Well yeh, that's what I mean, he just didn't like envelops.
Software Engineer 1	No that's no the same thing, actually........
Human Factors Engineer 1	I think it's the same. I got envelops not perceived to be useful model for messages. But...err.. I'm not sure what we can do about that....errm..as envelops are just central to the mailing part of Newwave.
Human Factors Engineer 2	I agree
Software Engineer 2	Yes, I don't think we can change the envelops, but I have the same classification and description. The problem is that if we.....
Software Engineer 1	OK, I seem to be in a minority...
Learning Products Engineer	At least we all agree on the fix (general laughter).
Human Factors Engineer 1	I'm not sure we need to go any further do we? Have we any other classifications?

PROBLEM 3

Human Factors Engineer 1	I though this was an operation symbol, because the symbol was in the wrong context.
Several People	Yes, Yeh.
Human Factors Engineer 1	Did anyone get anything different?....Oh, I don't know what we can discuss about this (general laughter).
Learning Products Engineer	Something of a rare happening when we all agree.....
Software Engineer 3	We're not used to it (more laughter).

PROBLEM 4

Human Factors Engineer 1 Number four, the user opened a post-it wanting to add text..errm..instead of just typing the text they were actually looking for something saying that that was what they wanted to do.

Software Engineer 1 Operationconcept

Learning Products Engineer Yes

Human Factors Engineer 1 Yeh, and the description of the mismatch?

Learning Products Engineer The user expected to have to select an edit mode.

Software Engineer 3 The user didn't realize that he was on edit mode (general laughter)

Learning Products Engineer Right.

Human Factors Engineer 1 Yeh, we've all got the same for that.

Software Engineer 2 I've also got an operation symbol m... well the description being that there's an edit menu, that merely being there implies that the user needs to act to do something.

Human Factors Engineer 2 (notdiscernable)

Software Engineer 2 Yes well that would be part of the concept....

Human Factors Engineer 1 Well that's a different one, that's the cursor, if the text cursor isn't clear enough to show it's inviting.

Learning Products Engineer Well no.

Paul Booth Let me get this clear, so you now think it's three? Or do you not all agree with that.

Learning Products Engineer I think that there's two

Software Engineer 1 I think two

Learning Products Engineer I thought it was just one actually, but....

Software Engineer 3 I got one because there's two bits to it. One is whether you have to do anything to get into edit mode and also if you do have to do anything where will it be.

Learning Products Engineer Yes, I thought Richard's (SE3) description was quite right, that fact that they didn't realize that they were already in edit mode and....

Human Factors Engineer 1 Which is (not discernable)

Learning Products Engineer Yes

Human Factors Engineer 1 Quite aside from the fact that he wasn't expecting it anyway (generallaughter).

Human Factors Engineer 2 Just one comment I put down here. This doesn't actually tell us very much about what we think we should do about it:...because (not discernable).

Software Engineer 3 Well it depends how you classify, if I say that he didn't realize he was in edit mode that seems to imply that it needs to be made more clear.

Human Factors Engineer 1 Which means you're changing the symbol, which means it should be classified as an operation symbol. That's the way this thing is supposed to work isn't it? You work out what it is you're supposed to change and what it is......

Human Factors Engineer 2 (not discernable) .,wasn't always that there was a symbol.

Human Factors Engineer 1 Well according to this classification it's the concept that's wrong and the user, whose natural concept is to think in terms of an edit mode, then what we should do is say OK you can press a button to get into edit mode (general laughter) if that's what we're going to do.

Human Factors Engineer 2 That's the danger you do what appears to be the xxxxx the user would already expect to happen in that case, and that might not be the best solution.

Human Factors Engineer 1 Well, I mean...

Learning Products Engineer I think we're right that it's a conceptual problem, it's just that normally when you open something you are in edit mode, that's just one of the basic rules of the system.

Human Factors Engineer 1 So the way you could get round that one by xxxxx a conceptual problem is to ..xxxxx.. flag it in training, because it's a conceptual xxxxx. So that would be one of the fixes.

Learning Products Engineer Yes, it's an interesting one because we do have objects now that are xxxx xxxx only, particularly when things are centred round a mouse system and people haven't got the software on their PCs.....

Human Factors Engineer 1 xxxxxxxx xxxxxxx I think with each problem we would need to have an indication of the number of people who have a problem as opposed to those who don't...(not discernable)...it wouldn't be sufficient to label this as a problem without any contest.

Human Factors Engineer 2 This scheme doesn't give you any idea of what the priorities are with objects and operations, or how many people they affected.

Human Factors Engineer 1 (not discernable)..I think it can help you understand where the fact of the problem is coming from and therefore what you can do to fix it because if you start with several fixes you could start seeing several errors.

Learning Products Engineer Presumably what we learn or report, in a full report, would in fact have the number of things that happen and actually.....(not discernable)...some alternative classification as to how serious it is.

Human Factors Engineer 2 It doesn't actually help us classify how serious it is.

Learning Products Engineer No it doesn't classify seriousness, but at least at least it tells us what the problem is, but seriousness is I guess a symptom that's not...not... and this technique is just to say, just to classify what the problem is rather than how serious it is.

Human Factors Engineer 2 Just coming from the team's point of view it's useful to know which one to fix first.

Learning Products Engineer Oh Yes, this is true, yes it's probably still...yeh useful to do that first so we can say OK we've got to fix this problem.

Technical Writer Some problems seems one-off, it seemed to be just one user who had problems.

PROBLEM 5

Human Factors Engineer 1 Right, number five...(not discernable).

Learning Products Engineer No I don't think so.

Human Factors Engineer 1 Err...the user was attempting to send a cv and a post-it together without an envelop and the user didn't realize that had to put them in envelops to send them together.

Software Engineer 1 An operation concept.

Several people: Yes, Yeh.

Software Engineer 2 And object concept.

Human Factors Engineer 2 I got an object symbol.

Human Factors Engineer 1 What object symbol (general laughter).

Software Engineer 1 Try and explain.

Human Factors Engineer 2 The envelop object didn't convey its meaning to the user.

Learning Products Engineer Yes, it would be interesting with this one to actually know whether they, they had been shown the envelop, or whether that missed or not.

Human Factors Engineer 1 Well, they'd all seen envelops during the training..errm. as far as I can see the user had forgotten about envelops as an idea..(not discernable)

Learning Products Engineer Yeh.

Human Factors Engineer 1 Have we anything else there?

PROBLEM 6

Human Factors Engineer 1	Errm..the user placed a memo and a CV in an envelop and stated that he would like to be able to number the items so that receiver would know which to read first.
Learning Products Engineer	I classified this one as operation symbol.
Human Factors Engineer 2	Why? (generallaughter).
Human Factors Engineer 1	I got three, operation symbol, object concept and operation concept(generallaughter).
Learning Products Engineer	Well the operation symbol I had was that you wanted number the parts in the envelop..errm...
Software Engineer 1	Isn't that the concept?
Learning Products Engineer	It's not the concept.....(not discernable)
Human Factors Engineer 2	There was actually a mechanism where you can actually order them...(not discernable).
Human Factors Engineer 1	That's exactly what I've got with operation symbol, because the user had said what he wanted to know to achieve was that the receive would know what to read first..(not discernable)..so that's the fact.
Software Engineer 3	I got that down as object symbol as he didn't know how items were ordered.
Human Factors Engineer 1	You've got that as operation..
Software Engineer 3	object
Human Factors Engineer 1	..object symbol
Software Engineer 3	Oh well, errm.. I got operation concept in that the user didn't understandthat items were ordered and object symbol the user didn't understand how items were ordered. So, ..(not discernable).
Human Factors Engineer 1	Well I got object concept...(not discernable) (general laughter)
Software Engineer 1	I've got that description (more general laughter).
Human Factors Engineer 1	I got an operation concept in that ordering is only achieved by removing items and replacing them in an envelop.
Software Engineer 1	There isn't that going out...(not discernable)
Learning Products Engineer	That's going outside the actual...
Human Factors Engineer 1	But basically the operation concept of order is to number, because he said he wanted to number the items it's the concept..
Software Engineer 3	The two are related, one is that they don't know that they're ordered at all and they want them to be in a different order. To me the operation concept was that he didn't understand that they were ordered. Perhaps in the same terms he wanted them to go in the order they were in, but he didn't realize that they were already in that order. He wanted them to be in a different

should be a natural concept (general laughter).

Technical Writer — The listing on the desk top is different to the ordering in an envelop, because on the desk top you can explicitly say I want ordered by name, date etcetra.

Human Factors Engineer 1 — This is where I got my object concept from........ It's very hard to know when you've got a problem that ends up being xxxxx.

Several people: — Yes, yeh.

Human Factors Engineer 1 — I mean different elements of the problem....

Software Engineer 3 — I don't think it's the case that you've got to have one and you can only provide the solution to one part of the problem. It's just a way of trying to understand the problem.

Learning Products Engineer — Certainly in the training it points to the fact that we have to point out that the order in which people actually receive, when they receive an envelop they'll actually see the things in the order in the envelop.

Software Engineer 3 — Also, presumably, somebody's making a decision about delimiting the problem. That's to say, that while one person might say that this is one problem, while somebody else observing it might say this is actually two problems. One they don't understand the ordering, and two they don't know how they could order it.

Human Factors Engineer 1 — Yes, I remember thinking there was a problem with this. In that it actually happened when I was doing the user testing and that was that if you had actually done this wrong you had to take everything out of the envelop.

Human Factors Engineer 2 — This reminds me of the advance mail problem of not being able to read what you just put into an envelop so you can't tell what order its in....it was exactly the same issue.

Learning Products Engineer — Well (general laughter) what's the classification? The fact that this reorder mail operation is missing.

Human Factors Engineer 1 — That's an operation concept...

Learning Products Engineer — That's an operation concept...

Human Factors Engineer 1 — In that you want to order the way you achieve it is conceptual...There's the operation symbol problem of you don't know what it means in the first place. There's the object concept problem....

Learning Products Engineer — (notdiscernable)

Human Factors Engineer 1 — Well lets see what we've got that we've agreed. There's an operation symbol problem that the user hasn't realize it's ordered in the first place.

Software Engineer 3 — I said that was a concept actually.

Human Factors Engineer 1 — And you still want to stick by that.

Software Engineer 1 Yeh, that they didn't understand the re-ordering, which I think's a concept.

Human Factors Engineer 2 It seems pretty abitary.

Learning Products Engineer I think we have to say that this is a symbol because you can actually look at the screen and things are ordered, its just that they haven't realized the significance of that.

Human Factors Engineer 1 I'm still lost....... So there's two operation concept problems and an operation symbol problem. And conceivably three or four object concept problems (general laughter).

Software Engineer 3 So how does this help us.

Human Factors Engineer 1 I guess that this would help us if we had the object concept problems laid out, at least we would be making conscious decisions to xxxx the system which would make the system more consistent.

Human Factors Engineer 2 At least this helps you identify where the problems are.

Software Engineer 3 It's better to be exhaustive and think of things that shouldn't be there.

Learning Products Engineer It's interesting that what seemed to be quite a simple problem when we started turned out to be quite a nasty one.

Human Factors Engineer 1 What's key may be the description rather than how you classify.

Software Engineer 3 Another thing that's interesting is to consider whether different classifications might produce the same solution.

PROBLEM 7

Human Factors Engineer 1	Number seven, any votes?
Several people:	Objectconcept.
Human Factors Engineer 1	Anydisagreements?
Human Factors Engineer 2	I got an operation concept.
Human Factors Engineer 1	I got an object concept. How many object concepts? One, two, three, four. Operation concepts? Three. What was your description of the problem?
Software Engineer 2	The user wanted to be able to set urgent.
Learning Products Engineer	I've got object concept down xxxx the user wanted an urgent facility.
Software Engineer 3	The object doesn't have the capability to...
Human Factors Engineer 1	The way I was looking at it was the object itself. It didn't have the notion of itself being urgent (general laughter).
Human Factors Engineer 2	Did the object in the envelop xxxx xxxxx?
Human Factors Engineer 1	Yes.
Learning Products Engineer	I must admit, I wasn't sure in my own mind. I plumped for object concept but I can see that it could be classified as operation concept. We've agreed as to what the problem is but..errr...
Human Factors Engineer 1	Again I called it object concept because I knew something I want to change is in the envelop. You know it's properties of something rather than how I'd provide the operation.
Software Engineer 3	It's the property of the envelop rather than a property of the Newwave system at all.
Human Factors Engineer 2	I think of it as an operation he wants to carry out on an object.
Human Factors Engineer 1	I think of it as an object concept as the system itself wouldn't know what to do with it even if the user did have an operation to tag it urgent.
Learning Products Engineer	There is not concept of an object being urgent, you can't tell in your in tray if something's urgent or not.
Software Engineer 1	I read this as tag being a noun rather than a verb.
Human Factors Engineer 1	At least we've agreed it's a concept problem.

PROBLEM 8.

Human Factors Engineer 1 Eight. Any votes?

Software Engineer 1 Operationconcept

Others: Yes, yeh.

Human Factors Engineer 1 And what's the description?

Software Engineer 1 The user expected the mailing operation to work differently.

Human Factors Engineer 1 Do we have anything else?

Software Engineer 3 Yes, the operation was misunderstood with respect to the extent of the mail operation.

Learning Products Engineer I had it classified as object concept, but y'know again...

Human Factors Engineer 1 So did I.

Learning Products Engineer ...No the user was expecting the message would be filed automatically.

Human Factors Engineer 1 I've got object concept for this in the same way as I had object concept the previous one in that there isn't a place in the system that can cope with autofiling.

Learning Products Engineer I'm willing to re-classify mine as operation concept.

Human Factors Engineer 2 That's not to say that they're not both things.

Human Factors Engineer 1 You need the concept of autofiling in the system before you can say that xxxx xxxx xxxx.

Human Factors Engineer 2 Yeh, but you need to tag the operation to the object.

Learning Products Engineer It seems a bit odd, because there's nothing about the filing cabinetobject that suggests that it will carry out automatic filing, or is there?

Software Engineer 2 I don't know why we're so bothered about the object. After all agents operate on objects....

Human Factors Engineer 1 Any votes?

Software Engineer 3 Operation symbol and operation concept.

Human Factors Engineer 1 What's your operation symbol?

Software Engineer 3 I haven't described that (general laughter). The operation concept was that the user was trying to open a menu head. So they picked up the concept of open and applied it to menus.

Human Factors Engineer 1 Yeh, I've got operation concept in that they've assumed a double click for all types of objects. Anything else?....I've got an object concept which is that the menu has been seen as a window, it was seen as something that was opened.

Software Engineer 3 I guess my operation symbol was that if you view that they're trying to open it, that's no the way you open it.

Learning Products Engineer I though this was the same as problem number one in fact. The fix is the same, which is that the menu stays when it drops down.

Human Factors Engineer 1 I don't know, 'cause the still going to try a double click if they want to open the menu.

Learning Products Engineer OK, this guy did not like having to hold down the mouse button and this one had to be told to hold down the mouse button. It's the same problem with a slightly different...

Human Factors Engineer 1 Oh no, no, no, no that's not what's happening here....

Software Engineer 3 tut, tut, tut, tut....

Human Factors Engineer 1 The user before knew she wanted to select and didn't like having to keep holding it to keep it open....

Learning Products Engineer Right...

Human Factors Engineer 1 Whereas this user doesn't know she's to select she's double clicked and what she's trying to do is open.

Software Engineer 1 (notdiscernable)

Software Engineer 2 It's interesting as well whether or not you'd be more concerned in this case about how the user opens that in the other case as well....(not discernable).

Human Factors Engineer 1 Well this problem doesn't look xxxx like it.....(not discernable).

Learning Products Engineer No, well we could say that this is the same problem, but that this one's been a lot worse than this one (general laughter).

Human Factors Engineer 1 No I do think this one's more serious, there's a definite object concept with them where a menu is seen as an object which is opened just like windows do........How would we fix it?

Software Engineer 1 Presumably it's part of windows training.

Human Factors Engineer 1 I was just wondering whether double click is a problem.

Software Engineer 3 It depends on how many people it happened to. This is where we need the frequency data really.

Human Factors Engineer 2 This one was actually quite common from what I can remember, it's just people learning that menus are not like windows.

Learning Products Engineer It's the fact that the user has to hold down and then drag and that has, y'know......(not discernable)

Human Factors Engineer 1 So did we all have operation concept?

Several People: Yeh, mmm, yes.

Learning Products Engineer Right, I put down object symbol for this one.

Software Engineer 3 I've got down operation concept. Just to link in with the previous one....

Learning Products Engineer Perhaps I ought to do the description. I'm saying it's object symbol. User did not realize that distribution list and text view was an object.

Human Factors Engineer 1 I've got that too, but I've also got operation concept.

Software Engineer 3 Operation concept I've just put it down that the pervasiveness of opening was not apparent. Which is in a way related to the previous thing where they tried to apply it too generally whereas here on the second xxxx there are not applying it generally enough. But that's presumably the same thing as saying they don't realize it's an object.

Learning Products Engineer That's right, I'm saying they're use to seeing icons on screens and opening those but when see something in text and it's just a name they don't realize that they can open it.

Human Factors Engineer 1 So how many object symbols did we get? One, two, three, four. Right, and operation concept? Operation symbol. How would you describe it as symbol?

Technical Writer (notdiscernable)

Software Engineer 2 I had it because it closely resembled we had before in that opening an icon was an operation ......(not discernable).

Software Engineer 1 The fix seems to be the same really...

Learning Products Engineer It's obviously something that needs to be addressed in training.

Human Factors Engineer 2 We seem to be able to spot the fixes, but not always the classification. I would have thought that it was the fixes that this should identify.

Human Factors Engineer 1 It's only when you've worked at the underlying causes and thought about a fix that you can actually get to the classification...So possibly the value of the classification is in forcing you to come to some conclusion.

Learning Products Engineer In some ways, yeh. And also it's stopping you to jumping to a particular conclusion and if you actually go through it well OK and say well could it be that sort of problem and could it be that sort of problem you can come up with alternative....

Software Engineer 3 Yes but what were looking at, and obviously any classification introduces bias so obviously we have to say is this a good bias that we look at things in terms of whether they're objects, operations, concepts or......there's not way you can........

Human Factors Engineer 1 Well that's what I was saying, would you like to target your problem descriptions around these classifications and think how can I justify the choice I've made or do you still go through the process of trying to work out what all the underlying causes are...Anyway that the final xxx

PROBLEM 11

Human Factors Engineer 1 Eleven. Any votes?

Software Engineer 1 Objectconcept.

Several People: Yeh.

Software Engineer 3 Object symbol.

Human Factors Engineer 1 Objectsymbol?

Software Engineer 3 Yeh.

Human Factors Engineer 1 Description?

Software Engineer 3 Yeh (laughter). Oh, The danger with intuitive in qutoes sysms is thatintuitionin quotes is not guaranteed to be conventional in quotes. (general laughter). I had thought whether it was symbol or not.

Human Factors Engineer 1 It's difficult, because I just thought of this xxxx because an envelop has an address on the front of it err... so in addressing it you think of putting an address on it instead of sticking in a distribution list.

Learning Products Engineer Yeh, that's right. The description I had was that the user did not expect to have to open an envelop in order to address it.

Human Factors Engineer 1 Yeh, so it's concept although it could be....the symbol......

Learning Products Engineer Well we can argue that you have a concept symbol mismatch in a way in that this thing pretended it's an envelop but isn't, it's not really an envelop here.

Human Factors Engineer 1 It could be an operation concept if it's the operation of addressing its.....you have to open the envelop.

Learning Products Engineer Yeh, in the majority of cases this thing behaves like an envelop, but in this particular aspect it's not quite the same.

Human Factors Engineer 1 It's interesting whether it's object concept because the envelop was wrong or whether it was operation concept because the act of addressing was wrong.

Software Engineer 3 One thing that came out of this.....

Technical Writer I had both. The envelop was at fault because it was an object concept problem and an operation concept because it was some sot of missing functionality such as edit the address.....

Human Factors Engineer 1 So how do we fix it?

Software Engineer 3 With a little scroll bar....(general laughter).

Human Factors Engineer 1 OK.

PROBLEM 12

Human Factors Engineer 1	Objectconcept?
Software Engineer 3	Yes. I got the user believed in what you see is what you have the ability to undo changes rather than to save changes seemed morenatural.
Human Factors Engineer 1	Absolutely, absolutely.
Learning Products Engineer	Well yes, I must admit that I've got this one down as operation concept user expected changes to be saved automatically.
Human Factors Engineer 1	I've got it as object concept the user thinks he's got it already.
Software Engineer 3	People have no idea of the concept of save changes, people believe they are editing.
Technical Writer	Well that's right.
Learning Products Engineer	Well that's what I'm saying, this operation should not be here basically.
Human Factors Engineer 2	Is it not the symbol of that operation (general laughter).
Software Engineer 2	I think we've got the description pretty much the same haven't we? How you actually classify it....
Human Factors Engineer 1	How many operation concepts here? Three. I really don't know.
Software Engineer 3	I think as James said it's not the operation save changes it's the operation of editing it that at fault. If you view it as an operationconcept.
Human Factors Engineer 1	I'm vaguly saying object concept because it's the object that's not behaving as people expect objects to behave. The object is something other than what he thinks it is.
Software Engineer 2	It could be a different type of operation concept it could be just the close...errm... operation which he's complained about, where she though that close....that she was working on a tempory copy, but that when she closed she expected it to be savedautomatically.
Human Factors Engineer 1	Because I classified it as object concept, because I don't think anybody actually xxxx the concept xxx xxx...
Human Factors Engineer 2	I think this is the same as the one before where the user had to choose edit to actually put the text into store......(not discernable)
Human Factors Engineer 1	It may well be that as well, it is definately an object concept.....

PROBLEM 13

Human Factors Engineer 1 Thirteen.Operationconcept?

Learning Products Engineer I've got it as operation concept.

Software Engineer 1 Operation concept and operation symbol.

Software Engineer 3 Operation symbol.

Human Factors Engineer 1 I've put operation concept because I knew a better way to do it.

Software Engineer 3 I put symbol because the mechanism by which messages are recoved is clumsy. Not that there's anything wrong with the concept.

Learning Products Engineer Well, I put operation concept, and my description was that the user wanted to remove individual items from the mail. I think they knew want they wanted to do it's just that they......

Human Factors Engineer 1 Well the operation to recover the mail that's already been mailed and they know they can 'cause it's there and the nmodel they have is that you take everything you have and tip it into the wastebasket and wade through the paper and then individually mail the individual item.

Software Engineer 1 I got an operation symbol problem because dumping something into the wastebasket doesn't sound like I'm saving it (general laughter).

Human Factors Engineer 1 I say that's another operation concept.

Software Engineer 1 Well, xxx if they'd been saved to wastebasket or...no....just empty into wastebasket sounds....

Learning Products Engineer That's not what she's said. She's prefered a method that was easier, she understood what was going on, it just seemed a clumsy way of doing it. But I take your point I just think ....errm...that...

PROBLEM 14

Human Factors Engineer 1 I've got an operation concept because the user didn't have the concept of the operation (general laughter).

Learning Products Engineer That's pretty much what I've got actually the user had no idea of how to do the....

Human Factors Engineer 1 What has everybody else got? We've got operation concept.

Software Engineer 1 Operationconcept.

Software Engineer 3 Operationconcept.

Human Factors Engineer 2 I've got operation concept.........I've also got an object concept as there's nothing associated with that object which would lead us to....(not discernable).

Human Factors Engineer 1 Is that an object concept or another operation concept?

Human Factors Engineer 2 It's a property of the distribution list.

Software Engineer 3 It's debatable as well, I mean objects have general properties, well objects which hae been choosen for the desktop, but the only problem is with operation such as merge and really that's a containing operation, it doesn't behave in the same way.

Learning Products Engineer Yeh, it's broken the rules. Yeh I'm not...guess what people wanted to do was make a copy of the distribution list and put it in the envelop.

Human Factors Engineer 1 It could be considered an object symbol problem in that the distribution list doesn't look like you could move something into it.

Software Engineer 1 We should have a hole (laughter).

Learning Products Engineer And flashing neon lights.

Human Factors Engineer 1 The fact that she didn't know how to do it, that she hadn't framed in her own mind what I want to do is merge the contents of that with that. I guess even if she did have that concept we knoe people had problems when we were looking at it.

PROBLEM 15

Human Factors Engineer 1 Fifteen. OK, any votes?

Learning Products Engineer I got object symbol the user did not realize that all messgaes in the mailroom are waiting to be transfered.

Human Factors Engineer 1 Anythingelse?

Software Engineer 2 Ahh, that's interesting. To me when I thought it should say give the subject of the messages that are waiting to go rather than they misunderstood that they were all going to go. I knew that, but if it says you've got two messages, I would like to be albe to know....

Learning Products Engineer I suspect what she wanted to was actuial...errr..

Human Factors Engineer 1 I got it the same as you for part of the problem where the representation of the item that are waiting is not what the user wants to see....so maybe the representation should be different...

Learning Products Engineer I think we're in danger of reading too much into this concise problemstatement

Human Factors Engineer 1 Well I think the problem is that she hasn't realsied that everything in the mailbox is waiting to go. I don't know, but I'm assuming that this comes from the situation where people are having so many problems mailing that they often have several items they don't want to mail, as well as the ones they hope they've done right.

Software Engineer 2 I think there's a problem with being concise, because we're certainly not being precise this for a start because I think people trying to use this thing would try to avoid using repetition 'cause where you've got people saying.......well they shouldn't have these problems, and I'm not too sure they do.

Learning Products Engineer When she says which messages she probably means, she probably wants to see the names of the messages that eare waiting to be transfered.

Software Engineer 2 Well it's because whoever wrote this problem isn't clear that xxx and transfer mean the same thing.

Human Factors Engineer 1 Well, the dam thing wasn't working properly at the time, we tried so many messages.

Technical Writer So, do we agree?

Learning Products Engineer Objectsomething.

Human Factors Engineer 2 I think the user expected to see a list of messages. A list view.

Software Engineer 1 I'm kind of torn between object concept and object symbol if they just want the list thebn I think it's and opject symbol, once we give them the list then...

Human Factors Engineer 1	It's the behaviour of the list...
Software Engineer 1	Yes.

PROBLEM 16

Software Engineer 3	Well I got operation symbol in that the mechanism by which input focus moves is hard work.
Human Factors Engineer 1	Operation symbol xxxx.
Software Engineer 3	Yeh.
Human Factors Engineer 1	I got operation concept....(not discernable).
Learning Products Engineer	I've got operation symbol, the description was that the user wanted to be able to edit the list without moving the cursor from the....
Human Factors Engineer 1	This is taking a long time.
Human Factors Engineer 2	I have to go soon, I've a meeting in London.
Paul Booth	Will there be any time for the interview?
Human Factors Engineer 2	Well it depends when we finish this.
Paul Booth	OK.

PROBLEM 17

Human Factors Engineer 1	Right, seventeen. I've got three again here.
Software Engineer 3	Right, the one that I've got is operation concept, and the description is the user misunderstood, or was unfamiliar with the concept of dropping an item into another.
Technical Writer	Sorry, could you say that last one again.
Software Engineer 3	I'm sorry. The user misunderstood, or was unfamiliar with the concept of dropping an item into another.
Human Factors Engineer 1	I got operation concept, not because what she was trying to do wasn't allowed, but because what she wanted to do is...errm...not available, which is a short cut. But I got two other two that were object concepts. The object concept being...errm...what you suppose an operation concept would be but not understand that items were..errm...that text items cannot xxxx xxxx items. And the other object concept was not understanding aboutan envelop (general laughter). Anything else?
Software Engineer 1	Objectconcept, the user wanted more capabilities from DLs and post-its.
Technical Writer	I got an operation concept in that......(not discernable)

PROBLEM 18

Human Factors Engineer 1 Right.

Software Engineer 2 Operationconcept.

Software Engineer 1 Operationconcept.

Human Factors Engineer 1 I've got an operation concept and an object concept. Mainly because, the operation concept is that they want the name checking that isn't there, but the object concept is not realising why it's not reported to be there.

Learning Products Engineer No, I think.....

Software Engineer 1 Oh well, we're no going to beef about that (general laughter).

Software Engineer 2 In fact the only reason we need to make it clear was when people have errors, that's the simplest reason, only when information is required.

Human Factors Engineer 1 I agree.

PROBLEM 19

Software Engineer 3 I got operation concept for this. There was no explanation provided as to why the whole list was returned. Also I've got, in a sense this is also a symbolic error 'cause I've said the correct names could be given to the user, there's no reason they had to see them.

Human Factors Engineer 1 I've got object concept as well as operation concept because..errm... it's called returned mail...errm...and of course none of the mail is being sent at all, right, it's not just the names in error that aren't recieving mail xxx xxxx xxxx. None of it was sent and I think that's an object concept problem.

Software Engineer 3 Yes, it's a concept if you view a distribution list as a way of getting it to loads of people.

Human Factors Engineer 1 Yeh,

Software Engineer 3 Otherwise.......

Software Engineer 3 Operation symbol.Well I put symbol because the user was unfamiliar with dragging.

Human Factors Engineer 1 I put several things, it's an operation something. I've put operation symbol.

Software Engineer 3 Yes, I should have put operation symbol going by this description. It was the end of the day (general laughter).

Human Factors Engineer 1 I think it's an operationsymbol problem 'cause she's saying she wants a move command and there is one. So there's a symbol problem.

Learning Products Engineer It just seems, it's just fundamentally misunderstood (general laughter).

Human Factors Engineer 1 Well she knew she wanted to move so she wanted to move so it's a concept problem.

Learning Products Engineer Well it's not even in list view, the thing's in perfectly iconic mood, it just that she doesn't know about dragging. I've.....

Human Factors Engineer 1 It's also a concept problem that we could remidy through teaching.

Learning Products Engineer Yes, I think so, it's a matter of.........

Human Factors Engineer 1 Well that seems to have come to an end.

Appendix 8:

The Second Study of ECM: Instructions and Questionnaire for Independent Judges

---

Name: ______________________________

Age: ______________________________

Sex: ______________________________

Present Position: ______________________________

Former Positions: ______________________________

Instructions.

Attached to this questionnaire you should find a transcript of a conversation. This is a transcript of a group of engineers attempting to classify problems that user's experienced with a software package using the ECM scheme.

Your task is to read the transcript of the discussion for each problem. First you should refer to the problem description that the engineers were discussing, which is shown in this questionnaire. Then you should read through the transcript and mark with a ***red*** ped any utterance where, in your opinion, the speaker is disagreeing with another speaker's classification or understanding of the problem. Please mark with a ***blue*** pen any utterance where, in your opinion, the speaker is indicating that he or she may be changing their view of the problem or classification of the problem.

After you have read, and marked, the transcript for any problem you should then answer the questions for each problem in this questinnaire, by ticking the relevant boxes.

To reiterate: the order in which you proceed is as follows:

1 Read the description of problem 1 provided in this questionnaire

2 Read the transcript of the discussion of problem 1 and mark those utterances where the speaker idicates disagreement with a ***red*** pen, and those utterances where the speaker indicates a possible change of opinion with a ***blue*** pen.

3 Anser the questions for problem one, provided in this questionnaire, by ticking the relevant boxes.

4 Read the description of problem 2, and so on...

If you have any problems, or if anything is not completely clear then please ask.

Questionnaire.

1 ___

SITUATION: The user was attempting to select the "create a new" command from the "file" menu.

PROBLEM: The user complained about the menu selection, stating that he did not like having to hold down the mouse button. He said he would prefer to click on a menu and for it to stay open until he clicked on the command he wanted.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

The engineers managed to agree a common understanding of the problem.

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

SITUATION: The user was looking at the incoming mail.

PROBLEM: The user complained that he wanted envelopes opened for him and that he was not altogether keen on things arriving in envelopes at all.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a common understanding of the problem.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

SITUATION: The user was trying to create a new Post-it.

PROBLEM: The user looked under the "items" menu and then explained that he was looking for the "create a new" command.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a common understanding of the problem.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

SITUATION: The user opened a Post-it with the intention of adding text.

PROBLEM: The user began looking under the menus and explained that he expected to have to select an edit mode from the "edit" menu before he would be able to add text.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a common understanding of the problem.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

SITUATION: The user was attempting to send a CV and a post-it together.

PROBLEM: The user had to be told that the way to send the two together was to put them both in an envelope.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a common understanding of the problem.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

SITUATION: The user had placed a memo and a CV in an envelope.

PROBLEM: The user stated that he would like to be able to number the different items so that the receiver would know which to read first.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a common understanding of the problem.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

SITUATION: The user had placed a memo and a CV in an envelope.

PROBLEM: The user said that he would like to be able to 'tag' memos "urgent" in some cases.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

The engineers managed to agree a common understanding of the problem.

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

SITUATION: The user had just successfully mailed a memo.

PROBLEM: The user stated that he expected the memos to have been automatically filed for later reference.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a common understanding of the problem.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

SITUATION: The user was attempting to select a command from a menu.

PROBLEM: The user double clicked the menu head several times in an attempt to open it. She did not remember, until told, that she should hold the mouse button down to kept the menu open.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a common understanding of the problem.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

SITUATION: The user needed to add names to the distribution list.

PROBLEM: The user had the envelope window open, but reported that she did not know how to open the distribution list. In other words; she did not know that it could be opened in the same way that other objects could.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

The engineers managed to agree a common understanding of the problem.

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

SITUATION: The user had just created an envelope and needed to enter the names of those to whom the memo was to be sent.

PROBLEM: The user stated that she wanted to address the envelope by writing on its front - not by creating a distribution list.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a common understanding of the problem.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

SITUATION: The user had added text to a post-it and was closing the window of the post-it.

PROBLEM: When the "save changes?" window appeared the user commented that she thought the changes would be saved automatically and that the window was unnecessary.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

The engineers managed to agree a common understanding of the problem.

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

SITUATION: The user had the mailroom window open.

PROBLEM: After the user was told how to recover items in the mailroom (by dumping the contents of the mailroom to the wastepaper basket) she commented that she would prefer a method which was "easier" and did not entail all of the mailroom contents being removed.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

The engineers managed to agree a common understanding of the problem.

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

SITUATION: The user needed to copy the contents of a distribution list to a distribution list of an envelope.

PROBLEM: The user stated that she had no idea of how to do this.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a common understanding of the problem.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

SITUATION: The user had the mailroom window open.

PROBLEM: The user complained that although it was shown how many messages were waiting to go, she would have liked to know which messages were waiting to be transferred.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

The engineers managed to agree a common understanding of the problem.

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

SITUATION: The user was editing within the distribution list windows.

PROBLEM: The user stated that she would like to be able to edit the list without having to move the cursor from one box to another.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a common understanding of the problem.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

SITUATION: The user had created a post-it and already had a distribution list.

PROBLEM: The user tried to put the distribution list on top of the post-it. She explained that she thought that it might be a short-cut, rather than creating a new envelope or placing the memo in the mailroom.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a common understanding of the problem.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

SITUATION: A mail transfer failed and the user was informed that it was because one of the names on the distribution list was incorrect.

PROBLEM: It was explained to the user that when the transfer failed then the distribution list and memos would be returned. The user commented that she would prefer address/name checking to take place before the transfer.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

The engineers managed to agree a common understanding of the problem.

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

I agree with this statement ☐
I do not agree with this statement ☐
Not sure ☐

SITUATION: A mail transfer failed and the user was informed that it was because one of the names on the distribution list was incorrect.

PROBLEM: The user stated that she did not like the idea that the whole list would be returned to her and she suggested that only those names/addresses which had failed should be returned.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a common understanding of the problem.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

SITUATION: The user needed to move an object from the filing cabinet to the desktop. The filing cabinet window was open.

PROBLEM: The user reported that she did not know how to move the object to the desktop and that she would like a "move" command.

After you have read the transcript for the problem then please indicate your agreement or disagreement with the following statements by ticking the relevant boxes:

The engineers managed to agree upon a classification.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a common understanding of the problem.

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

The engineers managed to agree a solution to the problem (ie a way in which they would change the system).

- I agree with this statement ☐
- I do not agree with this statement ☐
- Not sure ☐

Thankyou for your time and effort.

Appendix 9

The Second Study of ECM: The Subjects' Classifications.

Subjects classified the causes or elements of each of the 20 problems described in the problem description and classification questionnaire (see appendix 5). These classifications, or partical classifications are provided in this section.

1			operation concept (2)	operation concept		operation concept	operation concept
	operation symbol	operation symbol			operation symbol		
2	operation concept	operation concept	operation concept	operation concept	operation concept	operation concept	operation concept
	object concept	object concept	object concept	object concept	object concept	object concept	object concept
3	operation symbol	operation symbol	operation symbol	operation symbol	operation symbol	operation symbol	operation symbol
4	operation concept	operation concept	operation concept	operation concept	operation concept	operation concept	operation concept
				operation symbol			
5	operation concept	operation concept	operation concept	operation concept	operation concept	operation concept	operation concept
	object concept			object concept		object concept	
		object symbol		object symbol			

6	operation concept	operation concept	operation concept	operation concept	operation concept	operation concept	
	operation symbol	operation symbol		operation symbol		operation symbol	operation symbol
	object concept		object concept	object concept		object symbol	
					object symbol		
7		operation concept		operation concept		operation concept	
	object concept	object concept	object concept	object concept	object concept		object concept
8	operation concept	operation concept	operation concept	operation concept	operation concept	operation concept	
	object concept			object concept			object concept
9	operation concept	operation concept	operation concept	operation concept	operation concept	operation concept	operation concept
	operation symbol	operation symbol			operation symbol		
	object concept					object concept	
		object symbol					

10 operation concept — operation concept

operation symbol — operation symbol

object symbol — object symbol — object symbol

---

11 operation concept — operation concept

operation symbol

object concept — object concept — object concept — object concept — object concept

object symbol — object symbol

---

12 operation concept — operation concept (2) — operation concept — operation concept

operation symbol

object concept — object concept — object concept

---

13 operation concept — operation concept — operation concept — operation concept — operation concept — operation concept

operation symbol — operation symbol — operation symbol

object concept — object concept

14	operation concept	operation concept	operation concept	operation concept	operation concept	operation concept	operation concept
		object concept		object concept		object concept	
15				operation concept			
	object concept			object concept	object concept	object concept	
	object symbol		object symbol				object symbol
16	operation concept		operation concept	operation concept		operation concept	
					operation symbol		operation symbol
17	operation concept			operation concept	operation concept	operation concept	operation concept
	object concept (2)		object concept	object concept			
18	operation concept		operation concept	operation concept	operation concept	operation concept	operation concept
	object concept						
19	operation concept		operation concept	operation concept	operation concept	operation concept	operation concept
	object concept						object concept
20	operation concept		operation concept			operation concept	operation concept
	operation symbol (2)		operation symbol	operation symbol	operation symbol	operation symbol	

Appendix 10:

2nd Study of ECM: Designer's Replies in the Interview.

The interview was based on the questionnaire shown in appendix 6. Software Engineer 3 was not available for interview and so is not included in these results. Furthermore, questions 4 and 6 of the questionnaire shown in appendix 6 were not considered appropriate under the circumstances, and so they were not put to the subjects (designers).

Question	Human Factors Engineer	Human Factors Engineer 2	Software Engineer 1	Software Engineer 2	Technical Writer	Senior Learning Products Engineer
1	Quite Well	Quite Well	Quite Well	Quite Well	Quite Well	Quite Well
2	Quite Easy	Not Particularly	Not Particularly	Hard	Not Particularly	Moderately Easy
3	Quite Easy	Quite Easy	Quite Easy	Quite Easy	Quite Easy	Quite Easy
5	Possibly	Possibly	Quite Likely	Quite Likely	Possibly	Possibly
7	Almost All	Almost All	Almost All	Almost All	Almost All	Almost All
8	Most Certainly	Quite Likely	Quite Likely	Possibly	Quite Likely	Most Certainly
9	Solutions Creative	Solutions Creative	Solutions Creative	Solutions Creative	Solutions Creative	Solutions Creative
10	Quite	Quite	Very	Quite	Quite	Very
11	Solutions Creative	Solutions Creative	Solutions Creative	Solutions Creative	Solutions Creative	Solutions Creative
12	Quite	Moderately	Quite	Moderately	Quite	Quite
13	Some People	Only a few	Only a few	Only a few	Only a few	Depends on HF Programme
14	For some testing	For some testing	For some testing	All testing	Most testing	For some testing

Appendix 11:

2nd Study of ECM:
Judge's Replies to their Questionnaire.

The questionnaire that was given to the five independent judges can be seen in appendix 8.

Problem	Judge 1	Judge 2	Judge 3	Judge 4	Judge 5
Statement: The engineers (designers) managed to agree upon a classification.					
1	Not Sure	Yes	Yes	Yes	No
2	Yes	Yes	Yes	Yes	Not Sure
3	Yes	Yes	Yes	Yes	Yes
4	Not Sure	Yes	Yes	Not Sure	Not Sure
5	Yes	Yes	Yes	Not Sure	Not Sure
6	Not Sure	Not Sure	No	No	No
7	Yes	Yes	No	Yes	Not Sure
8	Yes	Yes	Not Sure	No	No
9	Yes	Not Sure	No	Yes	Not Sure
10	Yes	Not Sure	No	Not Sure	Yes
11	Yes	Not Sure	No	Yes	Not Sure
12	Yes	Not Sure	No	Not Sure	No
13	Yes	Yes	No	No	No
14	Yes	Not Sure	Yes	Yes	Yes
15	Not Sure	Not Sure	No	Not Sure	Not Sure
16	Not Sure	Not Sure	No	No	No
17	Not Sure	Not Sure	No	No	Not Sure
18	Yes	Yes	Yes	No	No
19	Not Sure	No	Yes	No	No
20	Yes	No	No	Not Sure	Yes

Problem	Judge 1	Judge 2	Judge 3	Judge 4	Judge 5

Statement: The engineers (designers) managed to agree a common understanding of the problem.

1	No	Yes	Yes	Yes	No
2	Yes	Yes	Yes	Yes	No
3	Yes	Yes	Yes	Yes	Yes
4	Yes	Yes	Yes	Not Sure	Not Sure
5	Yes	Yes	Yes	No	No
6	Yes	Not Sure	No	Not Sure	No
7	Yes	Yes	Yes	No	Not Sure
8	Not Sure	Yes	Not Sure	Not Sure	No
9	Yes	Not Sure	No	Not Sure	Not Sure
10	Yes	Not Sure	Yes	Not Sure	Yes
11	Yes	Yes	Yes	Yes	Not Sure
12	Not Sure	Not Sure	Not Sure	Not Sure	No
13	No	Yes	No	No	No
14	Yes	Not Sure	Yes	Yes	Yes
15	Not Sure	Not Sure	Not Sure	Not Sure	Not Sure
16	No	Not Sure	Not Sure	Yes	No
17	Not Sure	Not Sure	Not Sure	Yes	Not Sure
18	Yes	Yes	Yes	Yes	No
19	Not Sure	Yes	Yes	Yes	No
20	Yes	Yes	Not Sure	Yes	Yes

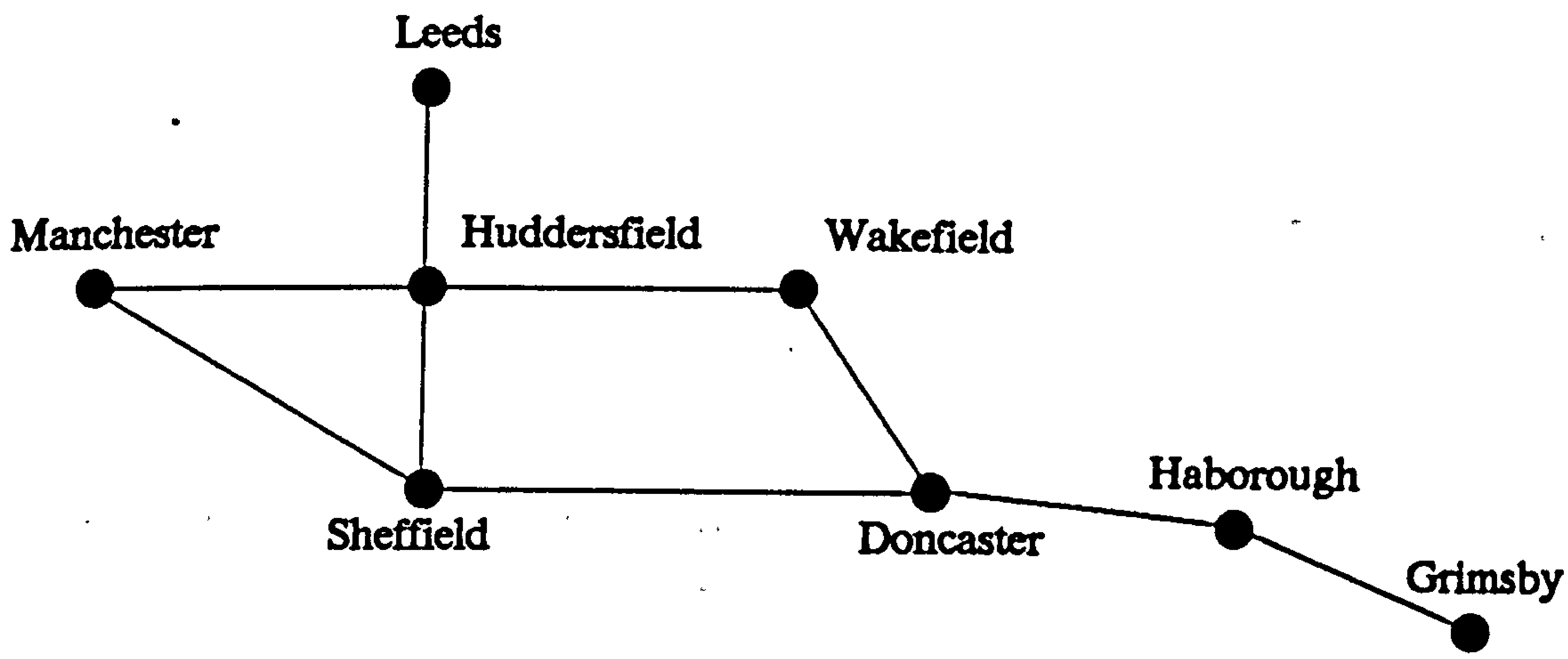
Problem	Judge 1	Judge 2	Judge 3	Judge 4	Judge 5

Statement: The engineers (designers) managed to agree a solution to the problem (ie a way in which they would change the system).

1	No	Not Sure	No	No	No
2	No	Yes	Not Sure	Yes	Not Sure
3	No	Yes	No	No	No
4	Not Sure	Not Sure	No	No	Not Sure
5	No	Not Sure	No	No	No
6	No	Not Sure	No	No	No
7	Not Sure	Yes	No	No	No
8	No	Yes	No	No	No
9	Not Sure	Yes	No	No	No
10	Not Sure	Not Sure	Yes	Not Sure	Yes
11	Yes	Yes	Not Sure	Yes	Not Sure
12	Not Sure	Not Sure	No	No	No
13	No	Not Sure	No	No	No
14	No	Not Sure	No	Not Sure	Not Sure
15	No	Not Sure	No	No	No
16	No	Not Sure	No	No	No
17	Not Sure	Not Sure	No	No	Not Sure
18	Not Sure	Yes	No	Yes	No
19	No	Not Sure	No	No	No
20	Yes	Yes	Not Sure	Yes	Yes

Appendix 12:

Third ECM Study: Instructions to Subjects



The train system shown in the diagram above runs according to the timetables and prices in the system you are about to use. You will be given a series of enquiries from passengers. Your task is to find the information they need using the system.

The enquiries are shown overleaf. You should you shoyuld write your answers below these enquiries in the spaces provided. Please try to be a quick and as accurate as possible.

1 Mr X would like to leave Manchester for Doncaster before 2pm. What is the departure time of the last train before 2pm?

2 Ms Y would like to know the price of a ticket from Leeds to Sheffield.

3 Mr W would like to know how long the journey from Manchester to Haborough takes.

4 Mr B is travelling from Doncaster to Manchester. He would like to arrive in Manchester before 1pm. Which train should he catch?

5 Mr Z would like to know the time of the first train from Wakefield to Sheffield.

He would also like to know the cost of the tickets for himself, his two children and his mother, who is a pensioner (children = half price, pensioners = two thirds price).

6 Ms C would like to travel from Huddersfield to Manchester to arrive before 10am. Which train should she catch?

After this, Ms C would like to catch the train to Grimsby to arrive just before 8pm. Which train should she catch?

Appendix 13:

The Third ECM Study: Problem Descriptions

After viewing the system and considering the tasks the users were asked to perform, together with their comments, could you please consider the difficulties the users experienced. If you have been given a short summary of the ECM classification technique then you should use this to analyse the problems. If you have not been given this summary, then could you consider the problems as you might normally.

After you have analysed the problems, whether you have been asked to use ECM or not, could you then draw up a list of recommended changes to the system.

TASK 1

User 1 commented that he expected to have to press either A, B or C to move to the next stage, he did not expect to have to press return as well.

User 2 commented that she did not know whether she had to type the names of the options, or just A, B or C etc.

User 2, after entering the departure point and destination, received a message stating that she had mis-spelt either the departure point or destination. She complained that she had spelt them correctly, and that she did not know the abbreviations that were now shown on the screen.

TASK 2

User 2 complained that she couldn't remember the abbreviations.

TASK 3.

User 1 complained that once he had mis-spelt either the departure point or the destination he was made to start all over again instead of being allowed to amend just that one he had mis-spelt.

TASK 4.

User 2 commented that she could not see a use for the system listing 1st, 2nd, 3rd, 4th etc train, she only needed the departure times.

User 2 commented that the calculation of the arrival time was difficult.

TASK 5.

User 2 commented that she found herself constantly checking that she had got the departurepointright.

TASK 6.

User 2 commented that the calculation of the half and two-thirds prices was "a pain".

User 2 commented "Why do I have to remember so much, I'm worried I'll forget where I'm up to".

Appendix 14:

The Third ECM Study: Usability Questionnaire.

This questionnaire is intended to allow you to express your feelings about the train timetable and prices system that you have just used. In this questionnaire a number statements are made about the train system. Your task is to indicate you agreement or disagreement with each statement by marking the scale with a cross at the appropriate point. For example, if you disagree strongly with a statement you might mark the scale as follows:

Disagree						Agree
-3	-2	-1	0	+1	+2	+3
X						

+3 and -3	indicate full agreement/disagreement.
+2 and -2	indicate moderate agreement/disagreement.
+1 and -1	indicate slight agreement/disagreement.
0	indicates a neutral position.

If you do not understand exactly what you have to do then please ask the experimenter.
Thankyou for your time and effort.

---

1 The system is easy to use.

Disagree						Agree
-3	-2	-1	0	+1	+2	+3

---

2 The system was difficult to learn.

Disagree						Agree
-3	-2	-1	0	+1	+2	+3

3 I am never sure exactly what the system is doing when I use it.

Disagree						Agree
-3	-2	-1	0	+1	+2	+3

---

4 I have to concentrate hard to use the system.

Disagree						Agree
-3	-2	-1	0	+1	+2	+3

---

5 I did not feel in control of the system.

Disagree						Agree
-3	-2	-1	0	+1	+2	+3

---

6 The system takes too long to use.

Disagree						Agree
-3	-2	-1	0	+1	+2	+3

---

7 The system is confusing.

Disagree						Agree
-3	-2	-1	0	+1	+2	+3

8 The system needs improvement.

Disagree						Agree
-3	-2	-1	0	+1	+2	+3

9 The system is frustrating to use.

Disagree						Agree
-3	-2	-1	0	+1	+2	+3

10 The system did all that I wanted it to.

Disagree						Agree
-3	-2	-1	0	+1	+2	+3

11 The system is more complicated than it needs to be.

Disagree						Agree
-3	-2	-1	0	+1	+2	+3

12 The system allowed me to correct my mistakes easily.

Disagree						Agree
-3	-2	-1	0	+1	+2	+3

13 I felt happy using the system.

Disagree						Agree
-3	-2	-1	0	+1	+2	+3

Appendix 15:

The Third ECM Study: The Design Changes.

These recommendations originate from the recommendations of the four designers. Two designers used ECM to produce their recommendations, while the other two did not. Each set of design recommendations was compiled by an independent judge. These recommendations were then handed to the author who implemented the changes, without knowing which set had been produced using ECM and which set had not.

The non-ECM Recommendations.

1 One the first menu the options, A, B, and C, should be in capitals, not lowercase.

2 There should be a short explanation of which buttons need to be pressed (including return) for the first menu.

3 The main menu should be labeled "Main Menu".

4 Theabbreviations for the stations should all be three letters long and should be displayed on the screen each time the user has to use them.

5 Each entry for the user should be checked for errors. The user should be allowed to retype just that entry which is incorrect, and should not be simply returned to the main menu.

6 The error message should not state that the user has mis-spelt the station's name, as this is not always the case.

7 The trains should not be referred to as 1st, 2nd & 3rd etc, this should be removed.

8 The arrival times as well as the departure time should be shown.

9 Children's and pensioner's prices should be displayed as well as adult fares.

The ECM Recommendations.

1 The first menu should allow the user the choose the stations (departure point and destination) he or she needs using just one key press. Furthermore, the user should not have to press return.

2 The first screen should explain which keys relate to which stations, and about what will happen once the keys are pressed.

3 All of the information should be displayed on one screen if possible, there should not be the different options that presently exist.

4 Use normal symbols for pounds (ie £).

5 Each entry for the user should be checked for errors and the user should be allowed to retype just that entry which is incorrect.

6 The error message should not state that the user has mis-spelt the station's name, as this is not always the case.

7 The trains should not be referred to as 1st, 2nd & 3rd etc, this should be removed.

8 The arrival times as well as the departure time should be shown.

9 The names of the stations should appear at the top of the screen when information is presented, and also above the departure and arrival times.

10 Children's and pensioner's prices should be displayed as well as adult fares.

Appendix 16:

Published Material

AN INVESTIGATION INTO BUSINESS INFORMATION PRESENTATION AT HUMAN-COMPUTER INTERFACES

In: H. J. Bullinger & B. Shackel (Eds.), *Human-computer interaction - Interact '87: Proceedings of the second IFIP conference on human-computer interaction, Stuttgart.* Amsterdam, Netherlands: North-Holland.

Paul Booth, Chris Fowler & Linda Macaulay

The HCI Research Unit,
Huddersfield Polytechnic, HD1 3DH, England.

Many commercial software packages aimed at the business community advertise graphical data presentation as a selling point. However, the general growth in the use of information display techniques in I.T. systems has not been matched by a proper consideration of their effects and at the moment, the evidence regarding display format preference is somewhat contradictory. An experiment is reported which was concerned with information display format (i.e: graphical and tabular) preference at the human-computer interface. Subjects from both business and non-business backgrounds performed a decision-making task after completing the visualiser-verbaliser and conceptual tempo cognitive style tests. The results suggested that the visualiser-verbaliser and conceptual tempo cognitive style dimensions are unlikely to be of use in predicting display format preference at the human-computer interface. Moreover, there were indications that individuals will use information in both graphical and tabular formats regardless of their own stated preferences. Nevertheless, as stated display format preferences were widely dispersed, it is suggested that users of computer systems should be allowed to choose the format in which information is presented.

1. INTRODUCTION

It has been suggested in the literature that cognitive style may be of use in predicting some aspects of user behaviour and preference at the human-computer interface [22, 6]. The purpose of the experiment reported here was to examine the effects of cognitive style, measured on two dimensions, upon display format preference and information use during a decision-making task. In the following sections summaries of the two cognitive style dimensions used in the experiment are presented, but first the rationale underlying the experiment is explained.

1. 1. Rationale

The general trend of presenting more information at the human-computer interface in a graphical format is particularly noticeable in many business oriented software packages. The effects of different data presentation formats however, are unclear [24] and the results of studies that have already considered this problem do not agree. Benbasat & Schroeder [3] found that graphically displayed information was preferred by operations management students in an experiment where they had to play the role of a warehouse manager. Lucas [13] on the other hand, discovered that subjects in a study where they were required to act as a buyer for a company, found data presented in a tabular format more helpful. One possible explanation for these apparently contradictory results is that the usefulness of a particular display format may be dependent upon the perceived requirements of a particular task or situation. Powers *et al.* [18] for example, have demonstrated that tabular displays increase the speed of performance while combinations of both graphical and tabular displays slow performance, but increase accuracy and comprehension on the part of the subject. Nevertheless, it remains difficult to reconcile the conflicting evidence regarding the usefulness of graphical and tabular display formats and alternative explanations (i.e: that preferences may be related to cognitive style) have not been fully explored.

It was hoped that the experiment would help to disambiguate the situation by demonstrating, if possible, that display format preferences could be related to subjects' positions on the visualiser-verbaliser cognitive style dimension. The possibility of using cognitive style to predict format preferences was prompted originally by Fowler *et al.* [6] who suggested that, "...cognitive style is an important concept in relation to human-computer interaction, and may be of considerable use in the area of interface design". For this study, it was hypothesised that verbalisers would prefer and use information displayed in a tabular format while visualisers were expected to be inclined to favour information displayed in a graphical format. A further possibility was that individuals may not only prefer information in one format or another, but might pay more attention to information in this format while performing a task. Moreover, the extent of this effect might be related to how individuals scan information. Some people search carefully through information (reflective individuals) while others are more impulsive and reach their decisions in a shorter space of time, often without considering all of the information. Therefore, it is suggested that those individuals classified as impulsive on the conceptual tempo (impulsivity-reflectivity) cognitive style dimension may only consider information displayed in their predicted format while reflective subjects might consider information in both predicted and non-predicted formats (predicted by the individual's position on the visualiser-verbaliser cognitive style dimension). However, before we go further, it may be useful not only to consider what the visualiser-verbaliser and conceptual tempo dimensions are, but also to briefly outline the concept of cognitive style.

1. 2. Cognitive style

According to Messick [15] cognitive styles are "...high-level heuristics that organise and control behaviour across a wide variety of situations". These heuristics produce,

"...consistent individual differences in [the] organisation and processing [of] information", [15]. An individual's cognitive style is represented on a set of bipolar dimensions which are thought to refer to differences in the selection of information processing strategies. Two of these dimensions, the visualiser-verbaliser and conceptual tempo dimensions, are the subject of the experiment.

Measurement of individuals' positions on the visualiser-verbaliser cognitive style dimension is based upon the visualiser-verbaliser questionnaire developed by Richardson [20]. Richardson used eye movement tendencies [11, 12] to pick out discriminating questions on Paivio's [17] 'ways of thinking' questionnaire to produce the VVQ (visualiser-verbaliser questionnaire). This is a set of fifteen questions which are intended to discriminate individuals who favour either, "...verbal or visual strategies when processing many different kinds of information", [20].

The conceptual tempo cognitive style dimension was first suggested by Kagan *et al.* [10] and is concerned with the behaviour of impulsive and reflective individuals in decision-making tasks. Kagan [9], who proposed the MFF (matching familiar figures) test of conceptual tempo, states that impulsives "...report the first hypothesis [answer] that occurs to them, and this response is typically incorrect. The reflective subject on the other hand, delays a long time before reporting a solution hypothesis and is usually correct". Differences between impulsive and reflective individuals on the MFF test have been associated with subjects' scanning strategies [5] and this is thought to be related to differences in 'cognitive control' between individuals [15].

These two dimensions were expected to influence stated display format preference and information use during the experimental task. However, other factors such as subjects' experiences of knowledge domains relevant to the task might also influence display format preference and information use.

1. 3. Business knowledge

The possibility arose that subjects with experience in business, through their training and experience, may have inadvertently been taught to deal with information in a particular format, No evidence for this view had been found in the literature, but still the possibility was considered important enough to warrant serious consideration. Therefore, in order to observe these effects, should there have been any, subjects in the experiment were not only tested to determine their cognitive style, but were also selected from both business and non-business backgrounds.

1. 4. Hypotheses

1 Subjects' stated information display format preferences should reflect their score on the visualiser-verbaliser cognitive style test (i.e: a visualiser for example should prefer information in a graphical or related format while a verbaliser is expected to prefer data in a tabular format).

2 When explaining their decisions during the experimental task, subjects will refer more often to information in the format predicted by their score on the visualiser-verbaliser questionnaire.

3 For impulsive subjects, as determined by the conceptual tempo cognitive style test, the effects predicted in hypotheses one and two will be greater than the average, while reflective subjects should show a lesser effect.

2. METHOD

The study was based around a pencil and paper type task which lasted from one and a quarter hours to two hours per subject (including administering the cognitive style tests).

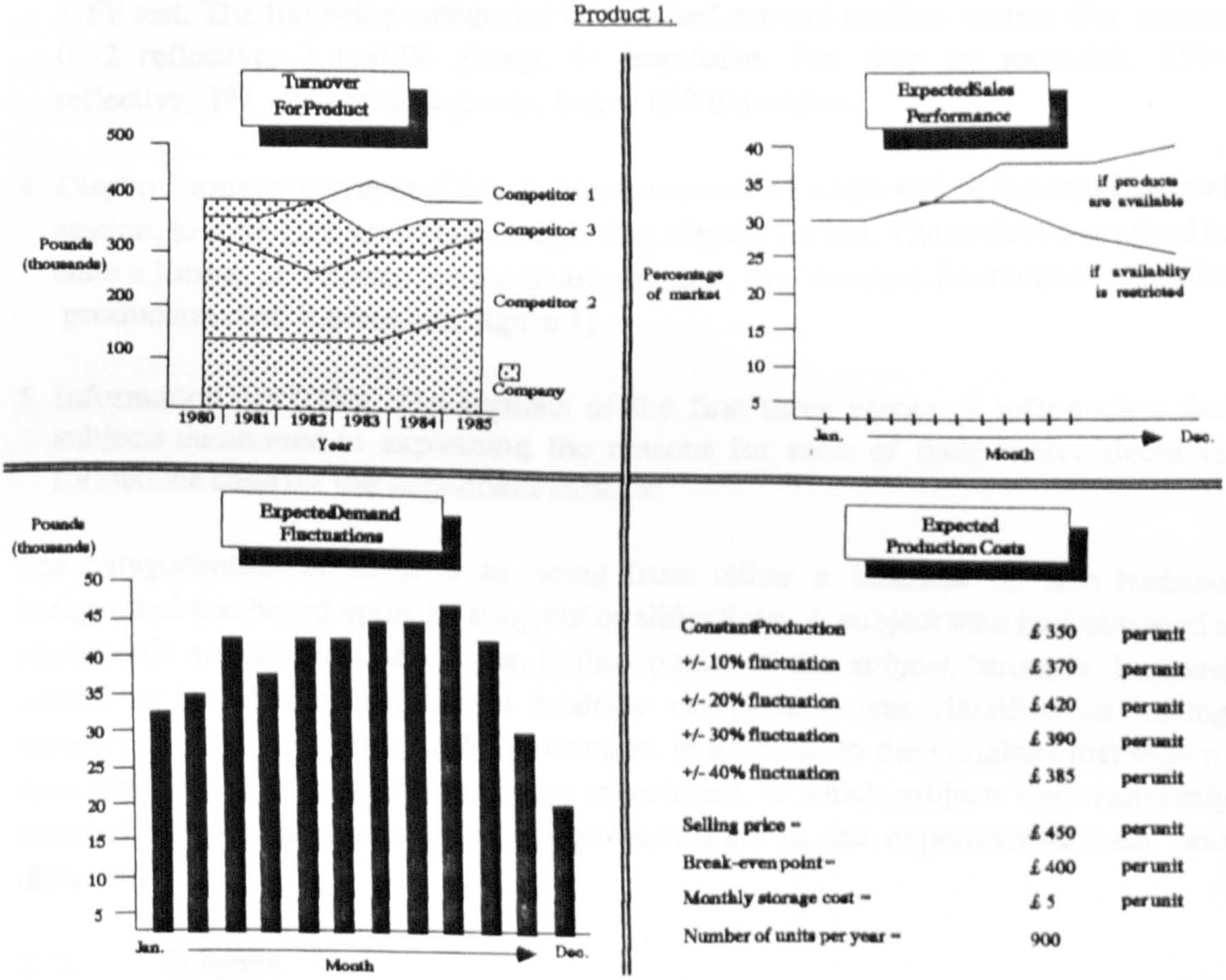


Figure 1: One of the presentations from the experimental task.

2. 1. Design

A mixed model design was adopted. The experiment was designed to reveal the effects of different display formats (within subjects) together with different psychological types (between subjects) upon display format preference and information use during performance at a decision-making task. The variables were as follows:

1 Display format (IV): Each of the twelve presentations involved eight pieces of information. One of these pieces of information was always presented in the same graphical format while a further three were varied between graphical and tabular formats. The remaining four pieces of information were presented as single figures towards the bottom of each presentation (see figure 1).

2 The visualiser-verbaliser cognitive style dimension (IV): The classification of individuals on this dimension was was based upon their VVQ scores as follows: 0 - 6 verbal, 7 - 8 middle group, 9 - 15 visual. These categories were based around the median score for all thirty subjects.

3 The conceptual tempo cognitive style dimension (IV): This was assessed using the MFF test. The following categories were based around median scores: For errors; 0 -2 reflective, 3 middle group, 4+ impulsive. For time (in seconds); 950+ reflective, 891 - 949 middle group, below 890 impulsive.

4 Display format preference (DV): Subjects were asked, at the end of the experimental session, to state their preferences regarding display format. The subject was asked to state a format preference for the 'market share', the 'demand fluctuations' and the 'production cost' figures (see figure 1).

5 Information use (DV): The formats of the first three pieces of information that subjects mentioned in explaining the reasons for each of their twelve decisions formed the basis for this dependent variable.

The categorisation of subjects as being from either a business or non-business background was based upon the subjects' qualifications. A subject who had obtained a educational qualification which was, in the opinion of the *subject,* 'strongly business related' or included a 'substantial business component' was classified as having experience of the business knowledge domain. In addition to the variables just shown, there were two further conditions in the experiment, to which subjects were randomly assigned, where the ordering of the presentations in the experimental task was different.

2. 2. Subjects

A total of thirty volunteer subjects were used for the experiment, twenty eight of whom were staff at the Polytechnic. The remaining two were managers from two local firms. The ages of subjects were as follows: 2 between 20 & 30 years; 9 between 31 & 40 years; 12 between 41 & 50 years; 7 over 50 years. Of these subjects twelve had business experience. Five of the subjects were female. There were no significant associations between age, sex or qualification and position on either of the two cognitive style dimensions used in the experiment.

2. 3. The experimental task

The experimental task involved choosing between three possible production strategies for an imaginary product using the information provided in the presentation. This decision-making task was checked by an associate with business qualifications and experience to ensure that the terms and concepts used closely resembled the terms and concepts commonly used in business and that the task was meaningful from an expert point of view[1]. Subjects could choose between fluctuating production, constant production with storage or constant production without storage (the results regarding subjects decisions are beyond the scope of this paper and will not be reported here). There were in total, twelve presentations for twelve imaginary products. The figures for these products were based upon three original sets of figures. The additional sets of figures were produced by multiplying all of the figures in the originals by constants so that the absolute figures changed while the proportionate differences between figures and the relationships between different pieces of information did not alter. An example of one of the presentations can be seen in figure 1. Display format was varied to produce four conditions; the first involved the expected sales figures and the expected demand fluctuation figures in a graphical format while the expected production costs were displayed in a tabular format. The second involved the sales and demand figures in a tabular format while the production figures were displayed in a graphical format. The third involved the sales and production figures in a tabular format and the demand figures in a graphical format while the fourth involved the demand and the production figures in a graphical format while the sales figures were presented in a tabular format.

The purpose of the experiment was to encourage subjects to make 'best guess' decisions for a complex task and to observe the effects of display format upon business and non-business subjects with differing cognitive styles. In the pilot studies some subjects deliberated, in the most extreme case, for over twenty minutes per presentation and explained that they were attempting to calculate the possible financial outcomes of the three possible production strategy options while others quickly produced the 'best guess' decision that was expected of them. If some subjects were attempting to calculate financial outcomes while others were making 'best guess' decisions then this was inadvertently introducing a further condition into the experiment. In order to control the experiment and to ensure that all subjects were making 'best guess' decisions, a time limit of one minute for the decision regarding each presentation was imposed. Most subjects were able to reach their decisions well within this time.

2. 4. Procedure

All instructions were tape recorded, although subjects were allowed to ask questions of the experimenter. Firstly, the subject filled in a one-page biographical questionnaire after which the visualiser-verbaliser and conceptual tempo cognitive style tests were administered. Following this, subjects were told to imagine themselves as management executives whose task it was to choose a production strategy for each of their company's twelve products. Subjects were shown figures for three practice products (tasks) where they could ask for further explanation. The experimental task was then performed for each of the twelve presentations. After each decision subjects were asked to explain their choice of production program and the type of information mentioned by the subject was noted. At the end of the experiment subjects were asked to state their information presentation preferences for each type of information used in the experimental task and were then debriefed.

3. RESULTS

3. 1. The results relating to the hypotheses

3. 1. 1. Hypothesis 1: Display format preference and the visualiser-verbaliser dimension

Tabular information was preferred by visualisers 11 times (n = 14) and by verbalisers 10 times (n = 12). Chi squared for a 1 x 2 table = 0.02 (df = 1) NS (not significant). Graphical information was preferred by visualisers 25 times (n = 14) and by verbalisers 21 times (n = 12). Chi squared for a 1 x 2 table = 0.01 (df = 1) NS. Therefore, the first hypothesis, which stated that display format preferences would be related to a subjects position on the visualiser-verbaliser cognitive style dimension, has not been supported.

3. 1. 2. Hypothesis 2: Information use and the visualiser-verbaliser dimension

In total each subject could make up to 18 graphical references, 18 tabular references or alternatively could mention nothing up to 36 times. Tabular information was mentioned by visualisers 88 times (n = 14) and by verbalisers 78 times (n = 12). Chi squared for a 1 x 2 table = 0.02 (df = 1) NS. Graphical information was mentioned by visualisers 111 times (n = 14) and by verbalisers 122 times (n = 12). Chi squared for a 1 x 2 table = 3.37 (df = 1) NS. The second hypothesis has not been supported.

3. 1. 3. Hypothesis 3a: Display format preference and the visualiser-verbaliser and conceptual tempo dimensions

As the MFF test can be scored using either time or errors as the measurement criteria, then analyses are presented for both. If the MFF test is scored by errors; tabular information was preferred by visual impulsives 6 times (n = 6), by visual reflectives 5 times (n = 8), by verbal impulsives 6 times (n = 6) and by verbal reflectives 4 times (n = 6). Chi squared for a 1 x 4 table = 1.02 (df = 3) NS. If the MFF test is scored by time; tabular information was preferred by visual impulsives, 3 times (n = 4), by visual reflectives 4 times (n = 6), by verbal impulsives 6 times (n = 5) and by verbal reflectives 4 times (n = 7). Chi squared for a 1 x 4 table = 3.58 (df = 3) NS. For both of these analyses some of the expected cell frequencies fall below '5' and so the values of Chi squared in this case must be treated with some caution.

If the MFF test is scored by errors; graphical information was preferred by visual impulsives 8 times (n = 6), by visual reflectives 17 times (n = 8), by verbal impulsives 7 times (n = 6) and by verbal reflectives 14 times (n = 6). Chi squared for a 1 x 4 table = 3.53 (df = 3) NS. If the MFF test is scored by time; graphical information was preferred by visual impulsives 5 times (n = 4), by visual reflectives 12 times (n = 6), by verbal impulsives 7 times (n = 5) and by verbal reflectives 14 times (n = 7). Chi squared for a 1 x 4 table = 1.40 (df = 3) NS. None of these results are statistically significant and therefore, with respect to display format preferences, the third hypothesis has not been supported.

3. 1. 4. Hypothesis 3b: Information use and the visualiser-verbaliser and conceptual tempo dimensions

If the MFF test is scored by errors; tabular information was mentioned by visual impulsives 40 times (n = 6), by visual reflectives 48 times (n = 8), by verbal impulsives 52 times (n = 6) and by verbal reflectives 29 times (n = 6). Chi squared for a 1 x 4 table = 7.23 (df = 3) NS. If the MFF test is scored by time; tabular information was mentioned by visual impulsives 25 times (n = 4), by visual reflectives 34 times (n = 6), by verbal impulsives 41 times (n = 5) and by verbal reflectives 40 times (n = 7). Chi squared for a 1 x 4 table = 3.58 (df = 3) NS.

If the MFF test is scored by errors; graphical information was mentioned by visual impulsives 51 times (n = 6), by visual reflectives 63 times (n = 8), by verbal impulsives 72 times (n = 6) and by verbal reflectives 50 times (n = 6). Chi squared for a 1 x 4 table = 7.53 (df = 3) NS. If the MFF test is scored by time; graphical information was mentioned by visual impulsives, 41 times (n = 4), by visual reflectives 43 times (n = 6), by verbal impulsives 52 times (n = 5) and by verbal reflectives 70 times (n = 7). Chi squared for a 1 x 4 table = 4.30 (df = 3) NS. With respect to information use, the third hypothesis has not been supported.

3. 2. Otherresults

3. 2. 1. Information use and display format preference

Although neither display format preference or information use have been found to be related to either of the two cognitive style dimensions, there was still the possibility that individuals might predominantly use information in the format for which they stated a preference, regardless of their cognitive style. Using Spearman's rank correlation for subjects stated preferences and their information use of tabular information r_s = +0.202 (N = 30) NS. For the correlation between the subjects stated preferences and their information use for graphical information r_s = +0.203 (N = 30) NS. It appears as though subjects' stated display format preferences did not determine the information they used during the experimental task to explain their decisions.

3. 2. 2. Display format preference, information use and business subjects

As mentioned in the introduction, it was considered possible that either display format preferences or information use might have been influenced by subjects experience in business. For information preference: Tabular information was preferred by business subjects 11 times (n = 12) and by non-business subjects 16 times (n = 18). Chi squared for a 1 x 2 table = 0.01 (df = 1) NS. Graphical information was preferred by business subjects 19 times (n = 12) and by non-business subjects 32 times (n = 18). Chi squared for a 1 x 2 table = 0.07 (df = 1) NS. For information use: Tabular information was mentioned by visualisers 71 times (n = 12) and by verbalisers 134 times (n = 18). Chi squared for a 1 x 2 table = 2.24 (df = 1) NS. Graphical information was mentioned by business subjects 104 times (n = 12) and by non-business subjects 170 times (n = 18). Chi squared for a 1 x 2 table = 0.40 (df = 1) NS. Experience of the business knowledge domain does not appear to significantly influence information display format preference or information use during the experimentaltask.

3. 2. 3. Display format preference, information use and non-business subjects

If business subjects are excluded from the analysis then the results are as follows: Tabular information was preferred by visualisers 8 times (n = 10) and by verbalisers 6 times (n = 6). Chi squared for a 1 x 2 table = 0.02 (df = 1) NS. Graphical information was preferred by visualisers 18 times (n = 10) and by verbalisers 11 times (n = 6). Chi squared for a 1 x 2 table = 0.02 (df = 1) NS. For information use: Tabular information was mentioned by visualisers 61 times (n = 10) and by verbalisers 41 times (n = 6). Chi squared for a 1 x 2 table = 0.21 (df = 1) NS. Graphical information was mentioned by visualisers 82 times (n = 10) and by verbalisers 55 times (n = 6). Chi squared for a 1 x 2 table = 0.30 (df = 1) NS. If business subjects are excluded from the analysis then all of the results are still non-significant.

3. 2. 4. Informatiorredundancy

At the end of the experiment subjects were asked: 'If, for each set of figures in the presentation, the information had been displayed in both a tabular and a graphical format would you be (a) happy with this situation, (b) unhappy with this situation or (c) do you believe that this is not necessary'. The results were as follows; 4 business subjects and 14 non-business subjects stated that they would be happy with redundant information, 8 business and 4 non-business subjects stated that they thought redundant information unnecessary, while none of the subjects stated that they would be unhappy with information redundancy. The question was asked without the subjects being given any examples of what a presentation with information displayed in both formats might look like and so can only be considered as a rough indication of true preferences.

3. 2. 5. Differences between business and non-business subjects

During the experiment clear differences emerged between subjects with and subjects without experience in business. Typically, the latter appeared to concentrate upon the absolute differences between the figures relating to production and other costs. Business subjects on the other hand, looked for trends and relationships between all of the sets of figures. This subjective account is viewed as tentative support for the idea that subjects with experience in business approached the experimental task in a way which was different from non-business subjects. Further support for this notion was found by analysing crude differences in word use during the experiment by business and non-business subjects when they were explaining their reasons for the decisions. If subjects used either of the words 'market' or 'competitor' more than three times during the experiment then they were classed as having adopted a 'market view'. Of the business subjects, 10 adopted a market view while 2 did not. Of the non-business subjects, 2 adopted a market view while 16 did not. Using Fisher's exact probability test it can be seen that statistically the result is highly significant ($p < 0.0002$), thus supporting, to some extent, the idea that business subjects approached the experimental task in a way which was different from non-business subjects.

4. DISCUSSION

4. 1. Display format preference and information use

The original aim in designing the experiment was to show the effects of the visualiser-verbaliser and conceptual tempo cognitive style dimensions upon information display format preference and information use during a decision-making task. However, before the results and their implications are discussed, it may be useful to consider some of the problems with applying cognitive style to HCI so that the results might be viewed in this context.

One problem with the application cognitive style to HCI relates to the tests used to measure cognitive style. There are no agreed standards for classifying individuals on these measures (with the exception of the juvenile version of the conceptual tempo cognitive style measure and the field-dependent/independent measures where norms are given) and classification tends to be based, in each case, around a group norm. This method of classification is clearly problematic, for the widescale generalisation and application of cognitive style findings depends upon consistent cognitive style measurement and this is not possible without at least some degree of measurement standardisation [7].

The second and more serious difficulty with the application of cognitive style to HCI is related to the lack of an explanatory theory. There is still no theory which adequately explains how different cognitive style dimensions relate to one another, or how cognitive style might account for behaviour at the human-computer interface, where it has already been established that task performance is closely related to an individuals use of relevant knowledge [2]. Witkin & Goodenough [25] suggest that cognitive style measures and field-dependency measures in particular, measure differences in perceptual processes, but do not say, in any detail, how these might relate to performance at a task. Using tests to predict behaviour, where the processes underlying performance on the tests are not fully understood, has been criticised by many as 'methodologically flawed' [e.g: 8].

Despite these difficulties, at the outset of this research it was considered potentially profitable to pursue the possibility of using cognitive style. If cognitive style could have been shown to have an effect upon user preference and information use at the experimental task, then it might have been worthwhile to then go on and consider the problems of measurement and theory in the later stages of the research. The results however, have shown that the measures of the two cognitive style dimensions (the VVQ and the MFF) were of no use in predicting display format preferences or information use.

One possible explanation for these results is that the effects of subjects applying their business knowledge 'swamped' the effects of cognitive style. However, if this was the case we might have expected subjects without experience in business alone to show an effect, but there are no differences between verbal and visual non-business subjects in terms of either their stated display format preferences or their use of information during the experimental task (see section 3. 2. 3). Another explanation is that the task was too complex for cognitive style to show an effect. The results of this experiment cannot be used to either confirm or refute this suggestion. It might be noted however, that many tasks at the human-computer interface are complex and it is suggested that the visualiser-verbaliser and conceptual tempo cognitive style dimensions must be shown to have an effect upon complex as well as simple tasks if they are to be of use in HCI.

To reiterate; for this complex experimental task the visualiser-verbaliser and conceptual tempo cognitive style dimensions were not found to have significantly influenced display format preference. Furthermore, no clear relationship emerged between display format preference and any other measurable factor in the experiment. Display format preferences were widely distributed for both business and non-business subjects and these preferences were not found to effect the information subjects chose to use in explaining their decisions during the experimental task. As a consequence of these results, it might have been proposed that information displayed at the human-computer interface should be presented in more than one format. However, the 'poll' carried outat the end of the experiment indicated that, although no subjects stated that they would be unhappy with the display of redundant information, a number indicated that they did not believe it to be necessary. Therefore, it is suggested that users should be allowed to choose the display format for different pieces of information and if this is not possible then information should be displayed in more than one format.

4. 2. The differences between subjects with and subjects without experience in business

There is already evidence to demonstrate that people with experience of the same knowledge domain share common representations of that domain [1, 23]. Furthermore, there is evidence to show that representation of a particular knowledge domain strongly affects how individuals reason about issues and problems associated with that domain [16, 4, 14, 21]. Therefore, it is not unreasonable to expect people who share a common experience of a particular knowledge domain to behave in a similar fashion when faced with a task associated with that domain. For although disparities in expertise are likely to give rise to differences in behaviour between individuals with a common background [19], these are unlikely to be as great as the differences between those who have and those who do not have an understanding of a particular area of knowledge.

Therefore, according to the literature there should have been marked differences between business and non-business subjects, both in terms of how they talked about the task as well as their overall approach to it. Indeed, this is what was found. Business subjects appeared to consider all of the information in the display while non-business subjects concentrated upon comparisons between one or two sets of figures. Moreover, subjects with experience in business tended to talk in terms of 'markets' and 'competitors' while non-business subjects did not. The contrasting behaviour of those with and those without experience of the business knowledge domain suggests that those with business qualifications shared a common representation of the domain relative to those subjects without qualifications and furthermore, that this representation influenced their behaviour at the experimental task. Moreover, the differences between business and non-business subjects suggest that user groups may differ from one another in their representation of the task as well as from the designer's representation of the task. All of this however, is well known within the field and the observations reported here can only be considered as weak support for what has been accepted within the literature for some considerable time [14, 1, 14, 19, 21, 23].

5. CONCLUSION

The results of this study have suggested that the visualiser-verbaliser and conceptual tempo cognitive style dimensions are unlikely to be of use in predicting display format preference at the human-computer interface. Furthermore, there were indications that individuals will use information in both graphical or tabular formats regardless of their own stated preferences. Nevertheless, as preferences amongst both business and non-business subjects were widely dispersed, it is suggested that users of computer systems should be allowed to choose the format in which information is presented. The differences that were observed during the experiment between business and non-business subjects were expected, given the extensive literature regarding the differences between experts and novices in various knowledge domains. However, these differences do underline the difficulties faced by system designers, who are trying to produce systems for users whose representation of the task is likely to be markedly different from their own.

ACKNOWLEDGEMENTS

The authors would like to thank Ken Eason, Peggy Newton, Gill Brown, Barbara Hurwood, Adrian Castell and Phil Marsden for their comments and suggestions on earlier drafts of this paper.

Thanks are also due to Kee Fan of the Department of Administrative and Management Studies at Huddersfield Polytechnic for his help with respect to the experimental task.

REFERENCES.

[1] Adelson, B., (1984) 'When novices surpass experts: The difficulty of a task may increase with expertise'. Journal of Experimental Psychology: Learning, Memory & Cognition, Volume 10(3), pages 483-495.

[2] Barnard, P., J., & Hammond, N., V., (1982) 'Usability and it's multiple determination for the occasional user of interactive systems'. In Williams (ed) 'Pathways to the information society'. Proceedings of the 6th international conference on computer communication. London, England.

North-Holland.

[3] Benbasat, I., & Schroeder, R., G., (1977) 'An experimental investigation of some MIS design variables'. The Management Information Systems Quarterly, Volume 1, pages 37-49.

[4] Chi, M., T., H., Peltovich, P., J., & Glaser, R., (1981) 'Categorisation and representation of physics problems by experts and novices'. Cognitive Science, Volume 5(2), pages 121-152.

[5] Drake, D., M., (1970) 'Perceptual correlates of impulsive and reflective behaviour'. Developmental Psychology, Volume 2.

[6] Fowler, C., J., H., Macauley, L., A., & Fowler, J., (1985) 'The Relationship between Cognitive Style and Dialogue Style: An Explorative Study.' In: P. Johnson & S. Cook (eds) 'People and Computers: Designing the interface'. Cambridge Press.

[7] Huber, G., P., (1983) 'Cognitive style as a basis for MIS and DSS designs: Much ado about nothing'. Management Science, Volume 29(5).

[8] Johnson-Laird, P., N., (1983) 'Mental Models: Towards a cognitive science of language, inference, and consciousness'. Cambridge University Press: Cambridge, London.

[9] Kagan, J., (1966) 'Reflection-Impulsivity: The Generability and Diagnosis of Conceptual Tempo'. Journal of Abnormal Psychology, Volume 71, pages 17-22.

[10] Kagan, J., Rosman, B., C., Day, D., Albert, J., & Phillips, W., (1964) 'Information processing in the child'. Psychological Monographs, Volume 78.

[11] Kinsbourne, M., (1972) 'Eye and head turning indicates cerebral lateralisation'. Science, Volume 176, pages 539-541.

[12] Kocel, K., Galin, D., Ornstein, R., & Merrin, E., L., (1972) 'Lateral eye movements and cognitive mode'. Psychonomic Science, Volume 27, pages 233-224.

[13] Lucas, H., C., (1979) 'An experimental investigation of the use of graphics in decision-making'. pages 11-17, School of Business, New York University, cited in Powers *et al.* [18].

[14] McKeithen, K., B., Reitman, J., S., Rueter, H., H., & Hirtle, S., C., (1981) 'Knowledge organisation and skill differences in computer programmers'. Cognitive Psychology, Volume 13, pages 307-325.

[15] Messick, S. (ed) (1976) 'Individuality in Learning'. Jossey-Bass, San Francisco.

[16] Newell, A. & Simon, H. A. (1972) 'Human problem solving'. Englewood Cliffs, NJ, Prentice-Hall.

[17] Paivio, A., (1971) 'Imagery and verbal processes'. New York, Holt, Rinehart & Winston.

[18] Powers, M., Lashley, C., Sanchez, P., & Shneiderman, B., (1984) 'An experimental comparison of tabular and graphic data presentation'. International Journal of Man-machine Studies, Volume 20, pages 545-566.

[19] Rector, A., L., Newton, P., Marsden, P., H., (1985) 'What kind of system does an expert need?' In: P. Johnson & S. Cook (eds) 'People and Computers: Designing the interface'. Cambridge Press.

[20] Richardson, A. (1977) 'Verbalisers-Visualisers: A Cognitive Style dimension'. Journal of Mental Imagery, Volume 1, pages 109-125.

[21] Riesbeck, C., K., (1984) 'Knowledge reorganisation and reasoning style'. International Journal of Man-Machine Studies, Volume 20, 45-61.

[22] Robertson, I. T., (1985) 'Human Information Processing Strategies and Style'. Behaviour and Information Technology. Volume 4(1), pages 19-29.

[23] Weiser, M., & Shertz, J., (1983) 'Programming problem representation in novice and expert programmers'. International Journal of Man-Machine Studies, Volume 19, pages 391-398.

[24] Wickens, C., D., & Kramer, A., (1985)' Engineering Psychology'. Annual Review of Psychology, Volume 36, pages 307-348.

[25] Witkin, H. A. & Goodenough, D., R., (1981) 'Cognitive Styles: Essence and origins'. Lawrence Erlbaum Associates.