

This item is held in Loughborough University's Institutional Repository (<https://dspace.lboro.ac.uk/>) and was harvested from the British Library's EThOS service (<http://www.ethos.bl.uk/>). It is made available under the following Creative Commons Licence conditions.



creative  
commons  
C O M M O N S D E E D

**Attribution-NonCommercial-NoDerivs 2.5**

**You are free:**

- to copy, distribute, display, and perform the work

**Under the following conditions:**

 **BY:** **Attribution.** You must attribute the work in the manner specified by the author or licensor.

 **Noncommercial.** You may not use this work for commercial purposes.

 **No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

**Your fair use and other rights are in no way affected by the above.**

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:  
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

**THE EVALUATION OF DYNAMIC  
HUMAN-COMPUTER  
INTERACTION**

**by**

**Neil Stephen Watkinson**

**A Doctoral Thesis submitted in partial fulfilment of the requirements  
for the award of Doctor of Philosophy of the Loughborough University  
of Technology**

**January 1991**

**Supervisor : A.A. Clarke, Department of Computer Studies**

**© by Neil Stephen Watkinson 1991**

**PAGE  
NUMBERING  
AS ORIGINAL**

## DECLARATION

I declare that I am responsible for the work submitted in this thesis, and that the original work described is my own. I also declare that neither this thesis nor the work contained therein has been submitted to this, or any other institution, for the purposes of obtaining a higher degree.

Neil Stephen Watkinson

1991

## DEDICATION

I dedicate this work to my wife Susan, who has offered me unflagging support throughout and has provided constant encouragement and inspiration.

## **ACKNOWLEDGEMENTS**

Firstly, I wish to acknowledge my debt to my supervisor, Tony Clarke, for his constant guidance and inspiration. His good humour and assistance, both practical and theoretical, proved to be invaluable.

Secondly, I wish to acknowledge the role of my director of research, Professor Ernest Edmonds, for making available the appropriate facilities and for the inscrutable guidance he has provided.

Thirdly, I would like to acknowledge the Science and Engineering Research Council and express my gratitude for their provision of the funding which enabled me to carry out this research.

## ABSTRACT

This thesis describes the development and evaluation of a theoretical framework to account for the dynamic aspects of behaviour at the Human-Computer Interface (HCIF). The purpose behind this work is to allow for the consideration of dynamic Human-Computer Interaction (HCI) in the design of interactive computer systems, and to facilitate the generation of design tools for this purpose.

The work describes an example of a design tool which demonstrates how designers of interactive computer systems may account for some aspects of the dynamics of behaviour, involved with the use of computers, in the design of new interactive systems. The thesis offers empirical and literary evidence to support the validity of the dynamic factors governing the interaction of humans with computers.

The design tool described provides a method whereby changes in interaction behaviour, over time and between different user groups, can be modelled. This constitutes a dynamic model of interaction. In evaluating the tool statistically significant variations in behaviour between user groups during interaction were identified. This suggests that useful design tools can be generated from the framework which is described. Additional experimental work lends support to the notion of dynamic evaluation.

The authors intention is that this work shall influence the way designers think about interactive behaviour so that they can produce useful design tools for HCI. This will therefore enable them to adopt design practices which exploit the dynamics of HCI behaviour.

## **TABLE OF CONTENTS**

### **SECTION 1: HISTORY**

#### **1 INTRODUCTION**

**Page 23**

- 1.1 A history of interactive systems
  - 1.1.1 Associated devices
- 1.2 Usability measurement
  - 1.2.1 Notable theories in the field
    - 1.2.1.1 The metric of Card, Moran and Newell
    - 1.2.1.2 The metric of Phyllis Reisner
- 1.3 Designing for the user
  - 1.3.1 Why take the user into account?

#### **2. THE DESIGN TASK**

**Page 34**

- 2.1 Designing an interactive computer system
  - 2.1.1 The iterative design cycle
- 2.2 The origins of usability assessment
- 2.3 Assumptions of usability metrics
  - 2.3.1 Validity of usability metrics
  - 2.3.2 Predictive evaluations and assumptions



- 2.4 Deficiencies of current metrics
- 2.4.1 Techniques for overcoming these deficiencies

### **3. MODELS OF USER, COMPUTER AND INTERACTION**

**Page 47**

- 3.1 A definition of a model
  - 3.1.1 Models and specification techniques
- 3.2 History of the user and user's model
  - 3.2.1 Keystroke-level model
  - 3.2.2 Moran and layer models
  - 3.2.3 Other types of model
- 3.3 What models represent
  - 3.3.1 How models help the designer
- 3.4 Deficiencies of the current models
  - 3.4.1 How models can be more powerful, accurate and realistic
  - 3.4.2 Dynamic user models

### **4. REPRESENTATIONAL TECHNIQUES**

**Page 60**

- 4.1 The role of representational techniques
- 4.2 Media for recording user or interaction state
  - 4.2.1 Knowledge requirements specification
  - 4.2.2 Functional requirements specification
  - 4.2.3 Data requirements specification
  - 4.2.4 Dialogue requirements specification

- 4.2.5 Task-based specification
- 4.3 Scope, power and accuracy of various techniques
  - 4.3.1 How the techniques constrain the design
  - 4.3.2 Validity and reliability of techniques
- 4.4 Deficiencies of current techniques
  - 4.4.1 Taxonomic description structures
  - 4.4.2 How representation techniques can be made more flexible, powerful and realistic
- 4.5 Rationale behind the choice of interaction device knowledge for the representation of users

## **5. MULTI-LEVEL ANALYSIS**

**Page 73**

- 5.1 Multi-level analysis
  - 5.1.1 Layer models and their relevance
  - 5.1.2 Types of layer model
- 5.2 High and low level analysis
- 5.3 Multi-level support
  - 5.3.1 The distinction between naive and professional systems and users
  - 5.3.2 Mapping of knowledge structures through the interface
    - 5.3.2.1 Usability as a facility for transferring high level knowledge structures
- 5.4 Task specific interfaces
- 5.5 Windows on the world

## **6. FUNCTIONAL AND COGNITIVE ERGONOMICS**

**Page 87**

- 6.1 Functional and Cognitive Ergonomics as design desirables

## **8. TRANSPARENCY AND VISIBILITY**

**Page 113**

- 8.1 Transparency
  - 8.1.1 Transparency as a design desirable for computers
  - 8.1.2 Accounting for Transparency in the design cycle
- 8.2 Visibility as a determinant of Transparency
  - 8.2.1 The distinction between Visibility and Seeability
  - 8.2.2 Active perception of system objects
  - 8.2.3 Transparency as an essential precursor to accessibility
- 8.3 Techniques for enhancing Transparency

## **SECTION 2: ORIGINAL THEORETICAL WORK**

## **9. CALIBRATION OF HUMAN-COMPUTER INTERACTION**

**Page 122**

- 9.1 Temporal calibration of Human-Computer interaction
  - 9.1.1 Types of scale appropriate for assessment
    - 9.1.1.1 Clock-based scales
    - 9.1.1.2 Event-based scales
    - 9.1.1.3 User- based scales
  - 9.1.2 Simultaneous modelling on different scales
    - 9.1.2.1 Simultaneous modelling as a basis for describing priorities and scheduling
    - 9.1.2.2 Some example scheduling strategies displayed on a scale
- 9.2 A temporal description for assistance in HCIF design

- 9.2.1 Machine-pacing
- 9.2.2 Location of events
- 9.3 Group representation theory
  - 9.3.1 Range of conditions for occurrence of evolution
  - 9.3.2 Evolution range as a basis for self-adapting systems
  - 9.3.3 Customised scales for each type of user
  - 9.3.4 Conceptual maps
  - 9.3.5 Group knowledge overlap tool
- 9.4 How output of the technique can be varied
  - 9.4.1 The utility to the designer of an 'a priori' knowledge of temporal structure
  - 9.4.2 Externalisation of temporal task descriptions
  - 9.4.3 Evaluation of temporal values and position
  - 9.4.4 Identification of simultaneous activity
  - 9.4.5 Location of decision and evolution points
  - 9.4.6 Location of user requirements
- 9.5 The role of calibration in the system development cycle
  - 9.5.1 Abstract models
  - 9.5.2 Simultaneous generation along with a prototype
  - 9.5.3 Retrospective application of a scale to a working system
- 9.6 Meaningful calibration
- 9.7 A standard task-path description technique
- 9.8 Behavioural data extraction techniques

- 10.1 Identifiable discrete interaction states
  - 10.1.1 Task sub-goals or states
  - 10.1.2 Finite states
  - 10.1.3 Dialogue states
  - 10.1.4 Knowledge states
- 10.2 Evolution of the user
  - 10.2.1 Knowledge acquisition and learning
  - 10.2.2 Regression of the user
- 10.3 Modelling knowledge states
  - 10.3.1 Evolution of the model via salient changes
- 10.4 Users information and knowledge requirements
- 10.5 Types of evaluation
- 10.6 Decision support
- 10.7 Physical requirements
- 10.8 A formal definition of usability in terms of user requirements
- 10.9 Strategies for increasing user processing capabilities

- 11.1 Dynamic Human-Computer Interaction
  - 11.1.1 Describing dynamic situations
  - 11.1.2 Dynamic information models
  - 11.1.3 The action models

- 11.1.4 The attribute models
- 11.1.5 The holistic models
- 11.1.6 The pragmatic models
- 11.1.7 Application of the theoretical framework
- 11.2 Current HCI design methods
  - 11.2.1 Frameworks and methods
  - 11.2.2 Deficiencies of static evaluation techniques
- 11.3 The contribution of this work to HCI
- 11.4 The contribution of dynamic HCI to design
- 11.5 Thesis statement
- 11.6 Requirements for a study for the evaluation of dynamic Human-Computer Interaction
  - 11.6.1 Requirements for the design tool
  - 11.6.2 Requirements for the theoretical framework
- 11.7 Selected evaluation methods
  - 11.7.1 Rationale behind the development of the selected evaluation methods

## **12. THE EVALUATION OF DYNAMIC HUMAN-COMPUTER INTERACTION Page 207**

- 12.1 A definition of evaluation
- 12.2 The design tool
- 12.3 The dynamic model in the evaluation
  - 12.3.1 Representation of the model in the exemplar
  - 12.3.2 The group model
  - 12.3.3 The task model

- 12.3.4 Robustness of the model
- 12.4 Modes of evaluation
  - 12.4.1 Abstract models
  - 12.4.2 Working models
- 12.5 Focus of evaluation
  - 12.5.1 The introduction of a temporal element
  - 12.5.2 The introduction of a calibration scale
  - 12.5.3 The introduction of an interaction path
  - 12.5.4 The introduction of decision points and knowledge states
  - 12.5.5 The introduction of user experience or context
- 12.6 The grain-of-analysis of the method
- 12.7 The validity of dynamic evaluation

### **13. THE APPLICATION OF THE THEORETICAL FRAMEWORK Page 222 FOR DYNAMIC HCI**

- 13.1 A definition of dynamic human-computer interaction
- 13.2 Principles of dynamic human-computer interaction
- 13.3 Attributes of dynamic human-computer interaction
  - 13.3.1 The generation of novel design tools
  - 13.3.2 Rationale behind the development of the selected design tool
- 13.4 Appropriate areas for dynamic evaluation
  - 13.4.1 Potential benefits of the evaluation of dynamic HCI
- 13.5 Assumptions necessary for the evaluation of dynamic HCI
  - 13.5.1 The viability of dynamic evaluation

13.6 The third dimension

13.7 The dynamic interactive experience

### **SECTION 3: ORIGINAL EMPIRICAL WORK**

#### **14. AN EXEMPLAR EVALUATION**

**Page 249**

14.1 The scope and relevance of the evaluation technique

14.1.1 The system specification cycle

14.1.2 Some assumptions made

14.2 Rationale behind the selection of the exemplar interfaces

14.3 The data extraction phase

14.4 The modelling phase

14.5 Induction of temporal information

14.6 Formation and evolution of the user models

14.7 Induction of regression data

14.8 Results of the evaluation

14.8.1 Feedback

14.8.2 Statistical tests

14.8.3 Evaluation of systems

14.9 Discussion and conclusion of exemplar study

14.9.1 Feedback

14.9.2 Statistical tests

14.9.3 Informal evaluation



14.9.4 Grain-of-analysis

14.9.5 Analysis of the synthetic constructs inherent in the model

## **15. EMPIRICAL WORK**

**Page 295**

15.1 Introductory statement

15.2 Study 1

15.2.1 Introduction

15.2.2 Method

15.2.3 Results

15.2.4 Key points from study 1

15.3 Study 2

15.3.1 Introduction

15.3.2 Method

15.3.3 Results

15.3.4 Key points from study 2

15.4 Study 3

15.4.1 Introduction

15.4.2 Method

15.4.3 Results

15.4.4 Key points from study 3

15.5 Study 4

15.5.1 Introduction

15.5.2 Method

15.5.3 Results

15.5.4 Key points from study 4

15.6 Discussion

15.7 Conclusions

## **SECTION 4: DISCUSSION and CONCLUSIONS**

### **16 DISCUSSION and CONCLUSIONS**

**Page 328**

16.1 Discussion of the empirical work

16.2 Limitations of the work

16.2.1 External Validity

16.2.2 Methodology Problems

16.2.3 Objective evaluation scales

16.3 The next step

16.4 The practical application of this work

16.5 Additional features offered by this thesis

16.6 Problems with timescales

16.6.1 Problems with the timescale of this work

16.7 The practical use of this technique

15.5.4 Key points from study 4

15.6 Discussion

15.7 Conclusions

## **SECTION 4: DISCUSSION and CONCLUSIONS**

### **16 DISCUSSION and CONCLUSIONS**

**Page 327**

16.1 A discussion of the empirical work

16.2 Limitations of the work

16.2.1 External Validity

16.2.2 Methodology Problems

16.2.3 Objective evaluation scales

16.3 The next step

16.4 The practical applications of this work

16.5 Additional features offered by this thesis

16.6 Problems with timescales

16.6.1 Problems with the timescale of this work

16.7 The practical use of this technique

## SECTION 5: APPENDICES

### APPENDICES

Page 348

- i. References
- ii. Bibliography
- iii. Knowledge Kernels from exemplar evaluation
  - a. Total Kernel
    - 1. Overall
    - 2. Macintosh
    - 3. UNIX
  - d. A list of generic actions
  - e. A list of generic objects
- iv. Proposed system description
- v. Structured Interviews
- vi. Glossary of abbreviations
- vii. Thesis Map

## **FIGURES**

- Figure 1 A five-layer model of Human-Computer Interaction
- Figure 2 Some temporal calibration scales for Human-Computer Interaction
- Figure 3 Three dimensional modelling of Human-Computer Interaction events
- Figure 4 A generic activity model for shipboard threat management
- Figure 5 A general knowledge model for the shipboard exemplar
- Figure 6 The overlap of similar generic activity in different groups
- Figure 7 The standard task-path description technique
- Figure 8 Categories of dynamic behaviour
- Figure 9 A 3-Dimensional model of HCI
- Figure 10 The interface as the product of dynamic HCI
- Figure 11 The system specification and evaluation cycle
- Figure 12 Amendments to the behavioural kernel due to user group
- Figure 13 Amendments to the behavioural kernel due to user level
- Figure 14 Scoring of data for correlation tests
- Figure 15 Correlations between and within groups (t) using single commands
- Figure 16 Correlation values for user groups using strings of commands
- Figure 17 Correlation (t) test results for user levels for single and strings of commands between and within groups
- Figure 18 Sign and variance test results for single commands and strings at various levels
- Figure 19 Sign test and variance values for user groups using simple strings of command
- Figure 20 Sign test and variance values for user groups using single commands
- Figure 21 Task times generated from behavioural kernels
- Figure 22 Error distribution in experimental trials

Figure 23 Total commands produced for 2 trials

Figure 24 Access points and times in experiment 2

Figure 25 Total task times (seconds) in experiment 4

Figure 26 Mean subtask time (seconds) in experiment 4

**SECTION ONE: HISTORY**

## 1. INTRODUCTION

### 1.1 A HISTORY OF INTERACTIVE SYSTEMS

The last two decades have seen remarkable progress in the advancement of technology for the implementation of computer systems and nowhere has the progress been more rapid than the field of Human-Computer Interaction (HCI) (Shneiderman 1986). In early computer systems processing power was located in one place and access to that processing area was allocated using time-sharing and batch processing techniques. Instructions to the computer were invariably coded onto punch-cards and input via a card-reader which converted the instructions into machine-code to enable the computer to execute them. System response times were in the order of hours or days depending upon the workload at the time. Terminals which users worked at were 'dumb' and were not 'on-line'(Gaines and Shaw 1984).

Advances in communications technology and microprocessor control combined with increases in local processing power in the last twenty years have meant that most user tasks can now be executed on local machines - microcomputers and minicomputers - or communicated down very fast lines to multi-tasking mainframes which can handle hundreds of users simultaneously. System response times have been reduced to subliminal levels (shorter than can be perceived by humans) for most fundamental instructions and reduced to seconds or minutes for more complex or larger typical tasks (Shneiderman 1981). This reduction in processing time was one of the major precursors of the 'interactive' systems which now exist, displaying interactive dialogue between a human and a computer on a similar timescale and level to that of inter-human communication (Gaines and Shaw 1984).

As a result 'ease of-use' issues have come to the forefront and, with the advent of interactive systems, response time is now so rapid as to facilitate true interactive communication. The



imposition of a large computer interface on a computer system was also previously impractical due to limitations in processing power, however the increase in processing power experienced over the last few years now makes some quite sophisticated interfaces possible (Shneiderman 1986).

Unfortunately for designers, computers do not automatically possess the sophisticated, inherent communication facilities which enable human beings to interpret and understand each other. This means that computers must be explicitly equipped with the facilities necessary to allow them to communicate both efficiently and easily with users (Chapanis 1974). Current operating systems incorporating a set of generic commands designed to support the user in the execution of a given set of tasks do enable the categorisation and description of many problems. However, a well-designed command set may not necessarily be sufficient in itself to allow a user to display interactive behaviour, or to make the system 'easy-to-use' for human beings (Carroll 1982).

A Human-Computer Interface (HCIF) only arises when a computer and a user come together; here, a number of complex and variable factors interact to create the interface (Green et al 1983). Hence, if the full interactive potential of the interface is to be exploited, then attention must be paid to all aspects of the HCIF during the design process so as to support all the possible channels of Human-Computer Communication (HCC).

Human behaviour is immensely variable, robust and adaptable and so, until recently, users were expected to exercise these faculties during interaction with computers to adapt to the machine's in-built formalism. Although, it is essential that such a formalism is included in interactive systems, as communication cannot proceed without it (Reisner 1982), the types of formalism originally used were derived solely from the structure of the computer or based on an expert's model of computers. This meant that users learning or interacting with the system were often forced to spend a large amount of time learning the system, and were required to adopt methods which were 'unnatural' to them and which required great effort to implement. Recent increases in processing power have meant that sophisticated HCIFs could be built which handle some of the communication component

of the interaction and which allow users to utilise methods which are more familiar, more natural and based on attributes of typical human beings (Moran 1984): these are known as 'ergonomically designed', 'easy-to-use' or 'usable systems'.

It is now widely accepted that the goal of producing easy-to-use systems is a desirable objective because it provides users of varying experience with the most efficient and palatable interface possible (Shackel 1982, Edmonds 1986). It is also reasonable to accept that many casual users are not willing, and therefore should not be forced, to spend a large amount of time in learning the operating system of a computer, as computer based activity may not be a principal skill within their job-role and may not play an integral part in them achieving their goal (Eason 1977).

Sophisticated computer interfaces give naive and novice users access to virtually all the powerful functions which were previously only available to the most experienced of professionals. Even where users are prepared to spend some considerable time in learning how to interact with a computer this does not justify the existence of unwieldy human-computer interfaces which impose unreasonable learning, operating and understanding overheads on the user where these are not necessary (Young 1985). It is true that in some systems the nature of the task requires a complex interface and hence there is little opportunity to design usable interfaces; however, these are rare.

The power of interactive systems lies in allowing the user access to the tools needed to perform a task through a transparent interface, and this can only be expedited through the design of the HCIF such that it reflects existing human behaviour, strategies and cognitive attributes (Shneiderman 1986).

### 1.1.1 Associated devices

A number of advances in interactive technology made in the last decade have facilitated unprecedented increases in the usability of computers (Smith et al 1982, Perry and Voelcker 1989):

The advent of high-resolution monitors and 'bit-mapped' screens have meant that users can communicate with machines through graphical interfaces, thus exploiting the potential of the human bias towards visual representations, and enabling the design of flexible interactive systems. This flexibility obviates the necessity for the user to have to initiate or structure interaction, or memorise a large number of commands or assumptions.

The introduction of such software features as icons, menus and windows plus the development of efficient pointing devices (notably the 'mouse') mean that users can now execute many of the more fundamental tasks involved in HCC using simple, physical actions rather than being required to generate complex and esoteric command sequences, which can overload human memory (Reisner 1984). Hence, the user's cognitive facilities can be utilised in performing the task rather than negotiating a complex interface (Rutkowski 1987).

Work has also been done in implementing knowledge-based and adaptive systems (Greenberg and Witten 1982) which have the capability to support natural human input such as speech, handle poorly formed input robustly, and solve problems using knowledge-based methods. These systems offer supporting interfaces which adapt to specific user needs during interaction and hence offer constantly optimal support to the user. These are believed to be some of the most promising areas of future advance in the field of usable systems as they exploit intelligence and user specialisation as well as good ergonomic design practice (Greenberg and Witten 1982).

## 1.2 USABILITY MEASUREMENT

Usability can be enhanced in the design of interactive computer systems using techniques which provide feedback for the evaluation of the usability of proposed or existing systems. This process enables the assessment of interfaces in terms of their abilities to meet users needs and provides the means for the assessment of specific designs (Reisner 1984). The efficient management of the measurement of usability reduces the overall effort incurred in producing systems by ensuring that

designs are usable within a few iterations of the design cycle (Bennett 1984).

Usability measurement is not restricted to the application of formal tools but embraces a number of fields of study, all of which supply useful information on the criteria which must be met in the design of usable systems (Monk 1985). The fields include: formal tools for the evaluation of proposed and existing systems; models and representation techniques which provide explicit and convenient representations and descriptions of interaction; psychological experimental methods for the examination of HCI; existing human science data and principles also form a body of knowledge to be exploited. Whether the above are sufficient sources of information for the comprehensive evaluation of usability will be addressed later.

Usability may, however, be so context or experience related that it is difficult to approach using the current range of methods (Barnard 1988). Some error behaviours may arise, for example, as a result of a mis-perception of the interface due to experience of a similar situation and inappropriate behaviour due to misplaced assumptions (Green 1984). These errors can only be analysed by using a 'deep' (implicit) interpretation of usability (Broadbent and Berry 1987). This thesis contends that many current models are insufficient to account for this type of usability problem and suggests that this category of problem is a key area which must be addressed in the design of many interfaces.

One example of a complex usability issue arises when the interface is viewed as an impenetrable and unspecifiable flow of multi-layered and constantly fluctuating information. This can sometimes be a fruitful approach to the analysis of usability, however a holistic approach to evaluation would contend that when the interface is rationalised it loses any concrete or meaningful interpretation which may have been possible (Licklider 1982). Clearly, assumptions must be made during the specification of interactive behaviour to make such evaluations viable (Young 1984). These assumptions can be valid due to the fact that many of the supposedly intangible aspects of interaction can be formally controlled by managing aspects of the target user and task and formal

target user and task and formal experimental controls of user attributes (Card, Moran and Newell 1985).

HCI appears to be 'a dynamic process involving a constant fluctuation of the state of the user and the interface throughout interaction' (Green and Van der Veer 1984), and this thesis contends that it would be advantageous if some account could be made of these 'dynamic' aspects of HCI during the measurement of usability. This approach retains some of the holistic integrity of the interface but would also make it amenable to analysis. Some current models of interaction are static in nature and do not reflect the true state of the user and interaction (Young 1984). This thesis asserts that current usability measurement techniques often fail to detect or evaluate certain dynamic precursors of usability and that tests which focus on some of the dynamic aspects of usability would remedy these faults. To this end 'dynamic' HCI is seen as an appropriate framework for the study of usability, and a number of tools are offered to implement this type of analysis.

### **1.2.1 Notable theories in the field**

A number of well-formed and empirically successful metrics (where 'metric' is taken to mean 'a measuring tool') currently exist and a number are now contending for that status: some examples of the most prominent are described below.

#### **1.2.1.1 The metric of Card, Moran and Newell**

Card, Moran and Newell In their book 'The psychology of Human-Computer Interaction' (Card et al.1982) chose to examine the user as a 'model' human processor in which the actions of the user are broken down into a series of largely physical primitives and allocated empirically assessed temporal values to give an overall task time for the performance of the human with a particular interface.

performance of a given task to be established from established empirical norms. This facilitates a primarily comparative evaluation between two competing designs in terms of the number of keystrokes (and hence the time-taken) required to perform the task. The large body of empirical work supporting Card et al is based on text-editing tasks and this body of work makes the normative values they adopt fairly reliable.

The metric shows a ninety percent success rate in predicting the time-taken to perform tasks on two competing interfaces when all the assumptions are met. The authors aim to predict task performance times for expert users performing 'routine' tasks assuming that they exhibit optimal methods and show no errors. The assumptions severely restrict the scope of the tool. However, within these boundaries the model is a strong one and is accurate to within twenty percent in its predictions involving short bursts of activity (Young 1985). Card, Moran and Newell's metric is, however, a limited instrument for the evaluation of all but the most well controlled situations.

#### 1.2.1.2 The metric of Phyllis Reisner

This approach to usability assessment involves the specification of interfaces in a formal grammar. The interface is described in terms of its complexity rather than via a solely temporal value. Reisner (1974, 1977) noted the use of formal grammars in linguistics and computer science to describe languages precisely and so proposed its use in the Human Factors (HF) field of computer systems. Reisner proposed this system as a design tool to map out the design process, whereby formal grammar may be used to describe an action language. Stress is placed on the users cognitive attributes rather than on a simple description of physical actions. Reisner tested and refined her system using formal grammar to suggest design changes in an interactive graphics system (ROBART 1). This model enables predictions to be made about human performance.

Reisner makes predictions about the performance of systems according to various criteria, and tests each of them empirically, allowing a comparison to be made. She does make some quite general

assumptions about the salient characteristics of language, user behaviour patterns and memory limitations (amongst others) but the model is still valid and powerful, not least because of empirical testing during the development of the tool. The model is based on a number of static attributes of the human cognitive system rather than a real-world representation of the user's behaviour.

### 1.3 DESIGNING FOR THE USER

It has already been suggested that consideration of user attributes in the design of interactive computer systems would facilitate the design of more usable computer interfaces. However, designing for the user also involves the creation of an interface which exploits capabilities and limitations of human cognitive, perceptual and motor faculties in producing a tool which is appropriate for use by human beings and which enables the user to perform his tasks with the minimum interference from the interface (Long and Whitefield 1989).

An alternative approach can be the creation of an interface which positively supports the user. The notion of support is a key one here: a machine can be said to be usable if it supports the user in the performance of a task, that is, it enables the user to perform the task readily without excessive application of mental facilities in the negotiation of the interface (Shackel 1986). A usable interface is therefore one which provides adequate tools and features, or alternatively documented assistance, to aid in the performance of a specific task. This requires the designer to possess a complete knowledge of the implicit and explicit requirements for a computer system which assists users in performing a task: often an even greater understanding than the users themselves. This implies the need for a set of sophisticated tools to assist in the identification and modelling of both explicit and implicit aspects of application environments.

Consideration of the users can also occur on two levels. This can be on a general level, in which universal human attributes are modelled and incorporated into a design in the knowledge that all users will perform in a particular way. Alternatively the focus can be on a specific level, in which

attributes of a particular user or user group are established and reflected in the design of an interface with the assumption that the system will only be operated by that type of user or in that type of situation. The first type of study involves psychological research to determine 'universal human' attributes which can apply to all systems, and to other fields of study. The latter type of study involves analysis of a particular task domain using techniques such as task analysis and the design of a system which reflects assumptions and structures found only in that domain. The second type of study provides material which could lead to the construction of more 'powerful' interfaces but which would not have the universal applicability of an interface based on general human attributes. The number of general human attributes which are applicable and relevant for the design of usable systems is limited, so major advances in the future may be in terms of domain specific criteria (Norman and Draper 1986).

Although, implementation of user designed systems must be done with reference to the users attributes and opinions the experience of the HCIF designer is usually required to configure a realistic system based on 'implicit' attributes of the user. Here, the user does not need to know how or why the machine actually works but merely needs to possess a working representation of the machine which will suffice to enable him to perform his task (DiSessa 1986). In fact, it is often prudent to to hide information about the 'workings' of the system from the user to prevent the formation of too complex a model by the user or to prevent the user from imbuing the interface with functional capabilities it does not possess (Weisenbaum 1977).

So, designing computer systems to give users what they really want is a very difficult and sophisticated process, due not least to the complexity of the behaviour displayed at the computer interface. This again strongly suggests the need for tools to assist the designer in managing that complexity and this thesis aims to provide sophisticated tools to this end.



### 1.3.1 Why take the user into account?

In summary, attention to both specific and general attributes of users allows for systems which are more:-

**USABLE** - They reflect user attributes and needs

**POWERFUL** - They incorporate specific functions which are useful in the domain

**FLEXIBLE** - They can cope robustly with differing input from a range of users

## **2. THE DESIGN TASK**

This chapter to discusses the role and contribution of usability assessment in design and the criteria upon which evaluation techniques can be built and assessed. The chapter specifies the context of design, within which the design tool described within this thesis must operate, and suggests some limitations that the design tool might overcome. Firstly, however, there is a discussion of the design of interactive computer systems.

### **2.1 DESIGNING AN INTERACTIVE COMPUTER SYSTEM**

The process of designing an interactive computer system is an extremely complex one involving the evaluation of a large number of parameters concerning performance, function and usability. The formation of interim formal models of the system helps in this evaluation by using paper-based tools or prototypes to assess the suitability of the software and hardware before and after it is produced (Shneiderman 1981). This ensures that performance goals for the system (including usability) can be assured early in the design cycle and that the costs of development are minimised.

Design can proceed in a number of ways. A typical method of system development is where a designer produces a specification of the target system and the software is generated to meet this specification. The software is then tested to assess its conformance to the specification and if there are any discrepancies the software is amended to remedy this. This process can often be very time-consuming and costly especially where the user interface is concerned. If the user interface does not meet the users requirements then many iterations can be required to correct the software. Usability evaluation can be employed at this stage to minimise the need for excessive development work by contributing to design in two ways:

- Providing an evaluation of the usability of a specification before the software is written
- Providing diagnostic advice about problems with the software after it has been written

Comparative evaluation of designs at the paper stage are often sufficient to decide between two designs at an early stage (Young 1984).

An alternative or complementary method typically involves the immediate construction of prototype software, which can be evaluated with the user, without a formal specification: this is known as an 'evolutionary' or 'rapid prototyping' approach. Using this method the interface can be assessed immediately but due to the ad hoc. nature of the approach there is no guarantee that it will meet the users requirements and the process can be as costly as the previous method. In practice, a mixture of methods is typically used.

A clear method for the capture and representation of user interface descriptions is required so that the possible designs can be evaluated and so that a consistent plan can be maintained during design (Moran 1984). This method must be sensitive to those features of the user and his environment in which the designer is interested and should express those features in an accurate, reliable and comprehensible way. These techniques are discussed in chapter 4.

The interface designer typically requires a sophisticated model of the user so that their behaviour and preferences can be accounted for in design. This 'user model' can be employed as the basis for usability evaluation: this is effected by assessing whether the system provides support for the users requirements as expressed within the model. It is not always possible, however, to create a realistic *formal* model of the user (Reisner 1982). The appropriate evaluation criteria will typically be established by the designer according to the demands of the development process and the application area of the HCIF. These models are discussed in the following chapter.

### 2.1.1 The iterative design cycle

The iterative nature of design can prove to be very beneficial if it is managed properly. Human Factors considerations, however, need to be established at the start of the development process as once the system has been realised in any physical form they will be difficult and costly to alter. Hence, it is extremely advantageous if a number of options can be explored and evaluated on paper before the construction process commences (Card, Moran and Newell 1983). The creation of a prototype is a good way of testing the dynamic components of an interface (Kidd 1988), however there are currently few guidelines as to how to move from the specification to an initial prototype. The work described here aims to assist in this respect.

## 2.2 THE ORIGINS OF USABILITY ASSESSMENT

Traditionally, the assessment of how well an interface was designed could only be done by evaluating a working machine using controlled psychological experiments or by obtaining subjective reports from the user on perceived usability (Chapanis 1984). This entailed producing a working prototype of the machine and was a costly process. The less effort spent producing the early prototype, the less it resembled the final machine and hence, the less effective the tests. Additionally, spending a long time in producing full working prototypes meant that a machine could be out of date before the final version was produced. Techniques were required for rapid but realistic prototyping that would obviate the need to produce a costly working prototype.

Mathematical models could be made of the design of the hardware of the computer and these models enabled the designer to validate the design, to some extent, using mathematical and statistical techniques, however it proved difficult to form reliable, rigorous, sound and testable mathematical models of human behaviour due to the variability of human behaviour in the vast number of contexts which occurred in interaction with computers (Kieras and Polson 1984). A genuine requirement existed for techniques which allowed the designer to perform useful and realistic

evaluations of interactive behaviour early in the design cycle. The requirements for such an evaluation technique are outlined below:

- it is applicable to a wide-range of interfaces
- it helps distinguish between similar designs but can ideally provide some sort of absolute value of usability
- it helps designers conceptualise or externalise descriptions of potential interface configurations
- it gives a meaningful and useful value for usability couched in terms which are useful to the designer (Reisner 1984).
- It also ideally assists the designer in making design decisions.

### 2.3 ASSUMPTIONS OF USABILITY METRICS

All usability metrics must, by their nature, be founded upon a set of often quite major assumptions referring to the situation in which ease-of-use is being assessed. We must assume, firstly that human beings display rational, ordered, consistent and structured behaviour when interacting with computers: if this cannot be assumed then any study or model of HCI behaviour would not possess sufficient scope and could not be said to be applicable to HCI as a whole. The identification of the above consistencies provides, in itself, a difficult task for the experimenter, and this is one of the fundamental activities involved in ease-of-use assessment.

Experiments also assume that the explicit or displayed behaviour recorded actually reflects the user's underlying intentions. Information derived from self-reports, for example, has, to some

extent, been discredited by Rich (1983) as users do show extreme difficulty in reporting accurately the motives behind their own behaviour. This type of method should not be abandoned, however, because where the sample is large enough commonalities can be derived which are realistic. Current techniques for gathering behavioural data such as Protocol Analysis (Andersson and Simon 1985) and Constructive Interaction (Miyake 1986) have limitations. New techniques are required for the analysis of HCI. This problem is addressed later in this thesis.

Usability measurement techniques also assume that a general metric (or set of metrics) or 'measurables' exists which can be identified and quantified which correlate well with perceived usability as reported by users operating a working system (Shneiderman 1981): these measurables will be referred to in this thesis as 'Criteria'. Essentially, the goal of the designer is to identify a 'measurable' which bears sufficient correlation to the user's perception of their chosen aspect of usability to produce useful results. Time-taken to perform a task is a common measure adopted (Card, Moran and Newell 1985). This criterion can provide a good, objective value for the designer in terms of performance and as long as the designer's goal is a rapid interface then the measurement is useful. However, a fast task execution time is often a symptom of usability rather than a cause and hence this type of measure rarely provides any qualitative or diagnostic information (Young 1984).

An alternative 'measurable' used in the field is 'complexity' (Reisner 1977). This is often expressed as the number of rules which the users are required to hold in working memory to be able to interact with the interface. It displays a closer correlation to true usability because whilst time is a truly objective performance measure (which may be useful) complexity refers to an interfaces match to existing human faculties and hence makes a comment on the ergonomic soundness of the design (Kieras and Polson 1984). It is clear that no one measurable can be correlated perfectly with usability as interaction is such a complex process, containing many factors which can influence usability. Analysis from a number of perspectives is required if a comprehensive evaluation is to be expedited (Shafer 1988). Testing inappropriate or insufficient metrics may lead to failure to

identify isolated but profound usability problems, especially if error-behaviour is ignored. This thesis espouses the virtues of multi-dimensional evaluation to obtain a comprehensive and complete evaluation.

Some further candidate criteria are outlined below.

**REDUNDANCY:** The potential and actual redundancy inherent in an interface and the utility of such redundancy as a ratio of its cost to the user in terms of processing space occupied.

**ACCESSIBILITY:** The transparency of the interface and the effort required to access function.

**VISIBILITY:** The ability of the user to discriminate, recognise, identify, perceive, comprehend and locate a given object, or set of system display objects.

**COMPATABILITY:** The appropriateness of an interface design to a particular user approach or strategy and the reflection of the 'cognitive style' of the user.

**FAMILIARITY:** The amount of prior knowledge required of computers and the esoteric nature of interface. The effect that existing experience has upon the design.

**CONSISTENCY:** The extent to which any symbols and functions mean and act the same in all contexts and the methods by which this is preserved.

**DETERMINISM:** The amount of user control over interaction including whether the interaction is machine or user driven. The actual function which the user can access.

**DEDICATION:** The amount of domain specific information necessary to operate interface and the extent to which the user indulges in specific domain related activity during interaction.

**MEMORABILITY:** The tendency of users to remember items in the interface as a result of the configuration of display objects and functions. The effect that interference from existing contents of memory has upon interaction.

**LEARNABILITY:** The tendency of users to assimilate, comprehend and store the operating paradigm and its features as a result of the design of the interface. The relation of any new material to the user existing conception of interfaces or similar situations.

**LOAD:** The parameters of the Interfaces demands on Short-Term Memory during interaction.

**REALISM:** The relation of simulated portions of the interface to their real-world counterparts and the realism of any simulations.

**SOCIAL:** The allowance for social facilitation during interaction by the interface and the amount of control over any higher level goals which would enable 'self-actualisation' (Newman 1984).

**NATURALITY:** The extent to which the machine follows the users preconceived notions about how a computer or simulation will behave.

**COMPLEXITY:** The amount of material (number of rules and facts) which a user is required to hold and manipulate to be able to interact with a machine.

**AESTHETIC:** The extent to which the *patterns and designs* of the computer (for example, the Display) appear natural to the user or resemble those familiar to the user.

**HOLISTIC:** How the user perceives the interface as a whole and the effect this has on Interaction behaviour.



**MOTIVATION:** The attractiveness of goals presented to the user and the users success in attaining those goals.

**FEEDBACK:** How concrete, directed and rapid is feedback especially in the handling of errors.

This list is not exhaustive but is distilled from the criteria most mentioned in the literature (Shneiderman 1986 ) along with some that the author has found to be relevant. There is clearly some considerable overlap among these dimensions, however it is believed that this redundancy is vital if all relevant aspects of usability are to be tested.

### **2.3.1 Validity of usability metrics**

The validity of a usability metric is taken to be the extent to which it measures what it purports to measure, under the circumstances stated, and the utility of the results in the design of a system: this is essentially a very powerful measure for the integrity of usability metric. The purported measurement is not necessarily just the claims of the metrics originators and its users but also includes the claims implicit in the nature of the metric and the mode of use of the output of the metric. Psychological testing has identified a number of types of validity which define the power and scope of a psychological test (Anastasi 1979): they include Face-validity, Content-validity, Construct-validity and Criterion-related validity. These types of test have been used, to some extent, in the assessment of usability metrics, and they share some of the assumptions common to tests in the Human Sciences.

Choice of criterion is perhaps the most vital factor in determining the validity of a test (as asserted in the previous section). For example, if the 'time-taken' element of Card, Moran and Newell's test is taken to be a valid criterion on which to assess usability, then the metric can be said to be valid, and the tool can be said to be sound in terms of criterion-related validity. Such an assumption may

prove to be unwise in many cases, however this criterion has been shown to be unreliable as a sole determinant of usability (Young 1985).

Another useful determinant of test quality can be found in reliability assessment (Gilb 1977). Reliability can be defined as the extent to which a test performs consistently across a range of subjects or the extent to which the test produces the same type of result from the same type of subject in the same circumstances (Anastasi 1979). Reliability can be maintained more readily with a thorough knowledge of the subject population as this enables identification and control of possible spurious factors. With close control over subject domain commonalities the actual effects of the metric can be isolated.

Production of valid and reliable metrics ensures that they are useful design tools which give meaningful, useful, consistent and reliable results which can be safely exploited by the system designer (Chapanis 1982). Whilst such test evaluation criteria may not apply as well to usability tests as they do to other Human Science tests they provide a meaningful method for the control and testing of the metrics themselves.

### 2.3.2 Predictive evaluations and assumptions

Reisner introduced the term 'Prediction Assumptions'(1981) to refer to the set of assumptions that could be made about the future performance of a user when faced with a particular design configuration. She exploited this notion to evaluate competing designs in terms of the future demands of the user due to complexity, consistency and compatability.

This is a valid approach when empirically proved principles of human behaviour are utilised, however it does have some inherent hazards. Reisner chose only well-documented examples of cognitive phenomena and so her model was very reliable. If this model could be extended to behavioural phenomena then it would be much more powerful. However, local contextual and

experiential effects on behaviour give rise to great variability.

The value of 'Prediction Assumptions' in Reisner's particular sense is in evaluating a particular specification against 'known' attributes to give a usability value before a system is prototyped. The accuracy of this method is based upon a number of features: The relevance of the model of the user (albeit a collection of hitherto unconnected cognitive facts), the completeness of the model of the user in terms of all the possible situations and users towards which the technique could be applied, and the accuracy of the system description in comparison with the level of description of the user.

The hypothesised description of the users behaviour should be compared to the actual strategy which would be employed by a user. This technique assumes adoption of optimal methods and whilst this technique offers an assessment of the potential performance of the system it does not give an accurate indication of the actual distribution of user behaviour. Hence, some questions can be raised about the models of the user, machine and interaction inherent within these models.

Kieras and Polson (1984), whose model is described later, overcome this problem by forming three independent models of the user, the system and the interaction. So, the role of this type of approach is to allow the formation of predictive models of the user for the purposes of the objective evaluation of a design configuration; this method is supported by Young's (1984) comments on the value of user models, which are:

- To provide a predictive evaluation of proposed (partial) designs, and so help to locate the design within the range of options
- To supply the designer with a story about the users behaviour enabling him to create a rational design. The form of the model specifies the design space and helps him to focus attention on the key variables

- To provide a framework for designing and interpreting experiments.

## 2.4 DEFICIENCIES OF CURRENT METRICS

The deficiencies of current metrics can be discussed in the context of the criteria mentioned in the last three sections. Many of the current techniques are reliant on the satisfaction of large number of gross assumptions about the conditions of evaluation, so the power of the metrics is vested in its limited applicability (Roberts and Moran 1984). In general, any metric will predict the performance of a particular group of users under certain conditions and consequently performance with one interface can then be compared against a competing design.

Deficiencies are also evident with reference to the reliability and validity of current metrics. Choice of appropriate or sufficient evaluation criteria has not been forthcoming and whilst Card et al (1985) choice of task-time is valid, again within its particular constraints, it does not correlate strictly with true perceived usability. Reisner's criteria refer to certain human faculties which correlate well with gross measures of usability, notably consistency and complexity (cited very frequently in the literature such as Shneiderman 1986) however Reisner's definition of these concepts within her metric is not valid in all circumstances. Essentially, most current evaluation metrics are based on criteria which have not been sufficiently researched as to their correlation with usability. These metrics do not apply the criteria very robustly when implemented in the metrics and do not use a sufficiently wide-range of metrics to achieve a comprehensive analysis of usability. There exist many specific domain-related criteria which could be identified as specific determinants of usability in specific circumstances if research were to be targeted to this area.

Again, the types of predictive evaluation and assumptions made by current metrics may be subject to substantial inaccuracies due to the combination of assumptions in modelling and simulating interaction. Hence, interaction is taken to be a homogeneous segment of activity in which assumptions or knowledge do not change (Johnson 1986).

Deficiencies remain in the type of measurement offered by current metrics. Measures are largely comparative, relative, ordinal and uncalibrated (non-linear). Chapanis (1984) states that all usability measurements will tend to be ordinal. However, if a set of valid usability criteria can be identified then absolute values can be given for performance on a number of scales and this will give clearer picture of usability as a whole. Experimenters are currently very reluctant to give true cardinal or ratio values partly due to the fact that trade-offs exist between certain usability criteria such that some are mutually exclusive (Miller 1972) and hence an absolute value of usability is difficult to identify. This situation can be remedied, to some extent, by stating usability requirements more explicitly and specifically and by elucidating a generally agreed set of criteria and related set of standard measures on which to base the assessment of specific and selectable aspects of usability. This would need to be a set which displays an exhaustive and comprehensive coverage of usability factors. Young (1984), however, points out that there are few complete and well formed usability assessment tools which account for both user behaviour and cognition and offer tools for its complete specification.

#### **2.4.1 Techniques for overcoming these deficiencies**

This section deals briefly with some possible techniques for the creation of more powerful usability metrics.

Most current models are essentially 'static' in nature, in that they produce one 'snapshot' type model of user attributes and their effect on interaction, which is deemed to be pertinent throughout interaction. In actuality, both the user and the interaction process may constantly be changing (Christie 1985) and the major assertion of this thesis is that interaction is dynamic. It is essential that usability metrics reflect this by offering tools for the collection and representation of dynamic behaviour and by offering the facility to form dynamic models (Christie 1985).

One of the ways in which this thesis approaches the evaluation of dynamic HCI is by adopting a more pragmatic approach to domain definition. Current metrics assume a general user and attempt to derive general user principles (Card, Moran and Newell 1985). This approach precludes errors which are an essential feature of user behaviour and can be related to specific user types (Gilb 1977). Closer modelling of domains and users could offer a more useful approach.

Various media have been used to form comprehensive and realistic metrics for assessment of the usability of systems. The evaluation technique proposed in this thesis adopts a tool which allows employment of a multiple media and hence does not assume an all-pervading general criterion.

It must be reiterated, however, that there is no one generally applicable evaluation tool available for the design of interactive systems. Different tools are appropriate for varying types of interface. Card, Moran and Newell's tool is powerful (Young 1984) when all the assumptions can be met and as a result there may be a demand for many more specific tools to examine specific situations.

This chapter provided the setting for the design tools to be offered later. This thesis adopts the stance that such tools are only useful if they can be actually employed in design and this ideal provides a significant goal for the empirical work.

### 3. MODELS OF USER, COMPUTER AND INTERACTION

This chapter aims to define the term 'user model' and provide some diverse examples of models to demonstrate their range of applicability. The chapter also presents a set of criteria for idealised user models and discusses the necessity to incorporate dynamic factors in any user model.

#### 3.1 A DEFINITION OF A MODEL

In the opinion of the author a model is an instance of a description of a situation, entity, event or object which facilitates storage and description of all of its salient aspects and supporting assumptions such that it can be easily manipulated, understood and accessed. It is, in some senses, a purposefully poor representation of a situation which obviates the necessity to represent or learn about a whole environment explicitly but is sufficiently information rich to convey all the material necessary for the modeller to attain a given goal (Rich 1982).

The notion of a model of some aspects of user, machine or interaction is a useful one for the purposes of providing the interface designer with a compact and concise representation of some of the issues involved in interaction (Young 1987). In HCI such models take many forms but they can be categorised into a limited number of types (Shneiderman 1982). Firstly, in terms of the description of the interaction there are three generally accepted types of model which can be found in the literature:

A USER'S MODEL represents a conception held by the user concerned with and describing the various salient aspects of the HCI environment which determine interactive behaviour; this is commonly referred to as a 'user's conceptual' or 'mental model'(Shneiderman 1986). The model may be constructed via a combination of formal instruction - personal or documented - individual experience - of interaction and perceptions of the interface - experience of other machines and the

domain, and from assumptions inferred from a combination of the above three.

Users models are, to some extent, subsets of general mental models, which human beings must possess to be able to interact with and survive in the world (Gentner and Stevens 1982). These models have been the subject of considerable work (Young 1981, Allen 1984) and much of the theory behind mental models supports the notion of user models.

Another type of model is the USER MODEL. There are two important forms of this evident in the current HCI field. Firstly, there is the 'designers model of the user' which represents the designers view of the user population. Secondly, there is the 'embedded user model' which represents the state and behaviour of the user stored within the machine, as a piece of software, which simulates the user and enables the machine to 'reason' about the users behaviour (Self 1984). User models can be represented in many diverse forms from primitive motor and behavioural descriptions (Rumelhart and Norman 1984) to social or goal-based abstract conceptual descriptions (Clarke 1986) and these would typically be realised in different media. However, functionally they are of one of the two types described above.

The third type of model of interest within HCI is the INTERACTION MODEL. This type of model aims to simulate interaction by either linking a user and a user's model (Kieras and Polson 1984) or by generating itself sufficient properties of an interactive system to sustain a meaningful description (Parnas 1981). Interaction models are displayed in a number of types. Models which aim to simulate the structure of the differing levels of, and the flow of information and knowledge through, the interface, as well as some of the mapping constraints available or the communication protocols used (Moran 1984 et al.). Such models are known as layer models. Other models simulate interaction as a set of dialogue states linked by transformational rules. One example of this is a 'state transition network' (Parnas 1981).



User or interaction models are necessarily generalisations of the real-world however they are useful for the designer in enabling him to identify some of the relevant components of interaction at each level and abstract some of the potential properties of a given HCIF.

Models can take many forms and can have many subjects, however a model can be seen, as a 'method for externalising a conception of a description or representation of a user in a formal manner to be used as the basis for the iterative evaluation of a proposed design' (Reisner 1977). Hence, models are extremely useful to an interface designer as an "enforced method of being precise" (Reisner 1982) or in other words a means of clarifying and rationalising a set of assumptions about interaction behaviour.

### 3.1.1 Models and specification techniques

A specification technique is a descriptive mechanism, technique or language which facilitates the description of a number of situations or states and is characterised by its flexibility of operation over a given domain (Jacob 1986). A specification technique can be used to construct more than one 'model' and should have sufficient flexibility, in terms of vocabulary and repertoire, to describe all situations of interest to the experimenter and allow distinction between particular models. A model, however, is one instance of a description at a particular grain-of-analysis and may include a description of the particular idiosyncrasies of a situation which are not typical of that class of description.

A metric is essentially a different class of tool to a model or specification technique. A metric is a quantifiable criterion which can be deduced from, and applied to, an entity or situation which enables an experimenter to quantify the performance of that entity or situation against some predefined scale (Young 1984). The identification of reliable and consistent measurables is a complex and sophisticated process which can sometimes be assisted by the creation of models to explore the salience of a number of potential performance parameters.

Each of these type of tools is valid in its own right and is appropriate for application at different levels and in differing situations.

## **3.2 HISTORY OF THE USER AND USER'S MODEL**

A number of examples of the above types of model have arisen over the past decade and the following sections outline some of the more prominent ones.

### **3.2.1 The Keystroke-level Model**

This user model (referred to earlier) represents the user as a series of behavioural-motor and cognitive primitives which combine to define some higher-level task which the user is assumed to be performing. This representation forms part of the GOMS model which outlines the structure of the task described by the Keystroke-level Model, and represents a behavioural user model. It also purports to model some of the higher-level aspects of behaviour via the 'M' operator. This is a primitive description in purely behavioural terms in reality and whilst it is constructed of empirically assessed behavioural units it is of limited value to the designer unless all the assumptions are met concerning its application.

Such low-level behavioural models may be of some use in the identification of valid action primitives, however they may not bear direct correlation to the conceptual and cognitive issues in interface construction (Whitefield and Long 1989). Unfortunately, these two aspects are irretrievably interrelated as lower-level behaviour only exists to support functions. There is a need to know how the higher functions are mapped onto the behavioural model.

### 3.2.2 Moran and the Layer Models

Layer models outline some of the communication protocols between man and machine and display the varying levels at which information is handled in the interface. Such models do represent users, to some extent, but they are essentially models of interaction. The pioneering work on this type of model was undertaken by Thomas Moran (1982) and led to the production of the 'Command Language Grammar' model.

He proposes a representational framework for describing the HCIF aspects of interactive computer systems and partitions the system into three types of component. The Conceptual components include tasks and abstract concepts, Communication components include command languages and Physical components which can be displays or keyboards. The components are further stratified in terms of four distinct levels: Task level, Semantic level, Syntactic level and Interaction level. Each level provides a complete description of the system at its level of interaction. Each level contains the required procedures for describing the tasks addressed by the system in terms of all of the actions available at that level. Moran assumes that it is possible to produce a consistent logical and natural description of any computer application using his grammar. Moran, himself (1982), points out some of the fallibilities with this type of model. The higher-levels are too informal and there has been no actual empirical test of the usefulness of the chosen concepts. Similarly a rationalisation of the high-level procedures has not been shown, and there is a lack of one standard set of generative mapping rules from Semantic to Syntactic levels. Tasks are not well-specified at the Syntactic-level and the level at which entity names are introduced is questionable. Despite these failings Moran's is a powerful model and design tool HCI.

Other more sophisticated layer models have been derived from this work, such as Clarke's (1986) 3-layer model which seeks to act as vehicle for the incorporation of psycho-social factors into HCI design at a number of levels. Faenrich and Ziegler's (1987) model is used for the assessment of

direct-manipulation systems and seeks to outline the potential allocation of tasks between human and computer. Nielsen's 'Virtual Protocol' (1986) model seeks to outline the communication protocols between man and machine and stresses virtual communication at higher-levels with a finer grain-of analysis than Command Language Grammar. Finally, the authors own model (Watkinson 1986) which provides an explicit operational model of machine communication layers, is an attempt to rationalise specific human and machine mapping rules and communication protocols.

This type of model is discussed more fully in chapter 5.

### 3.2.3 Other types of model

Self's (1984) 'student model' represents a machine-based description of a users knowledge of a particular topic which enables the machine to structure a tutoring programme - not a computer programme - to educate the user on particular areas in which the model is deficient. The users knowledge of a domain is elicited using a series of interactive question and answer sessions and the model is updated after every session.

This type of model differs from other types of model in that it is constructed from explicit knowledge and information about a given external domain and does not represent an inherent user state. It has seen less research than the other types; however, it shows promise, since the development of a powerful embedded user model is an important precursor to the creation of new intelligent and robust user interfaces.

Embedded user models can be used to equip the machine with information about the user to enable it to generate customised responses or to process poorly formed input.

### 3.3 WHAT MODELS REPRESENT

As is shown above, a model can be represented in many forms and can play a variety of roles, however a model is only viable if the assumptions made when constructing it, and the criteria on which it is based, are valid for all instances of the employment of that model. Hence much of the activity involved in modelling concerns gathering data to test the consistency of the model and establishing a set of controls which must be maintained when applying the model. As soon as the model no longer consistently accounts for new data it is defunct (Green 1981).

Unexplained or 'unexplainable' behaviour is often placed in a mental model and ascribed to user idiosyncrasy. This has to be accounted for in the design process. Such a model is essentially of limited value as it does not enable the designer to interpret target behaviour in the context of real-world situations. The model becomes a repository for all the unrationalised aspects of the user which must be designed for as exceptions. The representation of behaviour is not sufficient as the structure of the interaction must be elucidated by the model and some meaningful statements should be made about user behaviour (Norman 1988).

#### 3.3.1 How models help the designer

Models equip the designer with a stable, if temporary, impression of the user or interaction. The models tell a 'story' about the user which can be utilised in design and ideally possess pointers to the important aspects of the user or situation (Shneiderman 1986).

A model can provide a designer with a starting point from which to generate a design and whilst an original model may be quickly abandoned they provide a focus of attention for designers. This can be termed 'conceptual rapid prototyping'. Even poor models will provide the designers with the interim steps to a solution as the wrong models must be rejected before the correct one is identified. Models represent a medium for storage of what is important to the user in the real-world and what is

relevant at a particular time (Bennett 1984).

Models provide an accurate representation of the users behaviour which acts as a reference point when choosing between design options. This enables the designer to create a system which is compatible with the users existing behaviour and allows him to take into account users abilities and limitations when specifying interaction.

Models should also offer methods for generating instances of real behaviour for the evaluation of the type of behaviour that a user may typically adopt if faced with a given design. They can be viewed as a large amount of behaviour condensed as an economical description and represented in terms of a textual description of behavioural instances (Shackel 1986, Young 1987). Such rules can be represented as 'usage scenarios', for example.

In summary, models offer the designer economy of representation through reduced redundancy and hence, offer a convenient and accessible description of a user or interaction. The model also allows meaningful generalisation and a means for the rationalisation of behaviour whilst providing a concrete framework or structure within which to initiate and execute design work.

### **3.4 DEFICIENCIES OF THE CURRENT MODELS**

Models such as Card, Moran and Newell's 'Keystroke' model do not provide a concrete representation of higher-level aspects of interaction. Additionally, the inappropriateness of the temporal value produced in the comprehensive quantification usability seems to limit the scope of this tool. Reisner's 'Prediction Assumption' model focuses on general human attributes and suffers from poor resolution of the complexity metric selected to represent usability. Conversely, Moran's and similar layer models offer a concrete framework for the specification of multi-level aspects of interaction, however they do not facilitate the quantification of the descriptions produced by the

model, so in real-world terms they provide a purely static description of interaction which does not reflect the users dynamic behaviour and ignores key 'dynamic' temporal issues.

The approaches described above assume a number of important things about HCI behaviour:

- It remains constant during interaction (in the short and long-term)
- It remains constant between interaction
- There is a canonical or typical user approximation to which reference can be made, that will enable the design of an interface which is suitable for all users

Whilst such assumptions may be tenable in particular types of interaction, any machine which offers various degrees of flexibility, or allows any improvements in performance through learning or experience, would imply a number of potential user configurations each of which is equally valid and based on differing interaction styles (Green and Van der Veer 1984). Support of this type of variety desirable as humans do interact with computers in a number of different ways of equal validity, and interaction is typically more successful if users can employ their preferred interaction style. Hence, static descriptions and the techniques used to produce them would seem to be of limited use in many cases and to be defunct in others.

In summary, a number of general deficiencies of some user models can be identified and this enables the attributes of a user model which could help to address those deficiencies to be described:-

### AN IDEALISED USER MODEL.

- It is quantifiable and can produce a value which is meaningful in the real-world and in design
- It can represent both specific and general user attributes
- It accounts for dynamic behaviour and its associated factors
- It allows specification of temporal issues
- It considers all levels of interaction and resolves higher-level descriptions realistically in terms of lower-levels
- It is based on a set of valid and tenable assumptions

#### 3.4.1 How models can be made more powerful, accurate and realistic

The extension of user models to incorporate descriptions of the dynamics of behaviour at the user interface is one very promising approach. This method would increase the realism of user models by accounting for the temporal aspects of real-world behaviour and would extend the power of the predictions which models could make by basing such models on a broader range of assumptions about the users environment.

Dynamic HCI is offered as a vehicle for highlighting and expressing a range of issues which can be addressed to extend the range and power of current techniques. Metrics and models which are designed with some of the precursors of dynamic interaction as their basis will embrace the aspects of HCI behaviour which are necessary for the comprehensive evaluation and assessment of real-world interaction.

Users behaviour changes during, and as a result of, interaction, and hence the set of assumptions



which applied to the users behaviour at the initiation of interaction may not be the appropriate or applicable set at other points in interaction (Greenberg and Witten 1982). The support offered to the user at the start of interaction may not be appropriate or adequate for the behaviour displayed later on and hence a series of models may need to be formed to plot the change in user state throughout interaction due to dynamic changes at the interface. For instance, users learn, change and forget so the model formed of the interface, and hence specific ergonomic needs, are constantly fluctuating. A representational model which does not reflect these features cannot be said to describe all the important aspects of usability. Current models are not all static, however they do not incorporate all the assumptions which would be valid for a dynamic tool.

Some of the major design improvements in interactive technology have arisen as a result of the modelling, implicitly or explicitly, of dynamic HCI. For instance, the interactive 'desktop' metaphor used in advanced interactive systems allows the exploration of interactive scenarios (Smith et al 1982): in knowing how to interact with a real-world desktop the user has an implicit dynamic model of how to interact with the virtual desktop.

Adaptive systems and intelligent systems seek to interact dynamically with, respond to, and control for dynamic behaviour in their interaction with users: this demonstrates the benefit of the application of dynamic models (Cooper 1985).

Models created using the framework described above could be more complete due to analysis via a more comprehensive set of criteria. Additionally, models could also be made more realistic following a quantitative and qualitative analysis leading to the production of values which more closely resemble users' 'perceived' usability. Many of the relevant interactive components of HCI are embraced in a dynamic system, and this offers an overall framework for the explanation of behavioural phenomena. The explicit representation of temporal aspects, which appear to be an independent dimension in all human behaviour (Ornstein 1982), including interaction, would be a

useful property of some models.

It is not claimed that a dynamic approach is always relevant and in some cases it will lead to the modelling of redundant information which would not be necessary in the evaluation of a simple static system. If the full potential of interactive computing is to be exercised, however, then some degree of redundancy must be described in the majority of applications. Dynamic factors are simply properties of HCI, and whilst they permeate many vital areas of interaction behaviour they do not supersede other factors, but merely enhance the meaning of the more static aspects of human behaviour. Dynamic evaluation offers an alternative perspective from which to approach evaluation in general.

### 3.4.2 Dynamic user models

If systems are to be developed which build upon the flexibility of human behaviour by working co-operatively with the user, rather than merely exploit human adaptivity in order to make interfaces which work, then a dynamic representation of how users and machines work together is required. Dynamic models reflect the flexibility for adaptation to new experience and environments which is characteristic of human information models and hence offer enormous potential for the development of interactive systems.

The task of modelling the changing user is replete with difficulties, however it requires no larger assumptions and controls than were made in developing existing ease-of-use metrics, concerning the regularity of human behaviour. Dynamic models are, however only a part of a successful dynamic measuring tool, and although they are one of the more palpable aspects they require sound supporting behavioural representation methodologies, meaningful presentation and a sound underlying set of assumptions.

This thesis argues that HCI is a dynamic process and that design activity should be based on this assumption. This chapter has introduced the notion of dynamic HCI to establish a context for the interaction model constructed later. The chapter also establishes some attributes of dynamic models which that model might possess.

## 4. REPRESENTATIONAL TECHNIQUES

This chapter examines the features of representation techniques which make them useful in the specification of HCI. The chapter also highlights some limitations of the current methods and outlines the criteria which would need to be met to resolve these shortcomings. Finally, the chapter introduces a representation medium - Interaction Device Knowledge - and a taxonomy (Carter 1986) to describes users actions, which will be used in the empirical work.

### 4.1 THE ROLE OF REPRESENTATIONAL TECHNIQUES

Representational techniques usually consist of formal or semi-formal notations for the description of Human-Computer Interaction. They typically utilise a grammatical or graphical syntax to express the activity at the computer interface in a rigorous form. Such techniques can take many forms and these are discussed later, however it is first useful to discuss some of the features of notations which make them useful in the specification of HCI.

Representational techniques ideally possess the capability to represent descriptions of users and HCI in terms independent of a particular interface or task configuration (Jacob 1986). Hence, descriptions should not be constrained to implementation in only one fashion. This enables designers to delay full commitment to a design until the last possible moment in the design process and to utilise the greatest possible range of equipment, as well as modelling a large proportion of the available task configurations which the user may execute. Equally, such a technique should be able to model the essence of any situation in descriptions which are as unverbose as possible, show clarity and low density, and do not capture unnecessary detail about subjective issues or local considerations (Jacob 1986).

A representational technique should have the capability to reflect and highlight the structure inherent in the system to be modelled without applying any arbitrary structure itself. Whilst models provide structured descriptions of instances of HCI situations, the author believes that representational techniques (RT) provide the vocabulary for doing so, and should be flexible in their application. This requires the capacity to store some diverse, generic structures or concepts, and in this respect many techniques are based on, behave like or can be used in a similar fashion to computer languages (Johnson 1986).

The descriptions produced by a technique should reflect any biases inherent in the system to be modelled. Hence, systems which involve a great deal of complex information access and handling should be represented by a technique which stresses facilities based on the operations, structures, units and behavioural and cognitive phenomena displayed in these situations (Christie 1985).

Representations should also appear like the underlying phenomenon which is being described (Jacob 1986). This enables the user of the description to interpret the underlying situation being described. Often this requirement can necessitate the development of a new specialised technique, however extensions of existing techniques are often possible to encompass new HCI descriptions whilst the focus and the structure and content of existing tools is appropriate (Jacob 1986).

R.T.s will normally fit into a typical specification cycle between data gathering or knowledge elicitation and the formation of the user model. They offer designers a tool for transforming a mass of raw data or knowledge into a coherent model (Rich 1982). These tools also often provide the basis for constructing an initial user model although many possible models should be able to be generated from one specification. The R.T. should naturally map onto the data obtained by various means or act as a more sophisticated tool for structuring data gained through such techniques as task-analysis or protocol-analysis (Andersson and Simon 1985). Hence, raw data about application environments can be transformed into useful models by using appropriate R.T.s.

## 4.2 MEDIA FOR RECORDING USER OR INTERACTION STATE

User and interaction state can be realised in a variety of media dependent upon the required focus on the characteristics of the situation to be modelled. Applications which require integration of specific user expertise - for example, expert systems - would invariably use knowledge as the preferred medium for representation (Johnson 1986). Representation techniques could offer a hybrid approach which would facilitate description using a number of media dependent upon which is appropriate, and this would reflect the diversity of media inherent in many tasks (Shafer 1988).

### 4.2.1 Knowledge requirements specification

Applications can be described in terms of the knowledge the user must possess to be able to interact with them and so this can be seen as a highly flexible type of description (Johnson 1984). Such descriptions of user activity are particularly useful when designing interfaces to interactive systems as they allow the designer to formally specify the knowledge which the user possesses to execute the task. They also provide a set of requirements which can be addressed within a design whilst not constraining them to specific equipment or task configuration. A promising current technique is that offered by Johnson (1984). This technique is termed 'Task-Analysis for Knowledge Descriptions'(TAKD) and is particularly useful in HCIF design as it produces task-independent descriptions of users models.

Drawing distinctions between 'procedural' and 'declarative' knowledge (Anderson 1976) is common and this enables the generation of task- independent descriptions. Whether this distinction is practically feasible is questionable as the production of declarative descriptions often involves reference to procedural knowledge and the two may be defined in terms of each other especially in the case of human knowledge. This technique does, however produce a body of useful generic action and object primitives and allows the description of salient entities and operations in any target situation. This approach is believed to be potentially very powerful as it describes the empirically

established requirements of the user as they exist in the task-state, yet it is not constrained by specific task issues.

#### **4.2.2 Functional requirements specification**

This type of technique aims to describe the requirements of a task or set of tasks in terms of functional attributes which the system must possess in order to perform the task. This type of description proscribes the necessary functions which must be offered in the design of an interface. This technique typically presents static, functional descriptions and aims to produce distinct descriptions of the system from a finite-number of functional attributes. The focus on tasks rather than user descriptions limits the possible types of function offered to the user but creates a description which more closely resembles the users eventual performance.

An example of this type of technique is Smolensky's 'Formalising Task Descriptions for Command Specification and Documentation'(1984) which describes an input and output model which aims to increase the effectiveness of mapping between tasks and tools. This tool enables a top-down mapping of mental onto procedural tools. Attributes of tasks are considered instead of procedures and design is proposed in these terms.

Smolensky proposes the use of an attribute encyclopaedia with a task-to-tool index to indicate appropriate tools for given attribute combinations. Development and updating of such a task-to-tool index would be very difficult as it would need to describe a vast number of semantic to syntactic mappings to be comprehensive. The process of the generation of real-world tools remains problematical for this type of description.

#### **4.2.3 Data requirements specification**

A number of formal tools currently exist which aim to describe HCI in terms of data. These

techniques identify the processes involved in interactive processes and describe the data structures and flows which are inherent in an application.

A variety of techniques exist which focus on data and these are typically used for implementing data processing systems although they can theoretically be used in all types of applications. One example of such a tool is the Systems Software Analysis and Design Methodology (SSADM) developed by the CCTA (CCTA 1983). This technique offers a set of procedures and notations for the capture of data descriptions of office applications.

The resulting output can be used as the basis for generating implementable computer code as the data structures and processes inherent in the application are made explicit. Data descriptions however are not very appropriate for representing the behavioural or interactive elements of HCI as they only refer to the movement of data and not to the task or interactive activity of the user. As a result this medium will not be utilised within this study.

#### **4.2.4 Dialogue requirements specification**

Interaction can also be expressed as 'a series of pre-defined dialogue statements issued by both parties eliciting a given response which in turn forms a stimulus' (Shneiderman 1981). It is practical to model user system dialogues in this fashion and use them to represent HCI, as only a limited number of states need be defined, along with their driving inputs, to specify the 'interaction space' (Gaines and Shaw 1984). Responses to given user inputs can be specified in this way and commands to support a given interaction repertoire designed. Using this technique enables rapid specification and evaluation of interaction sequences and the discrimination of legal and well-founded command sets.

An example of this type of model is Kieras and Polson's interaction model (1984) which defines interaction as a generalised recursive state transition network. The nodes in this model are user or



machine states and the arcs are the messages that initiate transitions between them. This enables specification of a series of user and machine states and the transformations, due to dialogue, which may cause the system to move between them. Each of the nodes and arcs can represent a number of instances of dialogue and hence can describe a number of implementations.

Many practitioners in the field of HCI would propose that this type of R.T. is sufficient for the specification of any interaction and it is often the interface designers function purely to create a interface which sustains dialogue with the user (Jacob 1986). This approach may not be sufficient if any cognitive or conceptual issues are to be considered and it is very rare that an interface can be designed with no reference to the structure of interaction.

#### 4.2.5 Task-based specification

It is difficult to describe the structure of interaction without a task-framework. This is due to the essential function of computers in performing tasks, or helping users to do so (Shackel 1982). Some applications, however, do not possess an explicit task-structure for some of the interaction activity, for instance, creative applications such as painting. However it is often an essential pre-condition for the structured description and formal analysis of interaction (Shackel 1986).

R.T.s do exist which are based largely on task-descriptions and Card, Moran and Newell's GOMS model is an example. This model aims to produce a task-description for the execution of a set of goals with a given interface configuration and hence allows assessment of an example of the performance of that configuration. The task-description facilitates modelling and analysis of 'the way users do things' which is essential in exploring the behavioural connotations of a given design, however it is not an ideal medium for analysis in itself. Task-descriptions are a useful medium for the resolution of high-level models but explicit task-descriptions are not necessarily analogous to the users intentions and are not directly implementable. A good representation technique also offers facilities for the mapping of higher-level descriptions onto lower-level ones, such as task-

descriptions, which are meaningful to users usually because they are based on instances of actual human behaviour (Smolensky 1984).

### 4.3 SCOPE, POWER AND ACCURACY OF VARIOUS TECHNIQUES

The power of the various specification techniques is derived from the pertinent choice of medium for the representation of particular user or interaction characteristics. Differing media will be appropriate for the representation of different types of application and will allow a focus on particular aspects of the design: for example, data-processing systems will require a notation which provides a detailed account of the movement of data. Also important, however, is the potential of the notation for describing any inherent structure of the application model (Jacob 1986): tools should have the capability to reflect synthetic constructs which exist in the real-world. Hence, even when implementing a data-processing system account must be made of the key *behavioural* determinants of the success of the system.

Knowledge-based techniques are limited by the distinction between procedural and declarative knowledge which in the real-world is difficult to rationalise, especially when the goal is to produce a system which exploits procedural knowledge. Procedural knowledge does remain essentially task-constrained in many cases so it may be difficult to derive general mapping rules. It is highly desirable to restrict the prototype to declarative knowledge for as long as is possible, to delay commitment in the design process.

Scope can be determined, to some extent, by the designer employing the technique at a chosen level of analysis. This is constrained, however by the sensitivity of the notation. Task-descriptions, for example, are normally limited in size to keep the model at usable size however sufficient acuity must be maintained to account for all the relevant details in the model. Dialogue descriptions can be at the sentence level, word level or character level depending on which is appropriate. With knowledge descriptions the level can also be variable, however it typically focuses on conceptual issues, as a

user cannot be said to 'know' about lower-level entities such as 'pixels' or the letters in a command.

#### 4.3.1 How the techniques constrain the design

Techniques can help the designer constrain the design space in a number of ways. Some techniques may offer the designer a concrete and complete configuration which can be converted immediately into a design or may totally define the set of transformations necessary so the the designer can create a comprehensive command set (Jacob 1986). Others, meanwhile, may leave the implementation process to the designer and the role of these techniques is to provide the designer with an accurate and complete picture of the users behaviour to assist the design process: this known as a 'scenario-based' approach (Shackel 1986).

The extent to which descriptions must support the actual creation of a physical interface depends upon the requirements of the designer. If the designer requires a specification which formally describes structure of the system then a dialogue model may be appropriate. Here, the technique develops a robust technical specification. However, if the designer requires a specification which provides a model of user requirements but remains flexible and open to interpretation then a knowledge model may be more appropriate. Here, the technique simply informs the designer.

The knowledge model aims to outline the users particular conceptual workspace and allows the designers workspace to be mapped onto it accordingly so it defines the boundaries or parameters of the users knowledge as well as its structure and content (Johnson 1985). The functional model aims to outline the set of functions or attributes of functions which could combine to support all of the possible activities within a domain.

### 4.3.2 Validity and Reliability of techniques

Specification techniques can be evaluated with the perspective of a number of types of validity, in a similar way to that which has been applied to metrics (see 2.3.1). These tests of validity are briefly described here in the context of assessing representational techniques.

Criterion-related validity would be based on the appropriateness of the type of media chosen for the representation of the constructs and entities found in a particular situation to be modelled. Choice of medium may be, to some extent, discernible from an *a priori* examination of the domain, however it can only genuinely be determined by in depth empirical study of the correlation between type of description and the users actual perceptions inherent in the environment. The construction of sample model to test this correlation is also desirable.

Content-Validity is the ability of the technique to describe a representative sample of this domain of HCI behaviour. If the tool omits important aspects or segments of a situation in the final description then it can not be said to be valid. This could be stated as the capability of the tool to capture and represent a comprehensive, complete and realistic section of HCI behaviour.

Construct-validity can be taken to refer to the appropriateness and realism of the synthetic constructs formed by the tool in describing interaction. For example a taxonomy may be useful in representing a knowledge tree but not a dialogue. The synthetic aspects of the tool mean that the designer can create an appropriate construct which should always represent behaviour accurately. There are, however, commonly recurring structures in HCI which means that certain constructs are used more than others. An example of this is taxonomic structures.

Specification techniques should also be reliable in that they should perform consistently over a range of similar situations. Test-retest reliability would in this case mean that a technique should produce the same description when applied to the same sample of HCI over a number of trials and

should highlight the same issues in the second trial. The technique should, however, possess the capability to distinguish between two similar situations on a level-of-analysis to be determined by the experimenter and should enable the designer to distinguish the two different situations.

Alternate forms or versions of the same tool should produce the same results to control for subjective impressions and the two halves of a representation should be consistent with each other especially with reference to similar items or entities modelled at any point in the test.

Establishing reliability may be problematical in terms of knowledge-based techniques. Consistency and repeat performance are perhaps not as desirable in this type of technique as a broad range of alternatives or variations in highlighting may be required. It may be difficult to identify one coherent description although there must be some consistency between the descriptions. As has been noted validity problems may exist in knowledge based techniques due to the distinction between declarative and procedural knowledge. This distinction may also lead to unreliable performance particularly with the task-to-tool mapping.

Dialogue descriptions are usually valid for the limited domains to which they are applied, however responses to given transformations must be consistent if they are to remain reliable.

#### **4.4 DEFICIENCIES OF CURRENT TECHNIQUES**

A promising approach to interaction modelling is one using meaningful behavioural and knowledge units as such units could describe behaviour naturally and realistically in user terms (Johnson 1983). A significant deficiency of current methods in this area, however, can be found in terms of declarative to procedural or high to low level mappings. Smolensky (1984) suggests that the range and possible forms of these procedures could be isolated independently and a predefined set of task-to-tool mappings offered. Smolensky's work however offers no realistic solutions to the characterisation of the huge device and task space. This deficiency is symptomatic of an ignorance of hierarchical interface issues such as the realisation of concepts in different forms at different

different levels and mapping of the flow of information through various communication channels or modalities.

The basis of successful representation techniques is a sound empirical foundation. Many problems arise in the use of these techniques due to lack of suitable tools for capturing data from the user. New, specialised behavioural data extraction and description techniques are essential as these form the basis for reliable interaction descriptions (Rich 1983). Techniques used must be able to embrace realistic human behaviour to be useful and this thesis explores the use of ethological data capture techniques to this end.

Most current R.T.s reveal little of the dynamic structure of a typical interaction other than its absolute boundaries and constraints (Alexander 1984). The current range of notations for the specification of HCI lack the vocabulary for the description of the dynamics of human behaviour. Alexander offers a technique for the specification of a dynamic multi-process dialogue based on CSP (Hoare 1974) however this technique refers to the input and output data streams and does not describe human behaviour well. The current notations often do not reflect the need to model behavioural aspects of HCI which underlie interaction - such as environmental considerations - or knowledge which is implicit in interactive behaviour. This thesis contends that this information is vital to allow designers to make decisions based on knowledge of interaction and seeks to provide tools for its representation.

#### 4.4.1 Taxonomic description structures

A useful description framework which can be exploited in HCI is that of the taxonomic variety so often displayed in the human sciences (Lenorovitz et al 1986, Fleishman and Quaintance 1987). This type of technique allows the classification of interaction concepts in a structured form so that a rigorous analysis of the interface can be made. If the relations between terms are founded on empirical evidence, that is the relations employed by the user, then a taxonomy can be said to

possess validity.

Hence, it would seem appropriate to model human knowledge in particular, and perhaps some aspects of HCI behaviour, in a taxonomic form. The work described later uses this form of representation to describe Human-Computer Interaction in an objective and concise way (Carter 1986). Names are agreed with the user to describe functions and this presents a classification scheme for interactive activity.

#### **4.4.2 How representation techniques can be made more flexible, powerful and realistic**

Whilst a metric's quality is assessed on the quality of its general evaluation criteria, such as 'complexity', and a model's validity is based on a number of factors including its accurate representation of given instances of interaction behaviour, a representation technique is evaluated via:

- The relevance of its chosen medium
- The appropriateness of the level of analysis of any structuring tools
- Its task and implementation independence.

A number of avenues can be identified for the improvement of current techniques:-

- Improving the accuracy of representation of task-related issues with more realistic behavioural models
- Incorporating facilities for mapping from declarative to procedural, semantic to syntactic or high-level to low-level descriptions
- Appropriate facilities for representing dynamic issues as the user perceives them

#### **4.5 RATIONALE BEHIND THE CHOICE OF INTERACTION DEVICE KNOWLEDGE FOR THE REPRESENTATION OF USERS**

Knowledge of interaction devices will be taken to mean knowledge of any feature, tool, equipment or property of the interface which is intended to, or in actuality does, assist the user in the completion of his task (Kieras and Polson 1984). A user action repertoire is assumed to be limited to the knowledge of properties of the interface possessed at a given moment during interaction. Usability problems are thought to arise when this knowledge is not sufficient for the user to complete his task. A description in terms of the features perceived by the user could be mapped onto the features actually offered by the machine and a support strategy initiated (Carter 1986).

The major advantage with this type of approach is that 'Interaction Device Knowledge' can be observed empirically, whilst it is being applied, and can lead to formally testable models (Carter 1986). This form of representation will be used in the empirical work to follow.

This chapter has examined the features of representation techniques which make them useful in the specification of HCI. The chapter has also highlighted some limitation of the current methods and outlined the criteria which would need to be met to resolve these shortcomings. Finally, the chapter has introduced a representation medium - Interaction Device Knowledge - and a taxonomy (Carter 1986) to describe users actions, which will be used in the empirical work.



## 5. MULTI-LEVEL ANALYSIS

The purpose of this chapter is to outline the need to model interaction in a hierarchical fashion so as to make it susceptible to analysis as a realistic and convenient model of the communication protocols between humans and computers. This chapter introduces the first dimension in the multi-dimensional model which forms part of the original work of this thesis. The concept of multi-level analysis is assumed to be one of the principle properties of interfaces which defines dynamic HCI.

### 5.1 MULTI-LEVEL ANALYSIS

The concept of the analysis of interaction by means of a hierarchical framework which stresses the independent study of user concepts, communication and behaviour as it appears in different forms is already a well-formed one (Moran 1982) and has already been introduced in previous chapters. This type of model has been identified as being useful to the designer and it has been proposed that its properties be reflected in a design tool to be described later in the thesis. This chapter aims to examine the rationale behind multi-layer modelling and highlight the role it plays in comprehensive and effective interface evaluation.

Sceptics of multi-level models will often point to the fact that the only point of true communication between the user and the machine is at the physical level, and will conclude this is the only relevant level of study for HCI (Buxton 1981). It could be postulated that the user merely passes a series of symbols to the machine and receives a set of symbols in return (Nielsen 1985). This would seem to suggest, however, that linguistic primitives play a large role in determining the type of cognitive activity is typically carried out by the user. Although HCC (Human-Computer Communication) is always constrained by the available set of communication tools, the design of a optimum set of tools cannot be effected without consideration of the type of tools which would be most effective in describing higher-level processes to the machine (Faenrich and Ziegler 1987). This would suggest

that a sophisticated model of the higher-level structures would be necessary along with a formal analysis of the communication protocols with the lower layers. Study of interaction must account for the sophisticated mechanisms which enable processing and interpretation of given symbols and patterns and enable the designer to equip the machine with the specific functions which would support those mechanisms (Norman 1988).

Many of the pioneering metrics sought to describe the higher level activity involved with the Human-Computer Interface in terms of lower-level primitives which could be observed at the physical interface, hence any description could be described in terms of manifest, explicit and observable behaviour. Card, Moran and Newell's (1982) metric was constructed from observable behaviour and was termed the Keystroke-level model. This type of approach did not seek to categorise, represent or describe higher-level or cognitive behaviour in any concrete or meaningful way, although it was empirically testable. It was assumed that higher-levels of behaviour were not relevant for study in depth.

A purely physical model makes it difficult for the designer to match the style the interface or the type of facilities and representations it handles to those which would support the users needs. It is often necessary to know about the user's higher level and domain knowledge (and information stores) along with the strategies that the user employs to map such knowledge onto the available primitives at a lower-level so that the appropriate concepts can be supported.

Interaction can be meaningfully represented in terms of a number of media, such as information flow or knowledge transfer (Shneiderman 1981). However, it would seem to pose significant problems for these media to be expressed simultaneously without some form of hierarchical partitioning. Multi-level models enable representation of the user in several different forms with descriptions at many different levels. This allows for a richness of description at all levels which highlights in a much more acute fashion the particular issues which are relevant to a given type of medium at that level (Clarke 1986). This allows an interface description to be partitioned in to

meaningful segments (as the partitions are user derived) which facilitate elementalisation of the interaction so that it can be studied at a finer grain-of-analysis.

Ideally, layer models also allow a point-to-point mapping of facilities between layers to identify simultaneous representations and to allow the exploration of various procedural implementations. Identification of mapping rules between layers allows each layer to be developed independently and then matched with its superior or subordinate processes (Moran 1984).

A major shortcoming of single-level models is that they have no mechanism for representing simultaneity or parallel activity and hence they do not account for any user capabilities for 'Concurrent Processing'. Humans can hold concepts which can simultaneously be stored linguistically or procedurally (Chomsky 1957) and it may be the ability to transfer rapidly between these forms of information which enables them to learn, store information and most importantly, communicate. So, in addition to the benefits derived in terms of acuity of description the designer is aided via a more realistic model of the interface. Temporal issues such as prioritisation, scheduling and simultaneity can be modelled and approached as they would appear to the user. Usability problems can be addressed by studying simultaneity in that 'Cognitive Overload' due to parallel demands on processing power at a number of levels; the timing of mapping and scheduling procedures can also be derived.

Many conventional usability metrics are essentially serial in nature and do not provide the facility for explicitly representing parallelism (or chunking) whereas significant parallel activity is often occurring. Multi-layer models embrace this concept and would make these discrepancies visible to the designer so that they can be rationalised in the interface.

So, data can be integrated from a number of sources via this type of tool: results from linguistic or behavioural experiments, as well as human cognitive attributes and knowledge-handling strategies, which are all relevant to the nature of interaction. A coherent model can be formed from a collection

of seemingly disparate results by their careful incorporation into the overall model.

### 5.1.1 Layer models and their relevance

The layer models (mentioned briefly in chapter 3) have gone some way towards embracing multi-level descriptions in the representation of the Human-Computer Interface. They provide a concrete and meaningful structural framework for multi-level descriptions and many provide translation rules for the mapping of information between layers. Such models enable designers to explore the representations of information and knowledge at a number of levels and design interfaces which reflect all of these different forms of knowledge: structured environments for the task level, specialist interaction languages for the linguistic level and customised interaction devices for the physical or behavioural level. These models also reflect some of the properties of intra-human communication and information processing systems which have been demonstrated in other fields (Norman and Draper 1986).

Many human communication deficiencies (including those involving computers) may arise from problems in the semantic to syntactic mechanisms in which the human's initial concept is not supported by the available vocabulary and protocols at a lower level. Communication may not be achievable due to constraints applied to a given concept in its legal syntactic representation at lower levels.

Humans possess sophisticated knowledge of the world and contexts which enables them to overcome these problems via interpretation of an incoming message. Computers, however, do not possess such knowledge and a detailed study needs to be made of the higher-level communication requirements of humans if computers are to be equipped with the adequate resources for handling user's high-level concepts. Layer models are a method for supporting such study and whilst current models have difficulty in effecting descriptions of sufficient complexity they do offer the designer a useful methodological and conceptual tool in the design of HCIFs.

Layer models do not offer a panacea to the problems of interaction modelling in the design of easy-to-use systems. They are often not currently quantifiable in terms of the real-world values which represent interaction (Young 1984). Current models also fail to account for dynamic aspects of interaction and perhaps more importantly do not always represent a working or operational description of interaction at any level.

Essentially, the relative value of abstract and real-world models must be contrasted. Current real-world models almost always fall short of a realistic and reliable representation in all but the most controlled of circumstances and hence, offer limited assistance to the designer even in their limited domain. Conceptual models present a useful framework and storage device which may not purport to represent real behaviour precisely but which can suffer from lack of validity. Real-world models concentrate on the provable, observable lower-level behaviours and aim to form models of what higher-level constructs can be supported (bottom-up) whilst abstract models aim to generate high-level or conceptual structures and elucidate procedural or lower-level behaviours which could support them.

The purpose of interface description needs to be addressed here. Supplying designers with deterministic models of interaction is of limited value as their ability to implement the system flexibly will be impaired. It must be assumed that system designers possess sufficient experience to interpret models which are transparent and non-esoteric. As a result the most important attributes of the model may be its accuracy and realism and the extent to which the model offers a complete and comprehensive picture of the interface: these attributes are where this type of model is ideal. This, to some extent, mirrors the top-down design process and can generate a number of possible implementations without constraining the design, however neither of these techniques can necessarily be said to be totally valid. Both types of tool (serial and multi-layer) can be said to be valid in their own circumstances but not offer a comprehensive representation.

Unfortunately, this type of tool suffers from one of the deficiencies that has been identified in many other tools. These are essentially static in nature and do not reflect the changing nature of the interface as a whole both throughout and between interaction (Christie 1985). The model of each of the layers and the changing balance of activity between the layers represents a snapshot of the interface at one point in time. This model is typically formed before interaction commences, and hence cannot represent the rich interplay of information, change in relevance of different types of information or simultaneous activity within layers.

### 5.1.2 Types of layer model

A number of types of layer model exist which support particular features of multi-level representation.

Moran's (1984) model offers the kernel structure, in terms of the fundamental types of knowledge and information handled, and this is, to some extent mimicked by the following:-

Nielsen (1984) describes the interface as a seven-layer set of 'Virtual Protocol' dialogues in which higher-level protocols can be defined totally in terms of lower-level ones. He emphasises the need for the precise specification of all the possible dialogue formats which could support the transition of the higher-level concepts into physical signals.

This model is a hybrid of dialogue and layer model and this proves to be a powerful combination as it offers the precision and realism of a dedicated dialogue specification technique but allows the division of types of dialogue into meaningful layers so that the design decisions about specific dialogue aspects can be made independently (and implemented independently) with entities at given layers communicating via a set of 'analysers' and 'realisers'.

Faenrich and Ziegler (1984) use a four-layer interaction model to assist in partitioning the

interaction components of a generic interaction model into level dependent segments. They evaluate working 'Direct Manipulation' systems and identify generic modes within those systems. This type of model is particularly useful for the evaluation of 'Direct Manipulation' systems as all such systems are based on a set of generic elements in order to achieve simplicity and familiarity.

Clarke's (1986) model forms an overall perspective to integrate the incorporation of material from a number of fields, hitherto unresolved, in the HCIF design field in a *unified* tool. Notable among these unrecognised sources of data is Clarke's attention to Psycho-Social issues via a component of his model. This enables the designer to integrate material obtained from a number of fields of psychological and social research which would otherwise be unassimilable. Clarke's model also stresses a Goal-Driven model of HCI which helps define behaviour in a realistic (i.e. User perceived) sense for its description in interaction models.

Watkinson's (1986) model seeks to provide an action description framework for interaction and aims to incorporate specific task or human problem -solving strategies to interpret conceptual descriptions. Domain information is also incorporated where appropriate via the Task-Method level. Task- descriptions are seen as being based on and filtered through a set of domain contexts and constraints before they can be accurately described and implemented in the syntax of the machine.

A different machine level hierarchy is explicitly and separately demonstrated in this model. Anthropomorphism is seen as undesirable in describing machine properties as it is misleading and unrealistic. A design tool should support accurate and appropriate terms for the description of the computers communication channels such as compilers and virtual objects to enable designers to conceptualise machines without suggesting abilities they do not possess. The computer communication system is not the same as a humans and it cannot perform the same functions, although this is the objective of the designer, and it should be described differently. This enables the designer to perceive the limitations of the computers 'communication channels'.

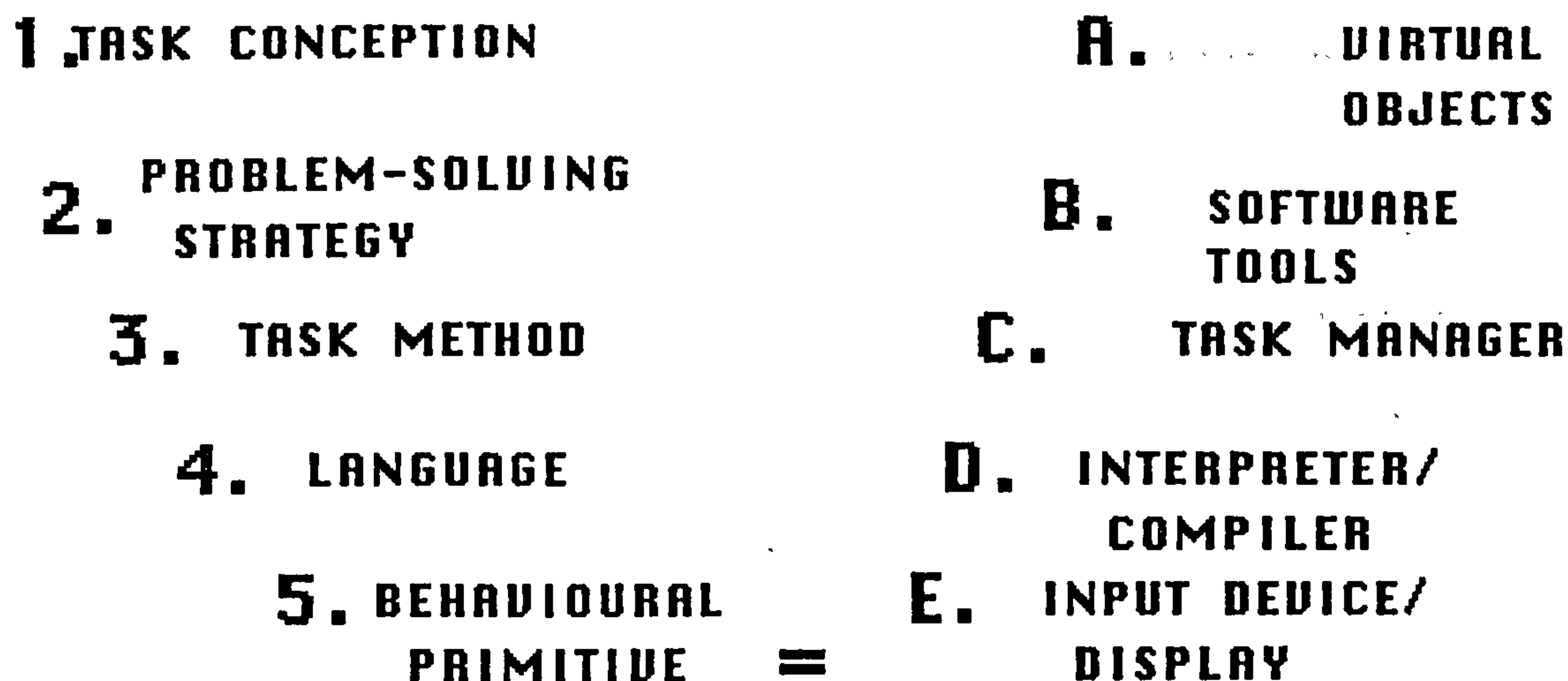


FIGURE 1. Watkinson's Five-layer model of HCI

Foley and Vandam (1983) offer a four-layer design tool which offers a structured method for the design of Graphical Interfaces: they see the higher-levels as a conciliation process between the designers concepts and the semantic constraints of software design.

Buxton's (1984) modification of Foley and Vandam's model adds a pragmatic component to lower-level design activities. This model is one of the few that diminishes the importance of Virtual Communication and hence it focuses on the pragmatic issues of inherent in physical communication bands.

## 5.2 HIGH AND LOW-LEVEL ANALYSIS

An alternative approach to focussing on one specific aspect represented by a given layer of the interface is that advocated by Cooper and Hockley (1985). They argue in favour of high-level analysis, that is, whole system level, as it is testable in the field and facilitates easy comparison between systems. However, they also argue in favour of a simultaneous low-level analysis as it can explain performance differences by allowing elements of an interface to be tested under controlled circumstances. This enables the evaluator to avoid bias so that test can be relatively sensitive and



the contribution of each of the features identified. Higher-level descriptions alone could fail to identify the relations between these parts, the whole system and environmental factors.

This measures the quality of interaction from both extremes and hence hopes to account for high and low-level features. This approach may facilitate the capture of most of the features isolated by a complex layer model with reduced effort. This is a less sensitive form of analysis than the layer model approach but is equally valid in the correct circumstances.

### 5.3 MULTI-LEVEL SUPPORT

A major interface design desirable which can be implemented through the use of multi-level techniques is that of multi-level support. An ideal interface will offer support at any level or in any form that is necessary to facilitate effective or 'easy' interaction. This may be effected by offering customised function at the 'Interaction' level or an appropriate interaction language at the dialogue level or even a structuring tool to help the user manipulate higher-level concepts (programming environment, for example). Support can alternatively be in the form of documented assistance at any of these levels.

Hence, support can be passive in terms of offering a palatable interface, or active in providing the user with a specific facility to assist with a specific problem. Support should be solicited in the more active cases as studies (Shneiderman 1986) have shown that users rapidly become irritated at unsolicited offers of support.

So, facilities should be available to support the structure and content of the users chosen style of interaction. Again, an easy-to-use interface is usually one which enables a user to adopt a method which is familiar (and which therefore does not make excessive demands on processing space) and natural and to display a style of interaction which is comfortable. All of these attributes are superfluous if the user cannot actually communicate the chosen task-structure to the interface which

means that support of the content of the user interaction is vital. Commands and facilities available should 'map' naturally onto the users chosen task-structure. This is the factor which largely governs many of the above aspects of easy-to-use systems. This notion of multi-level support can be implemented only with the assistance of multi-level analysis. Only examination of specific layers and types of interaction, along with knowledge of contextual and temporal constraints which apply can enable the designer to offer the appropriate, relevant and *well-timed* support.

Support can also be considered along a temporal dimension. The type of support which is appropriate at particular temporal instances during interaction due to a particular set of contextual constraints can vary enormously. For example, a user who is more experienced or familiar with a given machine will find support which is implicit or information rich useful as they will possess a specialised interaction vocabulary which will enable shorter descriptions of problem categories. This would suggest that different form of support should be available to users who have more experience of a machine, that is, they have experience of a longer total interaction-path (which can be assessed with a temporal value). This pitching of messages requires an analysis of the users level of experience with the system.

So, in conclusion, multi-level support could be implemented more readily if multi-level models could be further broken down into medial and temporal dimensions. This would require a new type of unified tool which could represent all of these aspects simultaneously.

### **5.3.1 The distinction between naive and professional systems and users**

The distinction between the support necessary for naive and experienced users and the implications for the differences in systems design deserve further treatment. The ability of experienced users to handle more dense and complex support messages systems, which cater for their experience level will often apply to different types of tasks than those typical for naive users (Kennedy 1977).

Experienced or professional tasks will often require a bias towards higher-level aspects of

interaction in the functions offered by the machine. Some tasks may be creative or novel and once the user is aware of the fundamental syntactic interaction paradigm of the interface support may be more appropriate in terms of task structure or novel aspects of task construction as well as provision of examples of previous interaction (libraries) or factual information. Hence, support demands can be very diverse and these discrepancies can be accounted for and modelled in a multi-level specification.

Ideally, a mechanism for the adaption of the support facilities is required to suit different users needs and the design of such a mechanism would require a method for the assessment and categorisation of users knowledge and experience. Users descriptions can be elicited using a specialised technique and its description along with some real-time framework. For example, it could be said that the user is at experience level 1 at time  $t_1$ ... so we can assume he will be at experience level 2 at time  $t_2$ .....etc.

Categorisation and description of experience levels has proved to be a hazardous process as experience can be specialised and can develop in different directions. A technique is required for modelling the assumptions and contexts which support a particular type of experience, that is, the rules for the users interpretation of symbols in terms of different assumptions.

### **5.3.2 Mapping of knowledge structures through the interface**

Another distinction which characterises the potential support which can be offered to experienced and naive users concerns the mapping of task or knowledge structures through the interface.

Support for higher-level structures in such activities as word processing, for example, involves offering such generalised structures as a 'document' for the configuration of the users activities ( or at the most complex a 'letter' or 'form' template) and the only relation this structure has to lower level structures is as vehicle for abstract, nominal data, for example, text. This is not the case with such professional activities such as programming where specific structures must exist at a

lower-level for the support of higher-level structures so that they can be communicated meaningfully to the computer in order that, for example, a conceptual description can be transferred into a dialogue one.

Hence, the usability of professional types system is, to some extent, determined by the ease-of mapping of such higher-level structures through the layers of the interface in order to facilitate communication of concepts and task-structure to the machine. This is on a similar level to those non-professional systems applications which support specific high-level structures for general applications.

#### 5.3.2.1 Usability as a facility for transferring high-level knowledge structures.

The usability of an interface can be defined by its provision of appropriate facilities or structures for allowing transfer of high-level task-knowledge or conceptual structures onto lower level descriptions for the purposes of describing a task to a computer (and the relative effort that this description incurs).

### 5.4 TASK-SPECIFIC INTERFACES

Multi-level analysis is essential in providing a model of the higher-level structures (and relevant lower-level support) which are required in the implementation of the above task-specific interfaces:

The distinction between the types of interface described in section 5.3.2 may not be, as was assumed a strict dichotomy, rather a continuum on which professional systems appear towards the top and general systems appear towards the bottom.

This would reflect the two major thrusts towards producing more usable interfaces. One thrust is towards the provision of task-specific configurations which offer particular constructs for the performance of specific tasks with given user groups in mind. The other thrust is the provision of

general facilities for the support of general or ill-defined tasks which require the provision of more open, widely-applicable or unconstrained frameworks. This second thrust is rather more problematical as it involves the definition of a generic kernel of constructs which can be useful in a number of applications. Beyond basing such general constructs on inherent user structures this process is difficult to specify and since such general constructs are limited in number the potential for future development in this direction is limited.

The scope for designing specific task-constructs is however only as limited as the number of applications which could possibly be implemented, hence enormous potential exists here. Since, such constructs are specifically designed for a given task or user group they are much more powerful and easy-to-handle as they are familiar to the user and they are knowledge and function rich.

The unavailability of a general model of a user makes the latter approach much more fruitful and easy-to-implement: specific models can also be recombined into a general model which reflects the commonality between particular applications.

## 5.5 'WINDOWS ON THE WORLD'

Essentially, interactive computer interfaces are generally more effective and usable if they are based on real-world situations, that is that they reflect and assist with the performance of actual tasks and their natural method of behavioural performance in the outside world. This is due to the users existing awareness of the properties of the knowledge constructs utilised, familiarity with the task repertoire and ability to recombine behavioural task primitives into novel interaction strategies. This does not mean that tasks are limited to those seen on the outside world or even to their implementation in the outside world, merely that they should be the product of familiar or typical generic behavioural interaction sub-units which can help describe those tasks.

Hence, the computer possess a number of channels or 'windows' onto the real-world through which the user communicates and it is the realism and appropriateness of the representations of the outside world and user, realised, for example, in supported knowledge constructs, and the resulting potential completeness of this view, in terms of the definition of all the users potential tasks, which determines the effectiveness of the interface.

This chapter has identified the need for modelling of interaction on a number of levels. As a result the model constructed in Chapter 13 demonstrates how multiple layers are required to present a complex dynamic structures.

## 6. FUNCTIONAL AND COGNITIVE ERGONOMICS

This chapter introduces the notion of ergonomic design and identifies this as the key design strategy. This chapter also introduces the concept of a cognitive workspace and refers to the process of optimising accessibility of function for users whilst performing tasks. The chapter highlights the variations in access requirements between individuals and groups of users at different points on the task-path and identifies this as a dynamic issue in interaction.

### 6.1 FUNCTIONAL AND COGNITIVE ERGONOMICS AS DESIGN DESIRABLES

Ergonomics is the discipline concerned with the design of artefacts which are especially appropriate for use by human beings. Ergonomics is applied in the design and construction of devices to be used by humans and by incorporating knowledge of the nature of human beings makes such devices more easy, efficient, safe or palatable to use.

The classical type of ergonomic study associated with the design and implementation of computer systems involved design of equipment such that all of the associated devices would be within physical reach of the user and that the user would not be forced to adopt an unnatural or forced posture in the operation of a machine. This notion now extends, however, to the design of a 'cognitive workspace' so that the resources required by the user are easily accessed by the relevant cognitive areas of the brain and that the machine does not force the user to adopt unnatural or inappropriate cognitive strategies or 'styles'. Additionally, the ergonomic design of computer systems is based around the cognitive limitations and capabilities of the user (Miller 1956, Baddeley et al. 1976). There exist a large number of similar human attributes, both general and specific, biological and behavioural, which can be referred to in the design of interactive systems to make the eventual configuration ergonomically sound.

There is a subset of ergonomics that is specified by the functional capability of a given machine. This will be described by the author as 'Functional Ergonomics'. This concept refers to the study of the means of access by the user of the purported range of functions which a machine possesses. This type of approach offers a particularly relevant perspective on the usability and utility of any given machine. One of the major issues involved with the usability of interactive systems is that of accessibility of function. A system is only useful to a user if the functions which it purports to offer can be accessed with reasonably few costs, in a reasonable amount of time and if the methods of access are convenient and helpful to the user. From another perspective excess functionality is useless if it is inaccessible. Hence, the functions offered by the machine must be within easy reach of the user and must be appropriate for the user's purposes. An important distinction must be made in terms of the difficulties which occur in the physical retrieval of the function to due an inappropriate configuration of the interaction devices and the problems which occur due to the user's inaccurate perception of the nature of the function.

### 6.1.1 Accessibility

Accessibility is an issue which has come to be highlighted recently in the design of computer systems. Traditionally very 'powerful' interfaces were designed with a large set of 'powerful' commands and facilities. Unfortunately, users were only able to sustain and utilise a small subset of a computers facilities due to memory and processing limitations (Eason 1977), hence a large number of commands are totally inaccessible and therefore the user had access to only a small portion of the functionality of the system. Similar commands were provided which 'Interfered' (Postman 1976) with each other in human memory and hence lead to application in inappropriate places, and confusion as to which command to use.

Additional studies have shown that users prefer to utilise a relatively small working command-set (Eason 1977) and are very reluctant to adapt this set so provision of excess commands would seem



to be a fruitless exercise. Users exploit their limited command set in a large number of ways to adapt it to tasks rather than learning new (perhaps more powerful) commands, and providing more commands can actually reduce usability.

Accessibility, when quantified, provides 'a reliable measure which correlates well with usability' (Shneiderman 1986). Accessibility is also related to 'perceived' usability although it is usually symptomatic of usability rather than a causal factor. Accessibility is determined, to some extent, by a number of underlying factors such as visibility, transparency and comprehensibility and such measures can be combined to form a measure of usability.

Moreover, perception of function is not only undertaken with reference to its value as an assessment must be made of the cost of a functions retrieval and this aspect related to any potential value. For instance, users will commonly try to preserve the ability to return to a 'stable' or 'safe' state at any time during interaction (Carroll 1982) without losing any of the body of work undertaken up to that point and they will be reluctant to attempt to access any function which may jeopardise that security.

Work has been done in eliciting the optimal state of perceived user goals (Malone 1984) as a determinant of interface palatability. If the users goal is of sufficient importance to motivate the user to access a particular (and incur the associated costs in terms of effort and risk) then the function will be accessed. This highlights the issue of the presentation of goals in the design of HCIFs.

The above definition presents some problems in terms of quantification or the description of effort in a meaningful form which would enable the assessment of user requirements. Current definitions are in terms of 'Mental Effort' (Moray 1979) or 'Cognitive Load' (Brooks 1980) where the first refers to steps required in the mental problem-solving process and the second refers to the demands on human memory. Both of these can be quantified but they are difficult to relate to interaction in real-human terms. The finding of Eason's (1977) work can be explained in these terms: users are

reluctant to embark upon productive thinking during HCI due to the large 'overhead' or demand on 'Mental Effort' which may exist in retrieving new commands or functions which would be incorporated in a new Problem -Solving strategy. Reducing those overheads would encourage users to embark upon more exploratory behaviour and hence enable them to learn both faster and a larger repertoire of commands for the description of their task (Levison 1979).

There also exists an important temporal component in the specification of users ergonomic requirements. The relevant attributes involved in the design of systems fluctuate throughout interaction and hence any specification system must have the capability to express these changes and incorporate them in the design specification. It seems clear that the relevant factors in an interaction will change as it progresses, so these emphases must be represented in any representational scheme.

### **6.1.2 Criteria for the assessment of accessibility**

Clarke (1987) offers a number of dimensions along which accessibility can be quantified:-

**Flexibility** - The number of routes to a given effect

**Complexity** - The number of choices at a given step

**Power** - The number of stages it is possible to execute with a given keystroke

These criteria overlap, to some extent, and a given criterion can be used to assess a given aspect of accessibility, however they can be combined to form a relatively accurate and comprehensive measure.

Quantification of these criteria could facilitate experimental evaluation of usability. The absolute number of steps required to access a function would contribute an effective measure - ideally as a ratio of the utility value - although difficulty could arise in formulating a meaningful definition of a 'step' in terms of user behaviour. This could be in terms of number of presses of the 'Return' key

or as identifiable, discrete and independent commands. Summation of these steps would provide a total accessibility value. Mental Effort could, for example, be quantified via analysis of subjective reports of effort and related to known generic examples in given contexts (Levison 1979).

Complexity may be assessed using the number of steps required to perform a task and the number of rules and assumptions invoked (Reisner 1982). So, experimental methods for the investigation of accessibility could involve a complexity measure which accounts for the absolute number of steps in a task with an initial qualification of step type in terms of temporal and effort values.

The power dimension could be assessed by establishing the potential number of steps which could be required to access a given function in comparison to a single, complex command. This dimension, however, should truly be measured using a more user-oriented perspective. Power would be more accurately defined as the ability of a command to achieve a given discrete effect to satisfy the purposes of the user. Hence, powerful systems can only be implemented when an accurate representation of the users needs is available in the form of a sophisticated user model. A command can therefore be said to be powerful if it displays the appropriate effect as assumed and perceived by the user.

It could be assumed that power considerations exclude complexity considerations, to a large extent, however the purpose of ergonomic design is to identify a sufficiently accurate and appropriate user description such that commands are both simple (easy-to-understand) and powerful. The simplicity derives from the naturality of the interface and the power from the appropriateness of the choice of command.

## **6.2 USABILITY AS DETERMINED BY ACCESSIBILITY OF FUNCTION**

Accessibility of function correlates well with perceived usability (Shneiderman 1986) as it is determined by the ease with which users can perceive and retrieve functions and objects. Since this

activity plays a major role in interaction as a whole it accurately mirrors quantitative values of usability in terms of effort and time. This is realised in the propensity of a user to interact with a given type of interface as a ratio of the potential value of that interaction (marginal propensity to interact). Interaction must be assumed to be purposeful for design to proceed and it is the identification of those specific purposes which provides a framework for design. If these purposes are met then accessibility concerns are met due to the naturality of the system and power concerns are met due to appropriateness of function.

A useful definition of the accessibility problem is in terms of the potential manipulation of the systems virtual objects. This can be stated on three levels:

1. Are they visible?
2. Can they be reached and retrieved?
3. Can they be manipulated to their full potential?

Assuming that the true functionality of the system can be expressed as the choice of virtual objects and the operations which can be performed upon them, then these three criteria comprehensively express the accessibility concerns involved with the system.

An additional issue which needs to be considered is the users ability to exert real and profound effects upon the machine. The user is the variable input into the otherwise closed system of HCI and it is the effectiveness of these inputs which determines accessibility. Commands may be trivial or minimal in some contexts. If the user is to allowed to be creative or novel then a deterministic system will not fulfil his purposes. True interactive and dynamic accessibility is required if creativity is to be supported and the full potential of interface is to be realised (Edmonds 1986). The user must be able to exert flexible, appropriate and directed influence upon the tools incorporated with the machine and must be made aware of that influence before and during interaction.

In summary, functionality *per se.* is irrelevant when not considered along with accessibility as functions are of limited use if they cannot be accessed, and indeed redundant functionality may actually reduce usability due to 'cognitive overload'.

### 6.2.1 The Mental or Cognitive Workspace

The nature of modern interactive computer systems means that many of the operations and activities involved with the tasks to be performed are concerned with mental or cognitive manipulation of data, information or knowledge using so-called 'virtual' tools. A virtual tool is one which exists only as a representation in the users mind and in the software of the machine and is not physically realised but which can be utilised by the user to exert a change in the data within the machine. Hence, the notion of a 'Mental' or 'Cognitive' workspace has arisen (Moray 1979) to describe the potential set of functions, operations and information, data and knowledge which can be manipulated using those operations.

Humans have pre-defined strategies for accessing such operations and applications which seek to exploit these strategies can be implemented more effectively if they are reflected in the structure of the HCIF. For example, a computer based calculator should operate in the same fashion as the users mental model as it equips the user with a set of pre-defined assumptions to determine interaction, such as the default use of infix operators (Young 1981). Also, explicit human cognitive procedures can be mimicked in the design of interfaces such as the use of mnemonics for the classification of data in the retrieval of information from a data-base (as they are for human memory) which offers a pre-defined strategy for information retrieval (Lansdale 1988).

Performance criteria for such simulations are easy to establish: they should perform as well as or better than the human counterpart in that, for example, the time required to access a database should be less than that to access human memory (Reisner 1984).

Reachability values can be produced using different criteria to that of physical systems. Temporal

values of reachability are common as they reflect the fact that access time usually increases along with difficulty of retrieval (Paap and Hofstrand 1988). The number of discrete steps in the retrieval process and the effort required within those steps is also a favoured measure (Moray 1979).

The size of mental workspace is related to the capacity of the human processing faculties and human memory (especially Short Term Memory). Hence, interfaces which demand the existence of an unrealistically large workspace are not usable as all the faculties may be taken up in storing the items and not manipulating them (Reisner 1982). The structure of a simulated workspace must also reflect the users real workspace with the most commonly accessed functions being placed in natural and readily accessible (in human terms): maps or routes are common device (Luria 1933). The most commonly accessed functions must have the simplest and most memorable but also the most natural indexing keys.

In addition to the 'virtual tools' manipulated by the user are the 'virtual objects' (Shneiderman 1986) which in this case are the relevant concepts and objects associated with a given task. These objects do not exist as they do in the real-world however as long as they behave consistently and realistically the user can interact with them as they would in the real-world. The tools and objects available are applied and controlled via a set of given problem-solving strategies or methods known to the user and stored in the workspace usually as task -descriptions. The problem-solving strategies should be implementable using the given tools and objects if they are appropriate. Hence, the strategies inherent in any interactive system should be those supported in the cognitive workspace.

### 6.3 COGNITIVE STYLE

A cognitive workspace will typically be unique to one particular user, or type of user, and will contain a set of tools, objects and notably methods which are individual to them. This unique set and the manner in which it is employed can help to define the 'cognitive style' of the user. This

notion embraces the preferred types and formats of the objects, functions and methods due to the prevalent structure of their cognitive workspace. This structure is in turn determined by the users experience of the world in general (and interactive systems in particular) and occurs as a result of preferences in task-structure, and the objects needed to support it according how the individual best, or most successfully performs the task in a palatable and natural fashion (Green et al 1983). Hence, Cognitive Style is a function of individual experience and is not necessarily based on an objectively economical task structure (in terms of lack of redundancy or least number of operations).

Many instances exist of improved human performance in various activities due to 'stylistic' cognitive or physical activity (Paap and Hofstrand 1987). Most sports display this phenomenon. Creative and motor activities allow for considerable interpretation in their implementation and performance, between individuals, and can be made perceptually simpler and cognitively more appropriate via the introduction of a controlled amount of redundancy by the actor to allow them to configure the task in such a way that it optimises the use of their best developed cognitive features or just reflects their particular cognitive configuration, in a general sense (hence, making it more natural). This type of strategy will offer significant performance improvements to the method of 'least redundancy' (Green et al 1983).

Real performance discrepancies in simple and complex tasks have been noted due to the application of cognitive style (Malone 1985, Loo 1978, Shinar et al.1978, Robertson 1982) and this would seem to be an appropriate vehicle in which to encapsulate the salient differences between user types and hence account for these in the design process. So reflecting the actual patterns of redundancy in the cognitive systems would enable us to identify where optimal models are inappropriate and study of users methods in the real-world counterpart of an application model would allow them to be accounted for in the interface design. The lack of objectively identifiable optimal task methods would suggest a move away from the aim of designing general purpose interfaces towards equipment dedicated towards particular user groups where known cognitive styles could be identified (Green et al 1983). Adaptive systems would make this type of interface a reality where a

particular user could configure the interface to his Cognitive Style at the start of interaction (Greenberg and Witten 1982). Elements of cognitive style supported could be 'known' visual search or feature extraction strategies, for example.

In conclusion, the focus in HCI design suggested by the above discussion would seem to be in matching the users existing way of thinking or 'doing' cognitive things.

### **6.3.1 Individuality of users**

Catering for individual user's cognitive style requires a differing type of empirical work to that currently favoured. Existing studies often aim to account for general human (often physical or perceptual) features in HCI design an activity which is based on traditional experimental methods for extracting data. These methods are sometimes inappropriate for the elicitation of data concerning specific user groups. Methods must be made available for the study of user groups, in controlled conditions or contexts, but sensitive to particular styles and interaction methods which help define that group. This thesis aims to offer a set of tools for capturing the salient aspects of user's behaviour and experience, through a variety of media, and in a range of environments for the purposes of implementing interactive systems which match user requirements. To implement the type of adaptive system described in this section a tool would also be required which outlines some of the dynamic interaction repertoire inherent within a given user group in terms of generic behavioural descriptions. This repertoire is addressed with the tools offered within this thesis.

### **6.3.2 User groups**

Typically, a large number of legal or possible methods exist for executing a given task using a computer. However, when actual user task-paths are studied both in terms of their true efficiency and their eventual employment by users then the number of practical or useful paths become identifiable and quantifiable: users do tend to use a limited set of non-optimal paths when



performing new tasks (Eason 1977) as they are reluctant to identify new methods, so there are an identifiable set of typical paths which can arise due to the users experience, preconceptions and many other environmental factors which can form a sound basis for a common task-path framework in many cases.

There may be manifold task-paths for achieving tasks successfully, however if it is valid to make assumptions about the canonical nature of a user it may also be valid to make some assumptions about the limitations on actual, legal and valid task strategies employed by users. Many of the problems inherent with assuming a canonical user can be overcome by using a kernel group of users as the basis for evaluation, in this way a useful set of assumed strategies can be outlined and the knowledge and information state of the user can be identified in many of cases. It must be realised that a perfect model can never be formed so the goal must be accounting for the greatest possible range of users in the model. This process would involve the study of the action repertoire and generic task-paths of a kernel group of users to form an approximate model of the user's possible behaviours. Most users should fall within this approximate model if it is sound and this is a matter of statistical probability involving the coverage the normal distribution of users. Barnard (1988) points out that all user models tend towards approximations as it is possible incorporate a vast amount of irrelevant user detail. The skill, for the modeller, lies in identifying the basis of the approximation.

### 6.3.3 Conceptual maps

A useful tool for the representation of users mental workspaces would be the 'conceptual map'. This form of representation would help the designer to externalise and model the structure, objects and activity in a user groups typical conceptual workspace. This map may be resolved as a network of objects and operations along with a set of control rules for combination (task-defined). The map gives the designer a meta-view of the workspace and facilitates the representation of the performance of the many objects and interconnections in an explicit, accessible and flexible form.

The model allows a realistic simulation of activity in the cognitive workspace via specific rules for the interpretation of novel tasks and data (elicited from the user group) by the existing objects and operations.

Work along these lines has been undertaken by Hitch et al. (1986) and Sutcliffe et al. (1986) on the construction of map based interfaces which reflected particular groups of users cognitive structures and significant performance improvements have been shown using this method. This type of description supports the representation of a large number of possible task-descriptions, namely the total combination of the task-descriptions on the virtual machinery.

Using map-type structures has been shown to increase the navigability of interfaces (Bolt 1986) thus increasing accessibility. This method may show such performance improvements due to the fact that it provides an explicit model of the interface structure which is usually learned over a period of time as an implicit interaction paradigm and may also be due to the humans phenomenal capacity for memorising routes (Luria 1923). The configuration of the map would reflect the real-world, with objects near to the faculties that use them and common groups of objects and procedures placed in close proximity to each other. Frequently used pathways or operations could be identified on the map and reflected in the design.

In this way a quantitative assessment could be made of interface performance. Cognitive maps will be discussed as a potential tool, later on in this thesis, for the support of the dynamic temporal calibration scales.

#### **6.4 DYNAMIC CHANGES IN ACCESS REQUIREMENTS AND THE DESIGN OF ADAPTIVE SYSTEMS**

Users will typically require a different set of tools to be readily accessible at different points in interaction depending upon the local demands of that particular task and an experienced user may

require access to a sophisticated set of tools which would not be appropriate for a naive user (Christie 1985, 1985). Indeed, it may often be prudent to restrict the access of naive users to commands which they do not require or which would confuse them, and which could potentially damage the system if used wrongly.

Such changes in access requirements are often definitive of dynamic HCI as contextual and experiential changes in the user and interaction environment lead to alteration in the perception of items in the interface, the perceived and actual requirements of the user, and hence the tools which could be seen to be appropriate for the point in the interaction path.

These changes can be anticipated and ameliorated by the design of an appropriate adaptive system using a dynamic description technique. The accuracy of the dynamic modelling tool is vital as the adaption of the system in the wrong direction or in the wrong areas can be disastrous. Introduction of new facilities should match exactly the users needs. Moreover, the most problematical aspect of designing adaptive systems is in maintaining their 'perceived functional consistency' (Dickerson 1988). Users are able to interact with systems, and indeed with the world in general, due to a perceived environmental stability which determines a set of working constraints or assumptions which govern their behaviour. If the system is continually changing then no such determinants exist and there is no framework for users to work within or from which to obtain feedback. Such issues are not relevant solely to adaptive systems as conventional systems could be strategically designed to offer a 'best-fit' model for the changing user.

#### **6.4.1 A method for plotting changes in access requirements**

A method is required for the identification and description of significant changes in user state which would necessitate changes in access requirements. This description should provide the facility for describing objective and absolute temporal locations as to where this should occur in 'real-world' terms.

This chapter has highlighted the need to plot changes in the users access requirements throughout interaction. The empirical work described later seeks to provide a modelling technique for this purpose.

## 7. INFORMATION FLOW MODELLING

The purpose of this chapter is to outline some of the evidence concerning the nature of Human-Computer Communication which characterises the humans communication channels with the world. This chapter also introduces the notion of a limited human information processing capacity and outlines the dynamic processing strategies which humans adopt to optimise this capacity.

### 7.1 COMMUNICATION CHANNELS

Evidence (Miller 1956) exists to suggest the existence of a number of discrete communication channels which allow the individual to receive information from, and transmit information to, the outside world. These channels to play an important part in the regulation of behaviour and the control of interaction with the outside world. It is sometimes helpful to treat the human being as an information processor for the purposes of designing some aspects of the HCIF. This has proved useful in integrating known cognitive phenomena and limitations into the design process.

Reception of information by the Human Information Processing (HIP) system occurs through one of a number of sensory mechanisms or modalities which detect and gather data and stimuli from discrete areas of the environment (Norman 1976). Two of these sensory modalities (vision and audition) perform preliminary processing on sound or light to make it compatible with, and in an economical form for, the cognitive centres of the brain (Newell 1965, Von Bekese 1953). The sensory mechanisms actually perform physical processing on the stimuli and configure their format to a state which is usable by the brain. For example, the bones in the ear act as a transducer and the cochlea transforms the waveforms into volleys of nerve impulses which can be decoded by the brain. Some theorists suggest that each of the sensory modalities projects onto a separate segment of the processing centres of the brain (Talbot 1984). This indicates some capacity for the parallel

processing of data from different senses. This may help to explain why humans are able to process information and initiate corresponding activity from different senses simultaneously. Other evidence exists (Newell 1965) to suggest some capabilities for the human as a parallel or rapid concurrent processor of information from various senses.

Current interactive technology exploits a small portion of a number of such communication channels and the majority of equipment relies on specific aspects of the visual processing channel with some element of tactile feedback (from the keyboard, for example). Recent ICON based systems have acknowledged that the HIP system is much more sensitive to eidetic rather than textual stimuli (Baddeley 1976) and that a hybrid of both types of stimulus exploit the HIP limitations more economically. Auditory channels have been used to provide feedback signals to the user, bells and buzzers, for example, however current work is being undertaken to realise the potential of auditory channels by implementing speech production and understanding systems with the future goal of creating not only speech-driven command systems but also fully-robust natural language understanding systems (McCauley 1984, Schmandt 1985). The design of Human-Computer Communication devices to exploit the full capacity and range of the given communication channels would seem to be a viable strategy for increasing the power of interactive systems.

### **7.1.1 Bandwidths**

The mechanics of the HIP system are such that the sensory communication channels are more sensitive to particular types of incoming stimuli than others. The visual system has in-built features for detecting lines (Hubel and Wiesel 1964) and the auditory system is particularly sensitive to clicks and loud noises (Von Bekese 1953). This has evolved from processing strategies which enable the human to shift attention rapidly to the most important, relevant or behaviourally significant stimuli in the environment (Sokolov 1976). The particular range of stimuli towards which the sensory modalities are singularly attuned to is described as the 'bandwidth' of the communication channel.

Bandwidths are fundamental properties of all communication devices and the visual processing system is attuned to a threshold intensity level below which neurons will not fire and this is when light levels are too low for 'rod' detectors to operate (Lindsay and Norman 1982).

Computer communication devices also display given bandwidths and in a well-designed system this encompasses the full range of all the communication activity that the user may wish to undertake. Work has been carried out (Card 1981) to investigate the appropriateness of particular pointing devices in communication with computers. The 'mouse' has been shown to perform significantly better than a 'joystick' and 'cursor keys', in terms of time-taken, at a given set of pointing tasks. This, it was hypothesised by the experimenter, was due to the fact that the mouse represented a better analogy to the natural human pointing mechanism and hence exploited the full bandwidth of the humans existing communication channels. Specific communication devices have been identified as appropriate for given tasks: for example, a graphics tablet is more appropriate for design applications as it allows the user access to the necessary sensitivity in a control of the cursor (Rheinhardt and Marken 1985).

### 7.1.2 Information processing limitations

Despite the existence of sophisticated information-handling and pre-processing facilities it has been determined that there are well-defined limits to the capacity of individual communication channels and their resultant projection upon the cognitive processing facilities (Miller 1957). This has led to the notion of 'Channel Capacity' which is a distinct limit on absolute information levels within any one modality, at any one time.

The processing centres of the brains can only operate at a finite speed known as the 'Information Rate' and can only accept new units once existing units have been processed. As a result of the limitation described Short Term Memory acts as a 'buffer' for the storage of novel incoming information prior to its processing. This buffering process is what makes the whole HIP system

viable as it can only work if it receives a constant supply of information about the environment in significant and meaningful chunks.

Beyond the processing exerted by the sensory mechanisms the HIP system has a number of powerful strategies for optimising the use of limited storage facilities and processing power. It has already been noted that information units can be of varying size or density and hence the HIP system can create larger or denser data objects than were originally presented by condensing them meaningfully into another form. This condensation activity is known as 'chunking' (Wicklegren 1956) and results in the creation of packets of information which are well-integrated and behaviourally similar and retain the essence of the real-world situation from which they were derived. This enables absolute information rates to be transcended by the continual reorganisation of data in order to maximise the contents of the Short Term Memory (STM) buffer. This memory buffer can be reinforced by continual repetition of its contents. This activity is known as 'rehearsal' ( Craik and Lockhart 1972) but it is of limited use in terms of time and the memory trace can only be maintained for finite period in this way.

Users are sometimes presented with large amounts of information in a dense format and are not allowed adequate time to process it. The time available is determined by the machine-pace (e.g. system response time) and this should be matched to the Information Rate of the users processing system. If the two rates are not matched a situation known as 'cognitive overload' (Levison 1979) can occur which means that new information is rejected or that interaction breaks down altogether because the user cannot form a coherent strategy. If interaction does persist communication is often on a very shallow level and the user cannot learn about the interface or execute the task in any level of detail.

Aids to Long-Term Memory (LTM) storage can also help with communication problems.

Structured presentation which allows the use of 'innate' human indexing methods, for example mnemonics, means that information can be retrieved rapidly from LTM and hence it can be seen as



an extension to the information processing system. Mnemonic based systems are currently under development (Lansdale 1988) and have been shown to allow the user to increase immediate processing facilities.

So, it is important to recognise that limitations exist within the HIP system and account for them in the design of interactive systems, by offering concrete support to existing human faculties, and in doing so help the user to optimise their exploitation of the said facilities.

### 7.1.3 Meta-information

One of the strategies adopted by the HIP system to optimise resources is the formation of hierarchical data models of the various novel data (Newell 1965). New items are integrated into an overall information structure which stores some of the salient qualities of its sub-units: such structures are termed meta-information and are, in effect, information about information.

Meta-information is in effect an abstract, representational higher-level model of an internal information state derived from existing knowledge about information patterns and handling processes. This may mean that strings of individual information items do not need to be stored one after the other, with the inevitable redundancy that this would suggest, merely the novel or local aspect of each new data item with the rest being assumed to resemble part of an existing model (Schank 1982). The utilisation of this type of model greatly expands the capacity of the HIP system.

Information can be stored under a series of meta-classes and by linking together similar items in terms of these meta-classes to form the most economical form of storage a type of 'relational database' can be created (Rumelhart and Norman 1986). It would be useful to identify the types of unit, categories, classes and strategies actually employed by given individuals to allow information presented by computers so as to be readily and easily assimilated by the HIP system. Also useful would be some of the processes by which the models are constructed and the temporal constraints

on information reception and access. These issues are addressed later in this thesis.

#### **7.1.4 The distinction between information and knowledge**

It is important for the author to draw a distinction between the nature of information and knowledge at this point in order to avoid confusion about the scope of the proposed model.

Information can be said to be generally abstract nominal and symbolic and simply serves to store given states or attributes in the world. Hence, it is a picture of the significant states and events of the outside world coded in a format which is compatible for the HIP system. Knowledge, however, can be defined more readily in terms of behaviour and experience (real and vicarious). Knowledge of a domain includes general rules based upon to existing experience and is capable of generating hypotheses about current and predicted behaviour. Hence knowledge can represent many different informational descriptions.

### **7.2 DYNAMIC INFORMATION HANDLING**

Humans are constantly required to attend to and respond to novel stimuli and, with the absolute limits on the processing faculties which exist, sophisticated strategies are required for the maintenance of a consistent complex data model in conjunction with the integration of new material (Norman 1976). The human system must be able to process data in 'Real-Time' (or as close as is behaviourally adaptive) hence, strategies are required to either manipulate limitations so that they can handle the required amount of data or slow down the necessary response rate (Lindsay 1982).

Human-beings are constantly engaged in forming new complex data structures to store the ongoing image of the world (Norman 1976). This is not the same process as, although it bears some relation to, the constant formation and reformation of hypotheses about the nature of events in the world. The latter is a dynamic knowledge model which is seeking to make sense of the events in

the world whilst the HIP system is merely a complex dynamic information structure which stores a realistic but abstract representation of the outside world. The formation of hypotheses utilises, to some extent the HIP systems facility to represent and store complex stimuli and information about the outside world but it builds upon these immediate representations to form longer-term knowledge models (Barnard 1988). So the HIP system is dynamically and constantly forming and re-forming a picture or model of the outside world to capture and maintain the consistency of its representation. This is achieved by defining items in the system in terms of each other or in terms of an overall meta-perspective. The information model can be rapidly amended or restructured by redefining the relational descriptions or re-classifying the items under different meta-headings (Rumelhart and Norman 1982).

The amount of ongoing data which needs to be processed at any one time is enormously variable depending upon the circumstances and can occur at various levels, in various forms and in various sizes of segment. The type of system which can process data on all these different levels and respond appropriately to the changes over time must, in principle, be a dynamic one (Barnard 1988). As has been suggested, the individual can manipulate the temporal component within the model to make it suitable by managing and adjusting time- constraints and by performing given operations in the optimal form to best utilise the limited temporal factor.

Hence, it would seem appropriate to treat HIP as a dynamic system in which temporal and logistical aspects should be modelled. Current models however treat the HIP system as a serial, atemporal, static processor which is somewhat mechanistic in nature and which cannot adapt its processing strategy flexibly (Roberts and Moran 1984). Sample temporal values are established but the models in no way offer realistic or 'deep' treatment of temporal issues. This is essence a descriptive model of the performance parameters of the HIP system, and whilst these are the only truly visible aspects, it does not seek to elucidate any of the processing structure or strategies or isolate the units, actions and adaptivity inherent in the system. All of these features would be of use in the design of interactive systems and they must be derived if realistic models of the

performance of the HIP system and interaction are to be formed.

The type and power of the predictions which can be made from current models is severely limited and amounts to no more than the insertion of a typical temporal value at given points in the interaction process. These temporal values are, to some extent, universal as they are based on common cognitive attributes, however they actually say very little about the predicted performance of a user in any given context (Young 1984). The problems of detecting and describing discrete human processing strategies in humans interaction with the world in general are manifold, due to the vast repertoires of human behaviour, however these are vastly reduced when the closed system of HCI is considered. Although the repertoire is still very large controls over particular types of activity executed by various types of users may reduce this repertoire to a experimentally viable size. Common task items in user repertoires can be expressed for example in the type of question such as: 'what does an accountant do first when faced with a new balance sheet?' There exist only a fixed number of valid and used strategies in given task domains.

Dynamic models of perception are common (Newell and Simon 1965, Neisser 1976, Selfridge 1965) with the individual assumed to form a number of competing hypotheses as to the events in the outside world. Hence, perception is deemed to be active and to involve explicit operations in the solving of perceptual problems. Newell's 'Demon' model of visual information processing shows this, however it only deals with the pre-processing undertaken by the sensory modalities and does not deal with the implicit, underlying operations performed by the human cognitive system.

Substantial work has been undertaken with a view to integrating current knowledge, and identifying new material, concerning information processing strategies, for use in HCIF design (Card, Moran and Newell 1982, Reisner 1977). However, this largely consists of the identification of static cognitive parameters which can be used to quantify simple interaction models. A specialised tool would be invaluable in the identification and description of dynamic issues which provides a coherent and unified approach to interface design.

### 7.2.1 Specification of user information state and requirements

A tool for the gathering of material concerning the typical states and constructs found in the HIP system would be useful. This thesis suggests that appropriate tools can be provided by adopting a user oriented perspective on HIP tasks such that they are defined for the user in terms of the task or the type of strategy used by a particular group or in a given environment.

Study and analysis of the completeness of the information model and the relevance and realism of the objects and structures described within it would enable the designer to hypothesise about the type of informational support which could be offered to the user at any given point in the interaction (Christie 1986). These support requirements would fluctuate dynamically according to the users position in the task and specific environmental conditions, hence there is a need for a dynamic model here. The specific structure of the model and its deficiencies in terms of structure and content could be identified and general and specific support mechanisms offered to remedy these problems. The interface should match the users chosen structure where it is efficient and new structures should only be suggested where learning requirements are offset by significant performance gains (Lansdale 1988). This process would require significant empirical investigation of user behaviour and a tool for these purposes is suggested later.

### 7.2.2 Temporal location of information access, chunking or loss points

Much of the dynamic activity undertaken by the HIP system is concerned with the selection and implementation of appropriate prioritisation, scheduling and ordering strategies (Miyata and Norman 1986). If these strategies can be identified, modelled and quantified then the specific informational requirements of the user at any given point in the interaction can be deduced. For example, if operation a. is a priority then information requirements x, y and z must be satisfied. Describing the behaviour which reflects these strategies is possible, however deducing the specific underlying meaning is much more difficult.

This type of temporal behaviour is important in defining the dynamic aspect of information handling and hence a temporal model of HIP activity in HCI modelling would seem desirable. This could be implemented as a multi-layer model of HCI therefore outline the informational state and activity at various levels. In other words the allocation of information to, and available processing strategies at, each level. This approach is more representative of dynamic HCI and is espoused in this thesis.

### **7.3 PROCEDURAL MEMORY AS A STORE OF INFORMATION AND KNOWLEDGE**

Procedural memory (Warrington 1954) is used as a method for the storage of procedural knowledge about how to perform tasks. This phenomenon suggests that this vast body of knowledge is not stored explicitly but implicitly as a series of behavioural sequences recorded in the so-called 'muscle-memory'. Alongside this knowledge is stored a large amount of implicit information which can rapidly be accessed by the individual reproducing the behaviour either behaviourally or mentally, in a similar timescale to STM. An example of this is the instance when someone cannot retrieve the information about how to spell a word from long-term memory but they are able to distinguish the correct spelling from the incorrect one when writing it. Hence, the correct morphology of the word is stored implicitly alongside the procedure for writing it. This can be exploited in the design of interactive interfaces. The WIMP interface allows the user to store the knowledge of the mechanics of the operating system in terms of physical actions, for example 'access that file by pointing the cursor at the ICON and double clicking on the mouse button'. A large amount of subsidiary information concerning the nature of the system is stored with these commands, hence releasing processing space for the execution of the task (Schank 1982).

Procedural memory (P.M.) is, in effect, another inbuilt channel of communication which supplies information from a procedural model of the world. Due to physical similarities between human beings and similarities between the methods they use in performing given tasks, the contents of P.M. is remarkably consistent, hence designs are able to exploit these consistencies and produce an

interface which is universally usable.

#### 7.4 INFORMATION AS CONTEXTUAL CUES

Much of human behaviour and the economy of knowledge-handling is based on human experience of given contexts (Norman 1976). Context can be very powerful in mediating human behaviour and shares properties in common with knowledge and meta-knowledge and meta-information. Context is difficult to define meaningfully, as it is a function of personal experience, but it could be described as a set of constraints on an environment elicited via experience which when met, to a given extent, will enable the user to make valid assumptions about the meaning of symbols or situations and to predict accurately the result of a given behaviour.

It is believed that the modelling of context and experience would provide particularly useful models of HCI behaviour as it would enable the designer of interactive systems to generate a simulation of a particular interactive situation and predict how the user will behave. This is done, to some extent, in Reisner's (1977) work in creating a simplistic simulation of the cognitive processes of the user. This simulation allows Reisner to make predictions about the performance of a given interface when evaluated against the simulation. The methodological problems behind the formal description and quantification of experience are manifold however these could be overcome, to some extent, by adopting a new and appropriate evaluative stance.

This chapter has presented some of the evidence to support the conjecture that human beings process information dynamically. This, in turn, suggests strongly that information processing in HCI is also dynamic. This forms the basis of a number of the dynamic principles identified later.

The constraints imposed by the human perceptual system also bring the users information state within the realm of empirical study. This fact, to some extent, confirms that the empirical work to follow is a tractable study. The area of study described within this chapter is characterised within

the information models which form part of the theoretical framework proposed within this thesis. This accounts for both dynamic information and knowledge models.



## 8. TRANSPARENCY AND VISIBILITY

This chapter introduces the concept of Transparency as a feature of good design and discusses the perception of interfaces and its role in determining user behaviour. This chapter defines the concept of Transparency as the ability of the user to interact directly with or manipulate directly the virtual image presented by a computer without referring to the physical structure of the interface. This chapter offers a number of methods for measuring Transparency and the concepts associated with it.

### 8.1 TRANSPARENCY

Transparency is a property of a tool which enables its user to access the tools functional capabilities, or manipulate its virtual tools without being explicitly aware of the presence of the tool or necessitating great effort or use of human resources. Transparency is property which is inherent in all well-designed tools (Shneiderman1986). When a human-being uses a tool he needs only be aware of the tool to the extent in which it aids him in his task. This statement is supported by Rutkowski's (1982) principle of Transparency: "The User is able to apply intellect directly to the task; the tool itself seems to disappear". The user should not be required to process superfluous information concerning the physical nature of the tool where it is not relevant to the functionality of the task.

The user should only be aware of the functional capabilities of any tool. An example of this is the use of a tool such as a pencil: The user of a pencil need not constantly be aware of the physical properties of a pencil when utilising it. The user need only be aware that the pencil will make a certain type of mark when employed in a particular way, hence the functional capability of the pencil is visible due to the transparency of the pencil as a tool. The pencil becomes an extension of the users arm and whilst the user will receive some form of tactile feedback from the tool the pencil

forms a direct analogy to the movement of the hand. The presence of the pencil is, therefore, not explicitly recorded.

Users must undergo a learning phase whenever a new tool is encountered (Fitts 1986), and hence there is a certain amount of learning time which is necessary before optimum performance is achieved. Once the properties of the device have been explored the tool can become transparent and the user need not relearn its attributes unless another type of task necessitates it or the previous learning session has been forgotten due to lack of practice. Users become experienced in the use of particular types of tool and particular formats will be favoured as experience is gained. Knowledge of the use of the tool will be stored in the users procedural memory so it need not be physically retrieved every time the tool is operated since the appropriate motor loop will merely need to be initiated.

The notion of a truly transparent tool must, of course, remain idealistic as there will typically be occasions where the user must refer to the physical attributes of a tool in order to improve performance by discovering new, and extending existing, modes of employment. Problems which will necessarily arise in the actual operation of a machine such as slips and errors will also require recourse to the physical state of the machine.

### **8.1.1 Transparency as a design desirable for computers**

Transparency has been identified as a design desirable for the creation of interactive interfaces (Shneiderman 1986) as it enables the user to devote all their cognitive and problem-solving faculties to the performance of the task and not to the understanding or maintenance of the interface.

An alternative definition of Transparency is in terms of the users ability to manipulate the virtual objects which define the functionality of a system (Nelson 1980). Virtual objects represent the notional conceptual entities which in conjunction with a set of generic operators allow the user to

execute their desired task-structure. Hence, the virtual objects should ideally represent or support the inherent functional concepts of a task or given set of tasks, in a form which is familiar or useful to human beings. For example in a word-processing system such as Apple's MACWRITE the type of virtual objects includes 'documents', 'menus', 'scroll bars' and 'icons' and the set of generic operators includes 'opening', 'closing', 'selecting' and 'dragging'. With combinations of these objects and operators it is possible to perform many of the tasks and sub-tasks involved with word-processing. The virtual objects in this type of system (Direct Manipulation) are perhaps more visible and explicit than in text-based systems, although in operating systems such as UNIX virtual objects (e.g. files, directories) and generic operators (e.g. move, copy) do exist. A good Direct Manipulation system allows the user to manipulate the objects within the interface in a similar fashion to that exhibited in the real-world. The extent of this manipulation is restricted by the inappropriateness of the current range of interaction devices, which offer little or no adaptation to the specific style of the interface.

Transparency is one of the key usability criteria which determines the accessibility of the functional capabilities of interactive systems (Hutchins 1986). If transparent interfaces are provided then the power of interactive systems can be accessed by naive and casual users rather than solely by computer professionals.

Another major factor which can define the quality of Transparency is that of feedback. Feedback needs to be 'rapid, incremental and reversible' (Shneiderman 1986) if an interface is to be seen as realistic. Feedback must be rapid enough to establish perceived causality for the user. System response to user action must occur within a given space of time if the user is to remain convinced that their action has caused an event in the interface. In a Direct Manipulation system the object must respond as rapidly as it would in the Real-world to maintain the illusion of reality. The maintenance of the perception of cause and effect is a fundamental principle of learning (Skinner 1933) and if the user is unable to determine the specific effect of his actions then they cannot learn about the consistent behaviour of the interface.

Feedback also needs to be incremental in that if a user performs an action faster or more intensely the system should respond proportionately. If the user moves the mouse faster more rapidly then the cursor on the screen should move equally rapidly. If the machine lags behind or pre-empted the user then the sense of control is lost.

Feedback also needs to be reversible so that if the user performs the opposite of a given action then the system should perform an equivalent reversed action. Finally, feedback should also be explicit in order to maintain Transparency. Users should be provided with manifest, explicit signals of implicit activity in the machine. For example, if the machine is loading from disc it should explicitly inform the user of this so that the machine state is Transparent. The user needs to know the machine state at all times with particular reference to the aspects which are concerned with the users task (Parnas 1981).

### **8.1.2 Accounting for transparency in the design cycle**

Transparency is a factor which offers many problems when attempts are made to quantify it or formally describe it. It is effectively a symptom of sound planning and is a function of the extent to which a system meets the users needs. An expeditious method for implementing Transparent systems is that whereby the user is equipped with a sound mental model of the system which represents a situation already familiar to the user and this enables the user to transcend the physical perception of the interface and interact directly with the virtual image.

Technology now exists, in the form of interactive computer graphics (Foley and Vandam 1984) to implement realistic simulations of real-world situations both in terms of appearance and performance. With the added robustness of the human perceptual system, which means that simulations only have to approximate to the actual situation (Thimbleby 1982), working representations can be built which perform in a functionally identical fashion to the real world and

hence these exhibit Transparency. An example of this is the photocopier control panel simulation device implemented by XEROX(1984). Named TRILLIUM this tool enables designers to create graphical simulations of proposed control panel configurations for the purposes of their rapid evaluation. However, these simulations typically perform better than the physical control panels they are designed to prototype hence all of the development could be done in software to allow rapid iteration and any real-world situation could be modelled with minimal effort. This type of design tool would allow the development of Transparent tools by iteration and the rapid amendment of simulations to maintain realism and incorporate new techniques.

The above examples represent actual design strategies which have been exploited however these metaphors offer little opportunity to evaluate Transparency at an early stage in the design process or before a working model has been constructed. A number of supporting criteria exist, however which correlate well with Transparency and which can be combined to facilitate abstract assessment of proposed designs.

## **8.2 VISIBILITY AS A DETERMINANT OF TRANSPARENCY**

One of the criteria mentioned above is 'visibility' and refers to the relative requirements of the user to be able to perceive objects at a given point in interaction according to task requirements. The features of an interface which may make it opaque to users are those which the user must explicitly store or manipulate but which do not play a productive role in the execution of the virtual task. The greater the number of such features the less Transparent the interface.

The ability of the user to perceive the 'Virtual Objects' and display items is governed, to some extent, by what the user is looking for at a given time and this is in turn determined by local processing demands at a given point in the task. Hence, a model of the requirements of the human problem solving process would help the designer predict what is visible to the user at a given point and indeed what should be made explicit at a given point. Perception and therefore visibility is

knowledge-driven (Neisser 1976), to some extent, and the knowledge which is applied in the solving of given problems will help direct the attentional processes (Newell and Simon 1962). Where no explicit problem-solving structure is evident users may resort to pragmatic assessment. This assessment could involve scanning the available items on the screen to determine which would suggest the most beneficial strategy.

### **8.2.1 The distinction between Visibility and Seeability**

A distinction between visibility and 'seeability' is merited by the role each plays in the definition of Transparency. Whilst 'seeability' refers to the users ability to discriminate an object on a given display in significant number of trials visibility is defined more accurately as the users consistent ability to to discriminate, recognise, perceive, identify, locate and remember a given object in a given location on the display, and in the interaction path, as a function of the users experience of the interface. Hence, the latter is a more useful criterion in the assessment of Transparency as it relates to the users perception of the object in the context of knowledge or a real-world activity.

The distinction between these two criteria is reflected in the problems accounted with their quantification. The former can be assessed using a simple tachistoscopic technique to identify threshold values (temporal and signal strength) for the users perception of the object on the display. Assessment of visibility would involve a more complex process. A temporal value could be derived for the total time needed to satisfy all the criteria which subsume visibility, outlined above, when faced with a simple identification task. Alternatively a self-report value could be obtained for the mental effort required to satisfy each of the criteria to facilitate comparative assessment resolved, for example in terms of the number of steps required to achieve the goal.

Both of the above measures are relevant in determining Transparency as they help define the characteristics of a particular interface which allows the user to readily perceive the systems virtual objects.

### 8.2.2 Active perception of system objects

Perception of system objects could, in the light of previous sections and work by Newell and Simon (1965), be assumed to be an 'active process' in which the user applies their knowledge of the task structure or the real-world analogy to interpret the possible meaning or function of an object. This is indeed consistent with a number of theories of human perception (Newell 1964, Neisser 1976) in which human beings are able to make sense of the world by applying their existing knowledge and experience to a given perceptual phenomenon in the context of a particular situation or activity. In addition work is currently being carried out to investigate the nature of the perception of interfaces (Green and Doubleday 1988) applying analogy theory to users interpretation of simulations. The flexibility that is necessary in human perception, however means that it cannot be constrained to one type of activity in making its assumptions. HCI is characterised by goal-directed and purposeful behaviour (Malone 1984) which makes the possible perceptual interpretations of stimuli much less diverse and allows the user to derive the meaning of symbols reliably from the task context. Hence, active perception would seem to be particularly appropriate for the type of activity which is displayed in HCI

If it is assumed that the user exhibits 'active' perception during interaction then it may be possible that the users mental state can be deduced from the ongoing task requirements and knowledge state at any point in interaction. Hence useful 'predictive assumptions' can be made about the users mental activity from the framework of the task structure and experience or knowledge of the user. If a model of the task or knowledge structure is formed in the context of empirically assessed real-world users behaviour then predictions about users mental states will be significantly more valid (Schank 1982). Tools are needed to draw knowledge, experience and context descriptions into models of HCI along with the dynamic component which would indicate where they are applied in tasks.

### 8.2.3 Transparency as an essential precursor to accessibility

Accessibility concerns have been established as being determined by the users ability to perceive and retrieve functions for their employment in a given task. Access to function, however, cannot occur unless the user can discriminate, recognise, perceive, identify and locate an object, hence it is necessary for a function to be visible before it can be accessed. Since Visibility is a major determinant of Transparency it should correlate well with Accessibility.

Access to function is also affected by the naturality of its location in relation to its role and importance in the users task (Moran 1984 ). The extent to which the location of the function is familiar to the user or is in a place determined by the real-world task depends upon the realism or appropriateness of the simulation or interface structure and therefore the accessibility of a function is governed, to a large extent, by the Transparency of the interface. Hence, any measures or techniques which can be seen to optimise Transparency will have the same effect on accessibility and will increase the available functionality of the system.

## 8.3 TECHNIQUES FOR ENHANCING TRANSPARENCY

Techniques are available for highlighting given areas of the interface and enhancing presentation of specific screen items (Shneiderman 1986). Such techniques could be applied more meaningfully if a temporal model of the users requirements were available. If a model could be formed of the requirements of users at given points in interaction then the effective addressing and resolution of these requirements would mean that transparency could be enhanced. Such requirements could be expressed in many forms and this would represent a point-to-point method of identifying discrete elements of the interface where Transparency could be enhanced. This model would be most useful if it could be based on real-world interaction behaviour or knowledge about interfaces. Only by possessing knowledge of the underlying structure of the user and the given task they are undertaking can specific requirements be addressed.



This chapter has outlined the value of transparent computer interfaces. The property is demonstrated in the empirical work to follow. The users knowledge of the systems virtual objects is seen to provide a useful model of dynamic HCI. The 'active perception' of various objects by subjects of different experience levels and groups shows how requirements for visibility of objects fluctuates according to experience. This reflects the hypothetical dynamic constructs introduced in this chapter.

**SECTION TWO: ORIGINAL THEORETICAL WORK**

## 9. CALIBRATION OF HUMAN COMPUTER INTERACTION

This chapter aims to outline the theoretical basis and the practical applications of the temporal modelling of Human-Computer Interaction. A three-dimensional model is introduced to assist in the description of the modelling technique and its role in the identification of variations in users behaviour is discussed. The mode of application of the model within the design cycle is discussed in terms of its utility to the designer and in terms of the types of systems it can be used to implement. A framework for the description of task-paths is described along with some novel data extraction techniques.

### 9.1 TEMPORAL CALIBRATION OF HUMAN-COMPUTER INTERACTION

Current models provide a good working representation of HCI without describing dynamic issues fully, however it is known (Cypher 1986) that a number of interactive phenomena do exist which are, to some extent, time dependent and these would be more amenable to categorisation, specification and implementation if an explicit model of temporal issues were formed. An example of this is system-response and the feedback issues associated with HCI (Shneiderman 1986). It is known that if certain temporal thresholds are not recognised in the design of interactive systems then interaction can break-down due to excessively slow or rapid feedback.

Temporal issues can be largely divided into two types for the purposes of this discussion. These include 'absolute issues' such as the temporal location of a given event during interaction. This type of issue is a useful area of study as it equips the designer with a typical temporal value for the display of a given behavioural phenomenon. This is especially useful in the design of adaptive systems but is also useful in the design of any interactive system as it allows the designer to plot the learning process of the user. Temporal considerations also include 'relative' issues of temporal description. Activities such as ordering and scheduling may profoundly affect the quality and

palatability of HCI and user behaviour has been identified within this work which indicate that there are pre-defined prioritisation strategies which can be associated with the users performance of many tasks, and incorporated in a design. Finally, the existence of simultaneous or perceptually indistinguishable phenomena, notably at differing levels of the interface, has been demonstrated by a number of workers (Miyata and Norman 1986). An explicit representation of such phenomena would enable them to be incorporated into the design process.

Any metric which recognises the changing user in its description would derive substantial benefit in terms of added realism if an explicit definition or framework of the temporal structure were to be included. Experimental studies performed as part of this work indicate that the user does 'evolve' during interaction and it is evident that the practice of relying upon a static User Model, which may be relevant at the start of interaction but not at the end, is unsound. Even perceptual interpretation which is assumed to be constant due to the structure of the perceptual system will fluctuate according to the users ongoing assumptions due to the fact that a large element of perception is 'knowledge based' (Green and Doubleday 1988).

One of the major deficiencies highlighted in the discussions of metrics, models and representation techniques was the lack of facilities generally available for the representation of real-world temporal issues in describing interactive situations. It was acknowledged that Card, Moran and Newell's metric ultimately produced a 'time-taken' value to give a comparative measure of ease-of-use, however this technique relies upon gross assumptions concerning the nature of users. These assumptions included a presumption of optimal methods and expert users and whilst this metric possessed a sound empirical backing the process for the application of temporal values seemed unrealistic. This sense of a lack of realism was compounded by poor definition of unit-tasks and a simplistic resolution of the 'Mental' operator. The value produced did enable some level of comparative evaluation but it did not produce an absolute value of usability. It was seen that many other metrics also embraced the use of temporal values (Reisner (1982), Kieras and Polson (1984)) for the quantification of their models on a comparative level.

So, whilst the Keystroke model does account for some of the temporal modularity of behaviour (unit-tasks) and uses empirically established task-units in seeking to represent a whole task, it provides no explicit representation of the temporal aspects of real-world interaction. The model formed uses tasks which seem unrealistic in terms of their applicability towards a specific user population; despite being described as 'optimal'.

It has been accepted that this model produces no more than an optimal exemplar value for the evaluation of a given interface and that that this represents an upper performance limit, however there would seem to be some potential benefit in the explicit modelling of deeper and more realistic temporal issues associated with HCI. This type of modelling would mean that a model could be formed of any level of user behaviour and that a fine analysis could be made of particular events along the task-path. Card, Moran and Newell's model is not a temporal one in all senses, it merely uses time-taken as a measurable value with which to quantify human performance.

It is believed that the underlying data gathering techniques used to capture HCI behaviour, to some extent, determine the constraints which apply on representation techniques. A description tool can only model the content of a HCI situation at any given level determined by the quality of the behavioural description with which it is supplied. This can lead to rigid and unrealistic descriptions of HCI situations. Hence number of potential techniques for the gathering of more useful data are highlighted later, along with a framework which supports more realistic and meaningful descriptions of task-paths and sub-task units involved with HCI

### 9.1.1 Types of scale appropriate for assessment

Temporal descriptions can be realised in a number of possible formats which enable the designer to highlight given temporal issues. Essentially, the application of an arbitrary or abstract temporal framework for the purposes of calibration may not necessarily identify the issues which are critical

in the development of a given interactive behaviour. To provide a meaningful representation any framework should be based on actual human behaviour and the calibration scale should reflect the behaviour of users. This means that the scale cannot be a rigid and inflexible device but should be based upon the particular behavioural and temporal patterns displayed by any user.

#### 9.1.1.1 Clock-Based scales

When representation of temporal issues is initially conceived the initial unit which is typically employed is that expressed by a chronometer; that is 'Clock time'. Time is usually expressed in the form of a series of arbitrary but regular units which aim to provide a regular and consistent form of temporal description. These units are, however, rarely derived from any meaningful value which occurs naturally in human behaviour and hence their appropriateness for describing such behaviour must be limited. Chronological descriptions are essential as they provide a rigid and quantifiable standard on which to base other types of data and offer an objective standard for human use, however they do not comment on the flexibility of the *perceived* temporal units of human behaviour (Ornstein 1977).

Card, Moran and Newell utilise clock-based units in their quantification of interactive behaviour and hence their metric is only able to produce a comparative value for behaviour on an objective scale. Their rigid adherence to a given task-structure allows rapid quantification of behavioural sequences, however only during the formation of task-units do they refer to naturally occurring human temporal units in any way and then only by producing an average chronological value. Since these units have no behavioural component they cannot be relied on as whole integrated entities for any other purpose than chronological modelling.

Repeated experimental studies (Ornstein 1977) have shown that humans rarely base their perception of events in the outside world (in the assessment of duration, for example) on chronological measures alone and indeed are quite unable to report reliably on chronological matters at all in many

cases. Hence, it would seem that each type of user and the inherent properties of tasks and behaviour include and innate temporal framework which should be identified and modelled if realistic temporal descriptions are to be made.

#### 9.1.1.2 Event-based Scales

Much of the work undertaken on the perception of time (Ornstein 1982, Doob 1953) indicates that duration in particular is assessed with reference to the events which occur within any given time-span. So, the number of events or stimuli perceived tends to be proportionally related to reported time. Hence, this would suggest that users will typically employ an event-based framework for the calibration of HCI. Interaction would therefore be more realistically be described in terms of the significant events which occur within an interaction path and any framework should be based around these. These events would be the most significant events from the users perspective and could be the achievement of a particular sub-goal, employment of a given function ('Login', for example) or some major event such as a system failure. Hence, significant temporal events can occur on *any* level.

Evidence, from this work and other studies (Miyata and Norman 1986) also exists to suggest that such interactive events are not necessarily independent or isolated and that inter-event effects can distort the perception of other seemingly unconnected events. This is a salient and definitive property of dynamic systems. Due to user learning and the necessary processing of interactive phenomena, at various levels, events early in the task-path may affect the perception of subsequent events. System failures or error-messages early on on the interaction can profoundly affect later behaviour whereas their occurrence later in the path may not be as profound. This is, therefore, a temporally constrained phenomenon. An experimental study with the purpose of examining this phenomenon is described later.

Event based descriptions will, by their nature, be partly functional descriptions as they will tend to

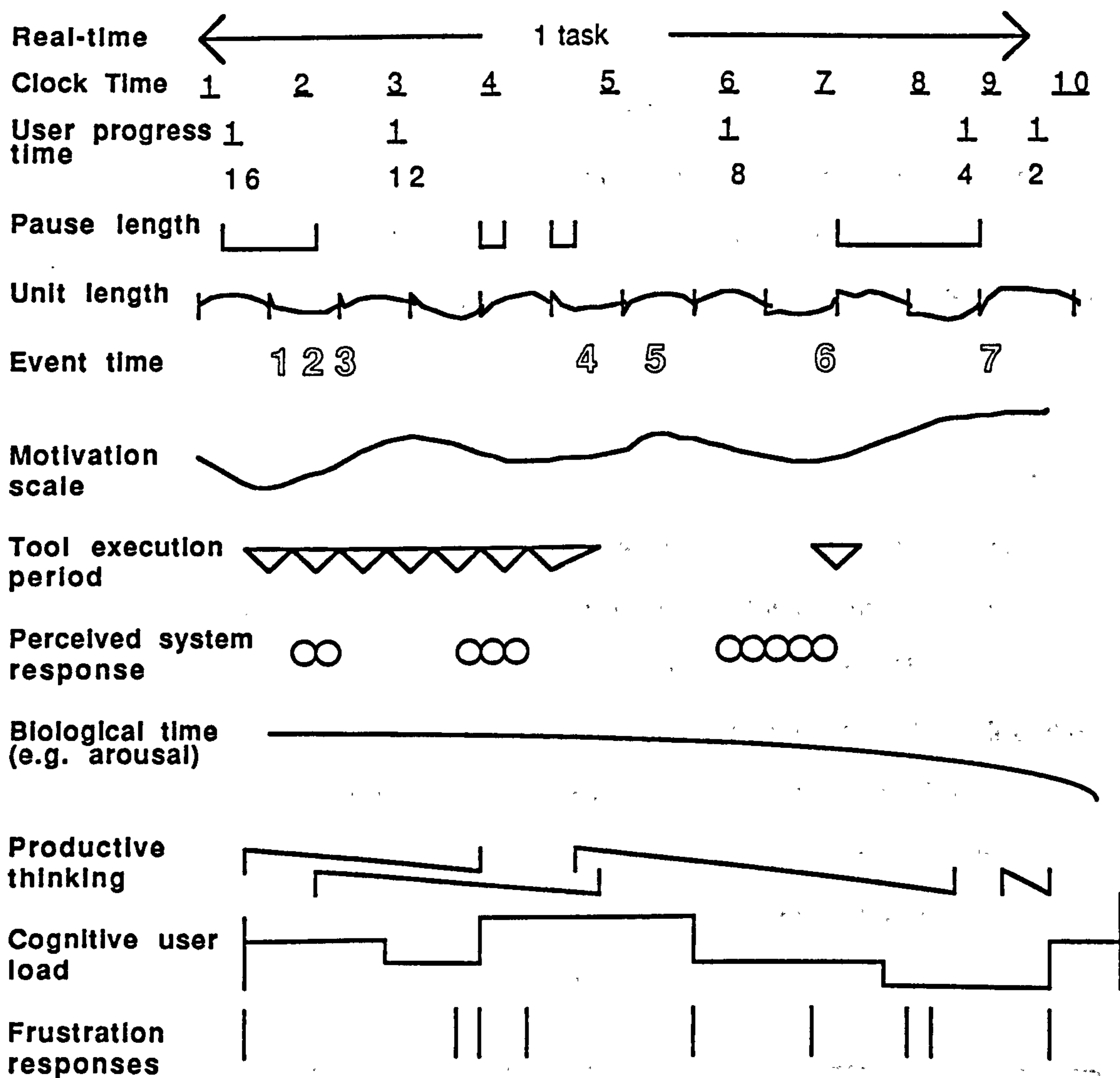
describe the users actions and will be based around the major task events. The task events themselves may not be synonymous with the users perceived events as the user will 'actively' interpret those events, however the users perception of interaction may be based around some attributes of those events. The task events can therefore offer a framework around which the users perceived events can be identified and modelled. This type of approach does not, however, describe any natural, inherent or innate activity within the cognitive system which may occur during interactive events. This approach is biased towards the external identifiable events which may be of variable significance. Since it is 'natural' human behaviour which is of greatest use in ergonomic design some purely user based data may also be necessary to form a realistic model.

### 9.1.1.3 User-based scales

Usable systems can be obtained by identifying, modelling and implementing natural human analogies in interactive systems (Shneiderman 1986). Hence, this would suggest that temporal descriptions should be couched in terms that describe meaningful temporal units and structures for human beings. A number of temporally based phenomena exist which, although not explicitly expressed in HCI behaviour, may implicitly affect its quality. Description of such user timescales would be useful in determining the appropriate machine-state for a given point in interaction. An example of this could be the drop in arousal displayed by users as interaction proceeds, which corresponds with work carried out on arousal in similar complex tasks (Hebb 1972), due to fatigue and habituation. This is variable among given users, however formal models of such deterioration could be formed with reference to user type and specified along a temporal framework its incorporation into a system. Hence, feedback signals could be pitched at a greater amplitude later in the interaction path: bells could be louder, flashes brighter, for example.

Some scales may be more relevant and more powerful than others, as their function in the determination of behaviour can be more profound, however it is believed that solely by examining the combined effect of a number of such scales that a useful model can be formed.





**FIGURE 2 SOME EXAMPLE TEMPORAL CALIBRATION SCALES FOR HCI**

The examples shown in Figure 2 are purely notional and the type of event which would be temporally described is likely to be more relevant to the particular task, however the examples provided outline the potential of dynamic multi-level temporal specification. The convergence of a range of types of temporal event can be modelled and their effects on user behaviour and the necessary interface requirements specified for a given temporal location. The events in themselves may not be significant, however when combined together they may decrease processing capabilities and increase demand on the said facilities to such an extent that interaction breaks down.

### 9.1.2 Simultaneous modelling on different scales

The above discussion suggests that it may not be fruitful to model HCI temporally on only one scale and that furthermore, if a comprehensive or complete model is to be formed, many alternative perspectives of HCI must be modelled. It is the temporal component, however, which facilitates meaningful comparison between scales as this represents the actual state of the given scales at a fixed point.

Meaningful temporal descriptions can occur at many levels of analysis. Conceptual or knowledge-based descriptions are no more appropriate than behavioural or dialogue descriptions within given circumstances. Many levels of description may be required for a complete description and it is often only by comparing activity at a number of levels that a realistic model of behaviour can be formed. Not only are a number levels required for complete description but it would be desirable if each level were modelled along its particular 'dimensions'. Examples of such dimensions could be that for instance at the knowledge level procedural and declarative descriptions must be accounted for. So, in effect what can be formed is a three-dimensional model of HCI behaviour with a entity or event being modelled according to its positions on three simultaneous scales.

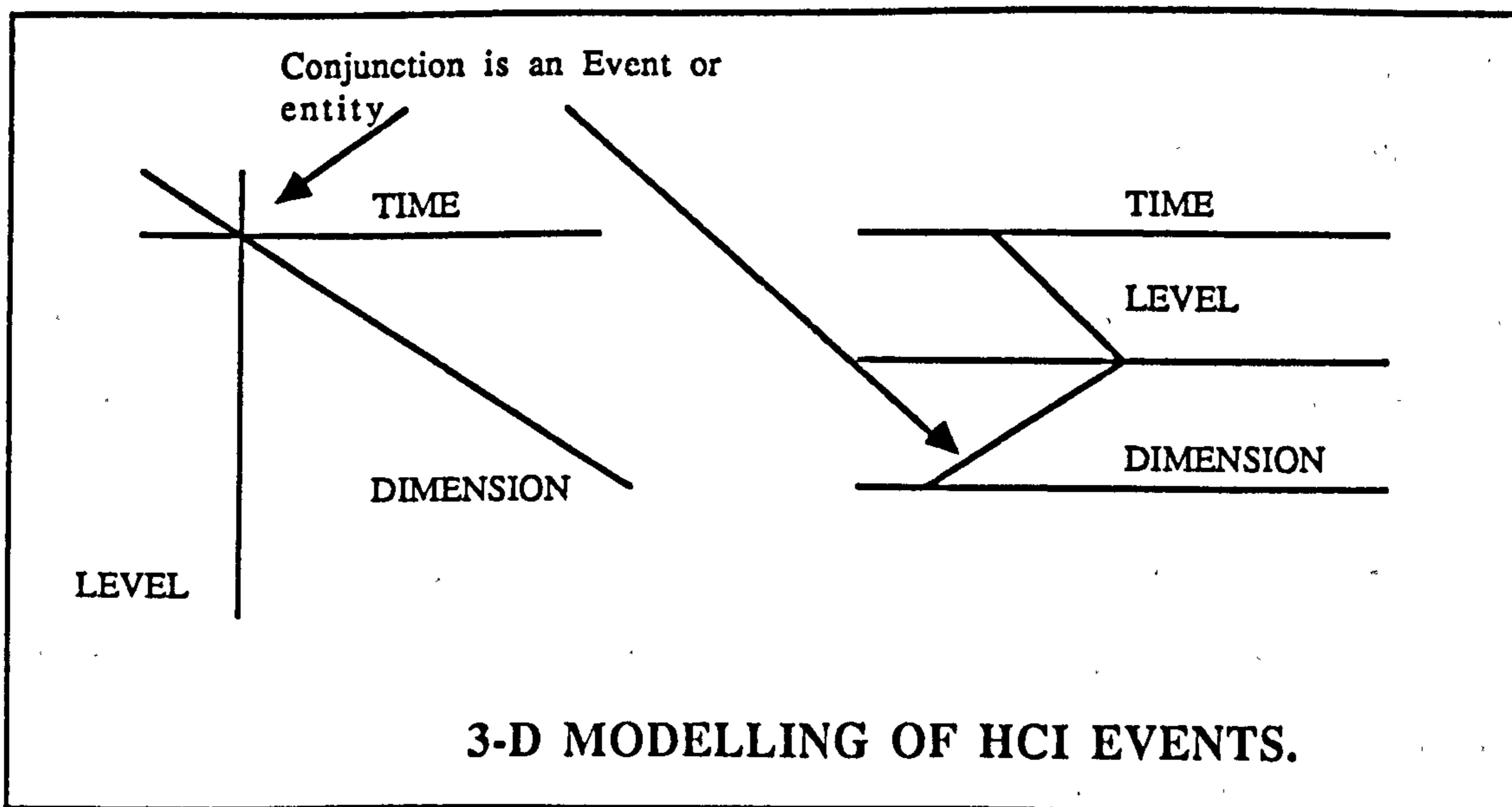


FIGURE 3 The Three-Dimensional Model

This description would seem to resemble the existing layer models, however the distinction is quite significant. The temporal bias of the descriptions means that an interaction structure is inherent in the model and therefore the conjunction of the dimensions means that a given event on the task path is being modelled in terms of a given dimension on a given level. The temporal element links the dimensional and level descriptions to real-world events or task-paths and hence the model can be quantified. Thus, events are naturally viewed from a number of different perspectives according to where the designer sets the scales. Hence, simultaneous activity on various dimensions and levels can be specified and it can refer to and control the same event.

#### 9.1.2.1 Simultaneous modelling as the basis for for describing priorities and scheduling

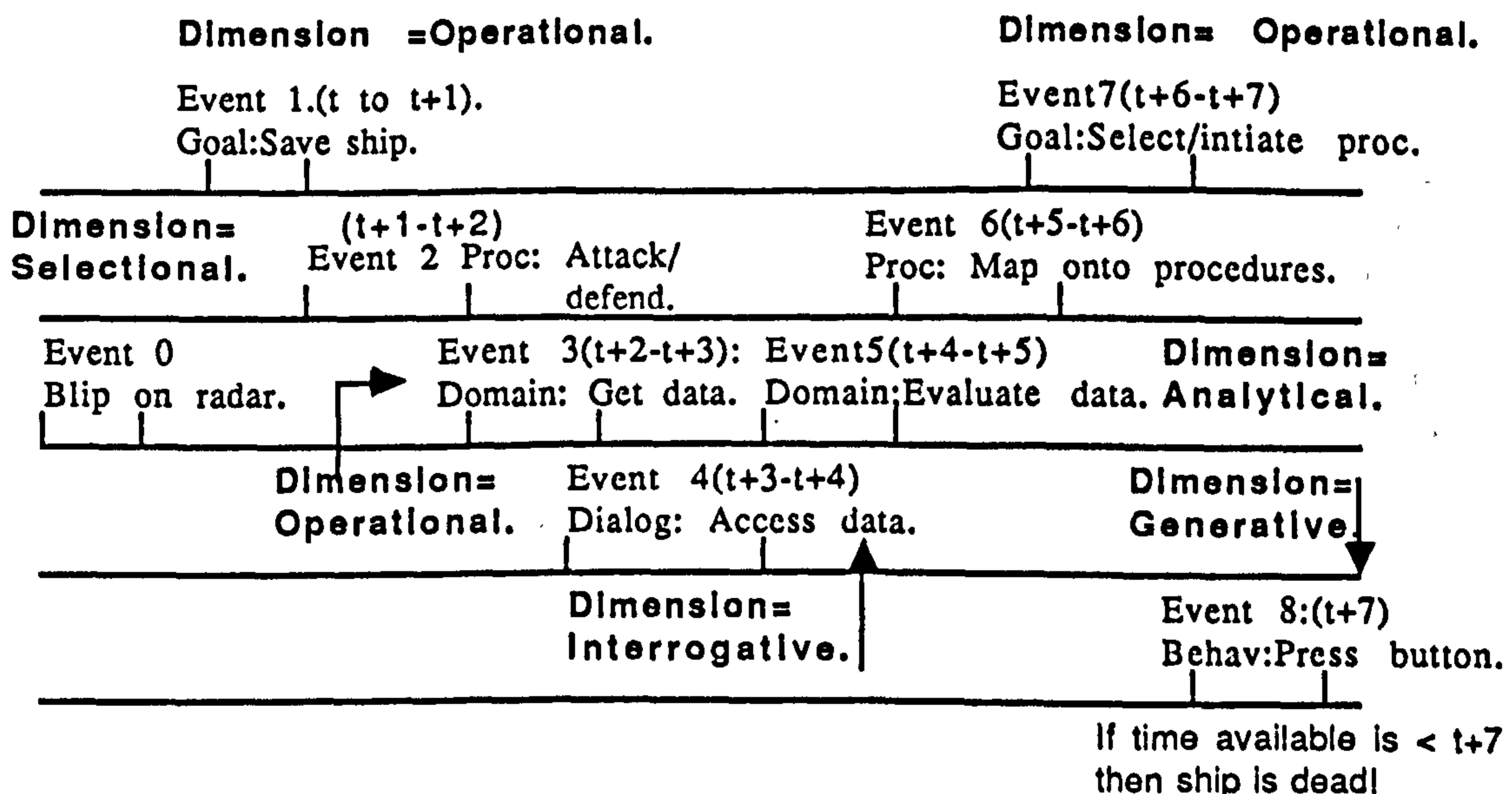
Simultaneous modelling on a variety of scales will enable priorities and ordering considerations to be represented within and between levels. Data concerning behaviour can be extracted by studying users performing a particular task in terms of the 'Generic Interaction Behaviours' typically used by that user group to perform the task. These units can exist on all levels and can be represented along a number of dimensions in relation to their role in the task. Once the total kernel of such units is established for a given activity on the task-path it can be reinterpreted in terms of the manipulation or exhibition of the Generic Interaction Behaviour along the task-path and priorities and scheduling strategies outlined within layers within given user groups. Subsequently, the multi-layer model can be formed with reference to these individual scales and intra-layer scheduling and priorities displayed. Manipulation of different types of material, for example information and knowledge can be specified at each level along with the operations which can be performed upon them.

Generic behaviours may not always appear to be completely discrete however the overlap between such behaviours can be defined as part of the task and scheduling strategies will still be discernible. Simultaneous activity on a number of levels can be quantified in terms of its total demands on processing space, for example, and the soundness of a given interface design evaluated in terms of

the quality of the design in meeting the inherent generic task behaviours used to operate it. The natural scheduling processes exhibited by the user can be elicited during the data gathering phase and inter-unit dependencies expressed as priorities and schedules. New models can be generated to match or optimise the design to meet these behaviours without overloading human faculties. Tools for the support of the user can be identified at each level and dimension and implemented independently.

### 9.1.2.2 Some example scheduling strategies displayed on a scale

A useful example is one of set of strategies in a real-world situation where the temporal factor is at a premium: 'hard' time constraints in real-time systems terminology. An experimental example will be offered later, however an appropriate 'anecdotal' example which would provide a good vehicle for discussion would be the design of an interface for assistance in determining strategies in a naval warfare context for shipboard handling of incoming objects, the useful prioritisation strategies exploited by current personnel to handle such events and how can these be supported and optimised with an appropriate interface design.

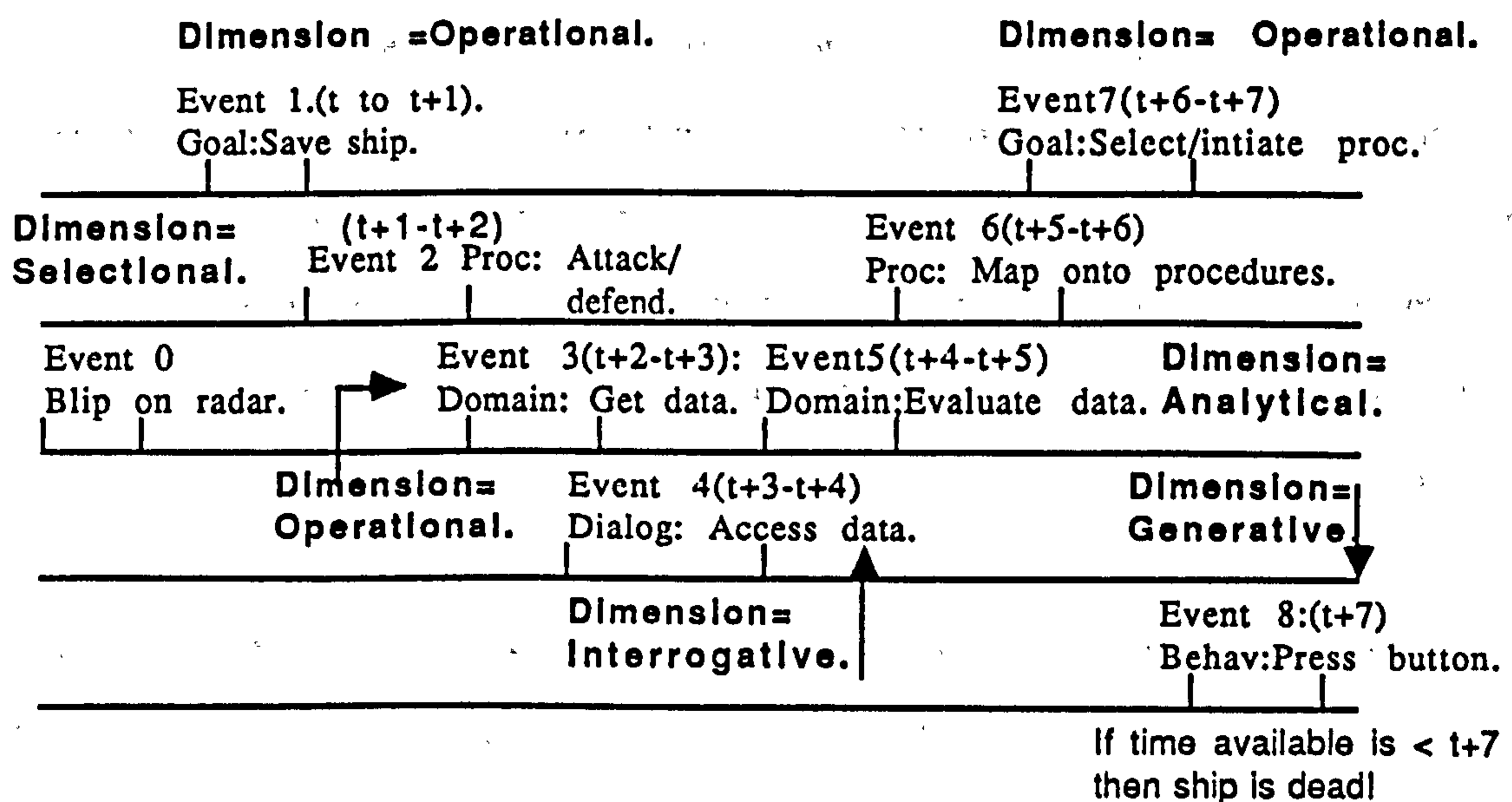


**FIGURE 4 A GENERIC ACTIVITY MODEL FOR SHIPBOARD THREAT MANAGEMENT**

Figure 4 represents a three dimensional set of scales for describing the temporal aspects of such an exercise and an exemplar resultant knowledge requirements description. The above example uses Watkinson's (1985) five-layer interaction model as basic framework and describes a hypothesised sequence of events including generic behaviours specified in terms of:-

1. Level of activity
2. The dimension within that level (for example, generative).
3. Implicit ordering constraints (from left to right)
4. Symbolic absolute temporal values ('t+3-t+4', for example).

A knowledge requirements model generated from the above situation is described below and outlines some of the required functionality of any interface designed to support this type of activity.



**FIGURE 5 A GENERAL KNOWLEDGE MODEL FOR THE SHIPBOARD EXEMPLAR**

## 9.2 TEMPORAL DESCRIPTION FOR ASSISTANCE IN HCIF DESIGN

A number of distinct features of HCIF design can be assisted by the provision of an explicit temporal model of HCI. These features form one of two types: firstly, control over the temporal performance parameters of interactive systems such as system-response, adaption rates, display rates, input device rates and animation speed among others. Secondly for the location of discrete events in a given interaction path to enable appropriate provision of adaptive changes, support facilities and tutorial support.

### 9.2.1 Machine-Pacing

Research indicates (Norman 1976) that users are able to interact with computers at a given pace determined by their information processing abilities and this is, to some extent, mediated by their experience of, or familiarity with, the interface or the application domain. Provision of an interface which exceeds these preset limitations by operating too rapidly can lead to 'cognitive overload'. Poor interactive timing, where the user does not have time to perform necessary mental activity before feedback can arise with inappropriate pacing. Mis-perception of feedback can occur when feedback cues are missed or if response is too rapid. Timing of feedback from interactive devices is critical as the illusion of causality can be difficult to perceive at a high speed of animation. Also the rate of machine adaption, in self-adapting systems, can be inappropriate for a users requirements.

Machines which operate too slowly can also cause problems for the user. These can include user frustration and lack of motivation. Poor interactive timing, where the user is left to think and loses concentration is also a problem. Mis-interpretation of feedback can arise if the user believes the machine to have failed and as a consequence reinitiates interaction. Feedback from interactive devices or animation sequences may be so slow that the illusion of causality breaks down altogether. With adaptive systems the interface may adapt too late so that the user is employing sub-optimal strategies for a long period.

An explicit representation of the temporal aspects of HCI which lead to these problems is required if they are to be resolved in the design of the interface. Design decisions concerned with machine-pacing could more readily be made if designers were equipped with a model which specified the inherent pacing criteria relating to a given user population. This model could be used as a vehicle for the derivation of such criteria if the task descriptions incorporated realistic generic temporal interaction data generated by the target user group.

### 9.2.2 Location of events

A number of the features of some interactive systems are only effective depending upon an appropriate facility being available to the user at a given point on the interaction path. For example, 'spell-checking' within a word-processor facility is only useful to a user after a sizeable document has been created. Hence, some means is required to specify the absolute location of such requirements in a temporal sense. This can be combined with representation of the demands of generic task activities such that the occurrence of an activity can be quantified as to its temporal position.

Descriptions can take two forms. Events can be described in absolute terms: for example, they can be said to be at time  $t$ , or two-thirds of the way into the task path. Alternatively, events can be described in a relational form by indicating that event  $y$  occurred before event  $z$ , or after time  $t$ . These types of description can allow typical absolute parameters for the possible temporal location of an event to be expressed or descriptions of possible ordering, and scheduling strategies typically employed to be described, respectively.

This type of locational information could be very useful in offering a framework for the specification of a users specific requirements or the representation of a knowledge state in a meaningful sense with reference to a task-structure. An analogy to this is specifying the users

position on a learning curve and customising support facilities to support that level. It is useful to consider experience as one long interactive session punctuated by breaks. Evolution of the user due to learning can be said to occur *during* interaction and 'regression' (due to forgetting) can be said to occur *between* interaction. Learning will occur at various rates due to user differences and these rates will be specifiable in terms of the specific experience gained by the user and the level of employment of that experience. Hence, a continuous model could be formed of the user according to the estimated level of learning achieved due total experience qualified by 'regression' models. This would enable the incorporation of a great amount of the psychological empirical work devoted to understanding the basic principles of learning (Skinner 1932, Bruner 1955) and the specialist work devoted to the learning of computer systems (Du Boulay 1981) into novel task models. The method for forming such a model is defined later.

### 9.3 GROUP REPRESENTATION THEORY

The use of controls over the proposed user group towards which designs are targetted is a critical issue in the measurement of the usability of proposed systems. The type of controls to be adopted within experiments could take many forms, however if they are to be of specific use in the design of a given application they should comment in some sense on the actual performance of a relevant task by typical members of that group. Hence, the ideal approach is a functional one, to some extent, as this reflects the nature of interactive systems and focusses the range possible user group attributes to those which are likely to be of assistance in the construction of a working implementation.

Card, Moran and Newell adopted controls by focussing on the performance of expert users and presumed that this would allow comparison between proposed designs by defining a standard upper performance limit for all systems. This, however, did not comment at all upon their proposed 'user population' or provide any interesting information which could be applied in any real sense about the possible nature of the interface. Their model was, in essence, a general human processing



model, however its grain-of analysis was limited as the attention paid to specific user groups was minimal. The general processing model is a robust one as it relies on universal human attributes to ensure that the model applies to all users, however it remains a static and deterministic model of interaction and hence is unable to comment on the underlying rationale behind any specific task or user group behaviour.

If specific and accurate predictions are to be made about human behaviour, however, a more comprehensive and realistic set of assumptions about the user should be incorporated. This thesis assumes that common behavioural strategies can be identified within user groups in terms of the type of generic activity they commonly employ and tools are offered for the identification of such common behaviours. User groups can be defined by the typical type of computer task undertaken by that group in the past as determined by existing machine constraints and the task constraints of the generic repertoire of behaviours in their particular restricted task domain. It is hoped that by modelling these aspects of user behaviour reliable predictions can be made about the behaviour of users performing given tasks with given interface configurations.

This technique would allow the incorporation of typical and 'known' user methods in terms of actual task and interaction descriptions. This would enable the explanation of those so-called 'deep' phenomena (Kidd 1984) displayed in human problem-solving. Such phenomena are those dynamic HCI behaviours which cannot be explained with reference to observable interaction behaviour and rely upon some prior knowledge of an underlying task-structure, interaction paradigm or world knowledge for their interpretation. These behaviours may arise in the field of HCI as a result of particular combination of user, task, machine and environment (Shackel 1982) via the modelling of actual real-world combinations and the resultant behaviour. They could also arise due to spurious and spontaneous environmental factors which could be controlled for and identified.

This process shifts some of the onus in terms of producing realistic models towards the behavioural elicitation phase and as a result appropriate techniques must be developed for the capture of

sophisticated behavioural data. Suggestions for these techniques are outlined later in the chapter. This approach holds much potential for the future as once specific user behaviour can be categorised and described it can be incorporated with the general cognitive models to form a comprehensive source of domain dependent *and* independent material. Consideration of both of these types of issue is essential to extract the full potential of user modelling.

### **9.3.1 Range of conditions for the occurrence of evolution**

The temporal structure outlined in the above generic task description will enable designers to identify the temporal locations of the given types of generic activity which would be associated with users of a given level of experience. The study of this generic activity will facilitate the identification of profound areas of change in the users activity or knowledge state where change in the support requirements of the users may be necessitated. Fluctuations in the overall local repertoire of generic concepts and actions manipulated by users can be identified and rationalised. These changes in the use of generic concepts and actions mean that typical gross changes in activity concomitant with a user groups performance of that given activity can be isolated. The hypothesised changes may occur at different points due to natural variation within user group hence a range of possible temporal values for the evolution of to the user to the new significant state is possible and a band or range of values can be established. These natural variations in user group behaviour would be supported by a statistical analysis of the normal distribution of generic interaction behaviours in the initial behavioural sample. The designer could then assess the statistical probability that a given type of user will evolve at a particular point and make predictions about their behaviour at that point.

### **9.3.2 Evolution range as the basis for self-adapting systems**

A particular application for the above type of model would be in the implementation of self-adapting

systems which are capable of evolving support facilities along with the natural evolution of the user. This would require the rapid capture of a significant sample of the users behaviour. This behaviour could be captured in the form of the commands or types of keystrokes or actions used or in terms of biases towards particular subsections of the system or as overall interaction behaviour via a medium such as video. This model of behaviour would facilitate rapid adaption to a new user of different experience. One way this distinction could be made would be in terms of command complexity: use of esoteric or specialist commands or alternatively use of long, complex and correctly formed command strings. The user could be prompted for a given sample via a simple task or query. If the user provided the right answer to the query the machine would adapt to that level. Given a wrong answer the machine would ask a simpler question until the user answered a question correctly and adapt to the relevant level.

### 9.3.3 Customised scales for specific types of user

Once a machine has gained a large sample of a number of users behaviour a set of default adaption levels could be set for the typical requirements of the relevant user group and level of experience. Obviously, domain considerations may play large role in determining specific temporal behaviour so these discrepancies must be accounted for in the type of scale produced, hence an overall model of group behaviour should be formed before specific users are addressed. Commonalities may be identified within each type of user so that existing material which has been recorded can be used in forming a new scale. Common temporal constraints may form such material, such as time-taken to retrieve a given piece of information from a given source in a given context. The method described above is the type of approach which is advocated. There exists no general timescale, although some general features of the types of event which might occur at given points in these timescales is given in the proposed 'Standard Task- Path Description Technique'.

Determining experience level within a particular user group could be facilitated with a model of the typical types of task activity, operation and concept which would be manipulated at various levels of

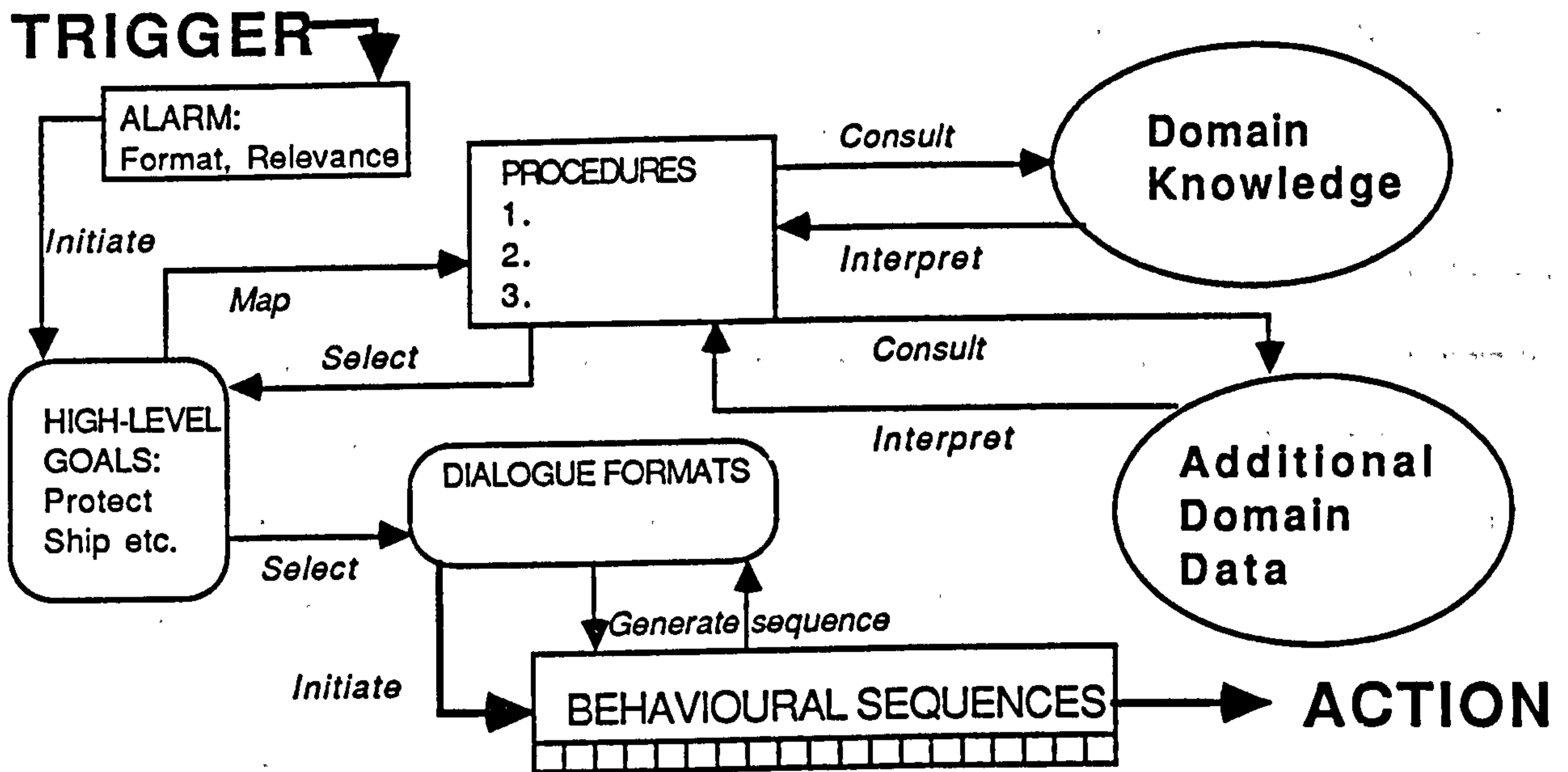
experience in that domain. This model would allow the incorporation of a specific independent task model which would be divorced from interaction knowledge but would offer an equally valid source of information on which to base adaptation strategies. This experience model would be again validated by a statistical analysis of the normal occurrence or distribution of typical task behaviours within hypothesised user groups. Experience descriptions could therefore be based upon realistic task behaviour repertoires from the given user groups.

#### 9.3.4 Conceptual Maps

One possible method for the representation of the users domain and interaction knowledge explicitly could be the conceptual maps mentioned in 6.3.3. These tools have been identified as being especially useful in the description of particular cognitive and behavioural attributes generated by the cognitive model (Hitch et al. 1986, Sutcliffe et al 1986) via the concepts and operations manipulated by the user. These include such real-world individual strategies as 'Cognitive Style' and hence, would be relevant in the specification of realistic generic behavioural repertoires.

Typical concepts and entities, and the operations by which they are manipulated, which occur within a user group can be identified and described using this technique and it has been shown to be of practical use in the design of interactive systems (Sutcliffe et al 1986). Maps could reflect typical cognitive biases and report on the users actual perception of entities in the cognitive workspace.

If the common mode of perception of given task related entities could be defined for particular groups or subgroups then the designer could make concrete predictions about the repercussions of using a particular representation of an object (for example, an icon) on the display at a given point. This could be behaviourally elicited by studying users in their application domain and represented as a set of user and temporally defined conceptual maps. For instance, in the ship example a Captain may perceive an alarm differently than will a radar operator. For the former it initiates the save ship procedure for the latter it is simply a message of a given type. The captain has responsibilities for saving the ship thus his conception of the problem is different. This is determined by the functions and generic activity of the two people.



**FIGURE 5.5 A SIMPLIFIED CONCEPTUAL MAP FOR THE CAPTAIN'S ROLE IN THE SHIPBOARD THREAT HANDLING EXAMPLE**

Each of the entities and actions will possess differing attributes according to the individual or user group performing the task:-

**ALARM Attributes:**

- Status - Green, Red, Yellow
- Format - Report, Hooter, Control Panel
- Relevance - Trivial, Significant, Important, Crucial

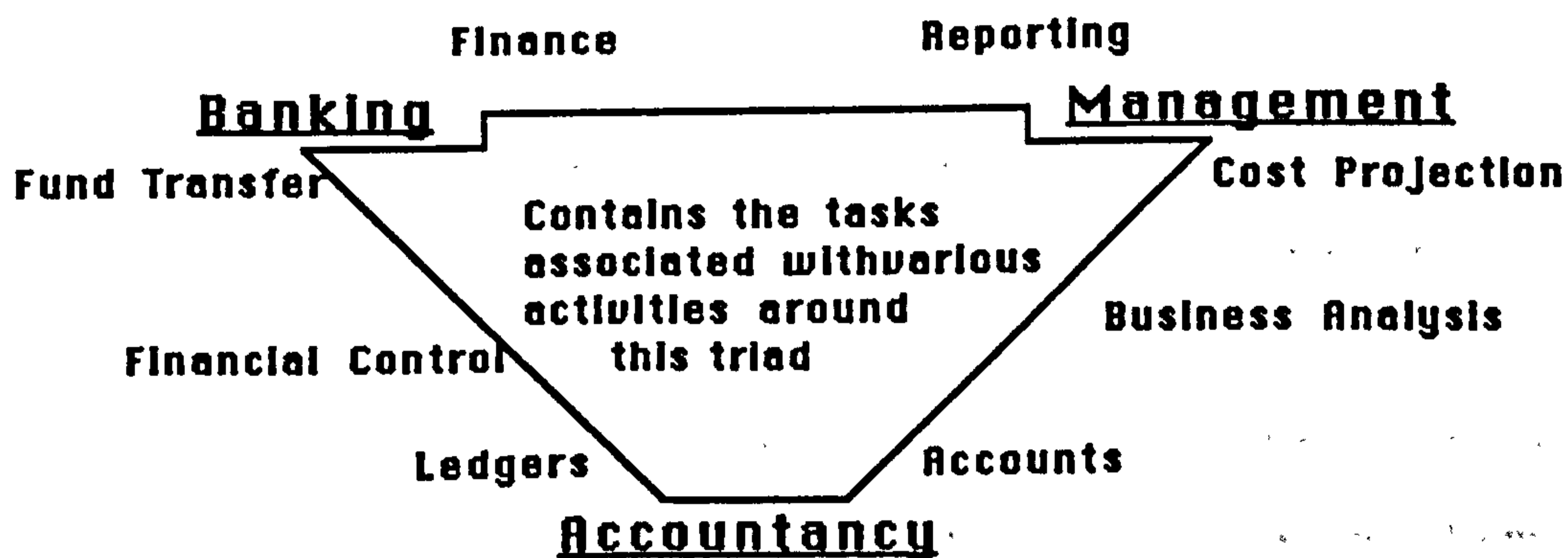
The above properties would be generic features of those entities for a given group.

### 9.3.5 Group knowledge overlap tool

Further use could be made of the generic group descriptions by identifying similar behaviours between groups and recombining the descriptions to form a more general model. Hence, an overall model can be formed in which similar generic behaviours will only be stored once. An example of this is that accountants, businessmen and bankers also share some common generic activity and this

could be exploited to form a tool which could specify a general interface for all three groups. This is already done, to some extent, as many interfaces are based around spreadsheets which are common tool for the above three user groups.

The proposed technique, however, enables the identification of the specific common generic activity within various user groups and allow the design of interfaces which could support these specific similarities. The problem with common interfaces based on spreadsheets, for example, is that they are not designed to support each user types specific task but are based on a common data model which requires compromise on favoured interaction strategies and imposes a higher learning overhead. This common generic activity would remain a model of the real-world activity in all the groups and any differences would be explicitly displayed.



**FIGURE 6 THE OVERLAP OF SIMILAR GENERIC ACTIVITY IN DIFFERENT USER GROUPS**

#### **9.4 HOW THE OUTPUT OF THE TECHNIQUE CAN BE VARIED**

The tools described in this chapter can be utilised by the designer for a number of different purposes and can play a role at various levels and stages in the design cycle. A number of the properties of the model can be exploited in various ways to assess various configurations. The descriptions produced by this proposed technique can be set at a number of level of analysis governed by the

requirements of the designer.

#### 9.4.1 The utility to the designer of an 'a priori' knowledge of temporal structure

One of the major contributions of this type of technique would be in allowing a designer to explore potential (absolute and relative) temporal structure in advance of the design of an interactive system. This would facilitate the evaluation of proposed structures and iterative refinement to identify the optimum configuration. This technique would be appropriate for the design of deterministic systems as it would allow the designer to identify the optimum behavioural configuration. This design would be based on natural and appropriate human scheduling and prioritisation strategies and would be empirically testable at the behavioural model stage. Candidate models could be formed of typical user configurations and they could be quantified and evaluated.

The 'a priori' model could also form a basis for the implementation of an adaptive system as the structure enables the designer to make reasonable predictions about what activity a given user will be undertaking at a specific point in the future. This extends Reisner's (1977) notion of 'prediction assumptions' to encompass specific user group information and intra-task predictions. Despite the fact that this model is not based on universal human cognitive and perceptual attributes it can display similar reliability to models based on those attributes by the careful application of empirically established and statistically tested behavioural models. This technique allows the designer to draw in a large amount of relevant detail on which to base predictions about future behaviour and as a result those predictions can be at a finer grain-of-analysis (down to subtask level) and can be located on a cardinal temporal calibration scale. Overall, the model can include specific and universal information for the purposes of specifying the location of generic activity, commenting on domain behaviour, outlining inherent task structures within humans, elucidating mapping rules between various levels, introduction of 'real-time' value descriptions and offering a simultaneous user and task event description.

### 9.4.2 Externalisation of temporal task descriptions

It has been noted that part of the value of HCI models lies in their ability to facilitate externalised, precise descriptions of interactive situations (Reisner 1981). A great deal of supporting work has been carried out highlighting the value of externalisation in a number of activities. Work on vocal externalisation of scripts (Miyata 1984) has shown that feedback via an audio recording of a given written work enables practitioners to conceptualise and comprehend the structure of the work. This technique is used in 'Knowledge elicitation' to enable the designer to obtain an overall view of a subjects knowledge (Hart 1985, Kidd 1984). Particular techniques for this include 'talkback' and 'think-aloud protocols' (Newell and Simon 1962).

Hence a tool which allowed conceptualisation would be of concrete use in the design cycle. A number of categories for the role of externalisation have been identified by the author. Firstly, the externalisation of a description followed by subjective evaluation by the designer. Secondly, externalisation and comparison against a known HCI structure using a given tool. Thirdly, structured externalisation with some reconciliation of realistic behavioural issues. Fourthly, externalisation and comparison against a usability measurable which purports to represent and correlate with all relevant HCI behaviour. Finally, externalisation to identify the syntactic and semantic structure of the HCI behaviour via a complex analysis, perhaps using a layer-model description. Some of these methods evaluate on the level of the subjective impression of the designer - expert opinion - some against objective structures or criteria whilst others allow a point-to-point mapping onto the syntactic and semantic requirements of the task.

The proposed tool aims to allow externalisation for the purposes of the final category of analysis. The multi-level and dimensional approach suggested should provide a comprehensive description which can be used to induct data from all relevant domain-independent (syntactic) and dependent (semantic) sources and facilitate evaluation via the analysis of the representation according to a formal match between user real world behaviour and the machines functional capabilities.



So, even a paper description of the proposed structure of the interface can provide the designer with a sophisticated tool for the conceptualisation of a situation and as a vehicle for the condensation and collation of a large amount of data and knowledge about a given HCI situation.

### **9.4.3 Evaluation of temporal values and position**

The focus of the tool for the designer can be on purely temporal issues or in terms of offering a temporal framework for the structuring and analysis of other factors. The rationale behind this approach has been the subject of much discussion in previous sections and this is perhaps a unique function of this tool. The designer may, however, wish to create a temporal description in isolation from other factors and treat other units in a purely temporally independent fashion. This tool would enable the designer to do this.

### **9.4.4 Identification of simultaneous activity**

The multi-level and multi-dimensional aspect of this tool means that activity can be resolved in terms of the simultaneous activity which it involves on various levels and dimensions and that simultaneous activity can be accounted for in the design of an interface. The human capacity for 'parallel' or concurrent processing of task data and the complex processing and scheduling strategies which are utilised to generate these abilities can be captured and represented using this tool. This enables these properties to be referred to and exploited in the development of a number of current and future applications. Specific human strategies are exploited to optimise use of processing space and support multiple levels of activity and it is such strategies which must be identified if they are to be exploited by interactive technology.

In British Air Traffic Control, for example, controllers are expected to control up to eighteen aircraft simultaneously. Two of these aircraft would be actively controlled at one point, with the remainder

being passively monitored. This parallel control process is possible due to sophisticated formal procedures which controllers follow, however the particular cognitive and behavioural strategies which enable the controllers to execute these activities are as vital as the procedures themselves (if not more so, or the activity could be automated). Modelling the specific generic behavioural patterns inherent within this activity would allow it to be structured and regulated by interface configuration to optimise the controllers abilities and reduce demands whilst still leaving the ultimate control with the human. The machine could typically handle the monitoring of potentially dangerous convergence of aircraft conflict whilst the user was engaged in other activity. Facilities for the support of time dependent behaviours could be included to exploit the controllers existing interpretation of the procedures. These facilities could be designed with reference to the type of temporal model hypothesised within this thesis.

Another property of this type of situations which make this tool particularly useful are the 'hard' time constraints implied by the process. Resolution of user strategies within such constraints is a particular ability of this tool.

#### **9.4.5 Location of decision and evolution points**

The existence of identifiable knowledge states and concrete evolution phases between these states have already been the subject of some discussion and the role of this tool in identifying those aspects determined. Another feature identifiable with this tool would be 'user decision points' wherein the user requires exhibits, specifiable knowledge or information requirements which must be satisfied at a given point if a decision is to be made. These requirements will be determined by the typical activity or type of task being undertaken and would be specified along with the generic behavioural model. The typical requirements associated with given real-world activities would be associated and particular active decision points identified. Again, a range of temporal values could be identified within which the user would require specific support facilities. This range could be statistically analysed to eliminate the fluctuation in demands due to the normal distribution. For

example, in the instance described in later chapters such 'evolution' is expressed in terms of a change in interaction device knowledge where a user requires a more powerful form of command to be able to perform an activity.

In an typical application, this change in user demands could be described in terms of a users information requirements to make a given decision. An example of this would be the market information to make a decision using a 'dealing room' system and the temporal constraints involved. These requirements should, however, be resolved in any form that the user may require to be able to achieve a given goal and this is a principle attribute of this technique. The medium used to describe such requirements within the model could take a variety of forms, and to this extent the medium utilised to maintain the user description is independent of the specification methodology and remains extremely flexible. It is perhaps this flexibility, due to lack of restriction to one medium (dialogue, for example) which is the unique and most powerful property of this method.

#### 9.4.6 Location of user requirements

The location of user requirements may not always be associated with the explicit performance of a given activity or execution of a decision. Users may require general variations in feedback requirements depending upon their position in the interaction path. Some of the scales described earlier reveal natural fluctuations in implicit user state which could be remedied actively by the interface.

Current interactive computer technology, due to finite memory and processing power, will only allow user access to a given subset of the machine's possible information and function.

Alternatively, all the possible support facilities can be offered to the user at any point during interaction (with corresponding system delays). The WIMPS interface, for example, only offers the user a given subset of functions at any one time dependent upon the ongoing task demands. Access to the data in a text-file is also limited at any time as the memory manager will ensure that

only a small segment of the data is loaded into the bit-mapped memory at only one time and although access times are short disk-accesses can become very irritating. The rationale behind this design is that the behavioural component and limitations of typical task activity mean that only a small portion need be accessible at any given time. This is based on a very simple temporal model. Access times for current data should be less than a given value as compared to novel data which should be less than a given higher value.

If a more complex model of the users typical behavioural repertoire were obtained then a more explicit strategy could be determined. Documents which are typically accessed from one another could be linked for easy access according to the users position in the interaction path and a significantly similar analogy of the user real-world behavioural access requirements formed. Priority and scheduling strategies in the handling of data could be outlined and integrated into the interface design. Hence, if it is known that access to an index file is usually made after editing a text file that link should then be automatically available.

## **9.5 THE ROLE OF CALIBRATION IN THE SYSTEM DEVELOPMENT CYCLE**

Temporal calibration can play a differing role in interactive systems design depending upon where it appears in the system development cycle. The variation within this role can be clarified by drawing the distinction between working and abstract representations. If the tool is to be used early in the design process it can be employed to produce descriptions and allow evaluation of the temporal aspects of proposed designs. This approach will allow the incorporation of temporal behavioural models into the design process and the iterative evaluation of any such model before any prototype is built.

A model could also be developed simultaneously along with the construction of a working prototype and hence, will act as a consistent representation of the temporal aspects of the system

which would be an accessible representation of the temporal issues involved with the design without actually influencing the actual issues embedded in that prototype.

Thirdly, the model can be used to apply a retrospective scale to a working machine to facilitate comparative evaluation against existing machines and the results produced could be extrapolated towards proposed designs if an appropriate set of controls were applied.

The first model is an *abstract* representation of a proposed system based on real-world user behaviour. The latter two are representations of *working* prototypes and are based on both real-world behaviour and interactive behaviour. The flexibility of application is one of the major virtues of this technique.

### 9.5.1 Abstract models

The specific value of the model at this stage is as an interim representation for purpose of the iterative evaluation of a number of proposed configurations. This model would provide temporal user models for the designer and would provide the basis or a number of sample experience levels of users to determine the necessary provision of functionality. Essentially, the function that this model would perform would be in the induction and structuring of an amount of generic temporal interaction behaviour which would allow the designer to make prediction assumptions about the users future behaviour. Specific features of designs could be addressed such that they can be refined and resolved during design by creating simulations of users performing sample tasks.

### 9.5.2 Simultaneous generation along with a prototype

The function of the model at this stage is as more of a representation than an evaluation tool. The design is already, to some extent, committed at this stage and cannot be influenced by evaluation using this tool. Storage of the details of the temporal aspects of the chosen configuration in a

consistent and accessible format is a desirable feature here. This data can be used, for instance, to create meaningful documentation which actually resembles the structure of the interface as it is used.

This type of tool could be used to ensure the consistency of particularly the temporal aspects of the design throughout the construction process and would preclude impossible or illegal temporal behavioural sequences or demands. This would also make the model a useful representation of the ongoing progress of the design as these issues are resolved. This would be useful for the reference of various members of the design and construction team.

The principle function of such a representation would be as a method of storing an assessable representation of the machine which could be stored and correlated with its eventual performance. Hence, the successful aspects of the design could be identified and rationalised and the generic interaction description used in the design of similar systems in the future. The re-usable nature of these descriptions, due to explicit control over machine and user factors, allows them to be useful in subsequent development and indeed this material would be essential in providing examples of realistic interaction behaviour to validate future abstract models of similar systems.

### **9.5.3 Retrospective application of a scale to working system**

The final mode of employment of this type of tool would be as a formal tool for the assessment of existing equipment. This would not only allow comparison between machines on the basis of one or several general metrics rather it would allow point-to-point analysis of similarities in user *behaviour* on a number of machines and as a result would facilitate a 'diagnostic' assessment. The formal analysis of systems using controlled psychological study will typically offer more valid and reliable results due to the ability to provide specific controls over spurious factors, however this tool however would represent a less costly and more rapid form of analysis supported by the controls inherent in the behavioural model.

Meaningful comparisons could be made between performance with various configurations by virtue of the standard behavioural model and, as in the last section, features of successful systems could be integrated with potential future designs to validate any proposed model. Retrospective analysis, however, is envisaged as the primary mode of employment of this tool as much greater value can be derived from the evaluation of proposed temporal configurations in the early stages of design. It does, however, possess significant benefit as a sound, consistent and objective evaluation model for testing existing applications.

Retrospective analysis can be applied to working prototypes to suggest design changes for the re-build phase of the cycle. This would typically be part of a battery of tests to comprehensively evaluate the success of the design.

## 9.6 MEANINGFUL CALIBRATION

It has been suggested that traditional task or clock based calibration scales are meaningless to users when describing HCI as they do not represent real-world interaction behaviour with any accuracy. An alternative approach is now suggested in terms of 'Generic Temporal Interaction Units'. These can be defined as the natural identifiable, discrete, temporal units which exist naturally in the users performance of tasks - in this case with computers - and which can be combined to form a temporal description of any task of that type. The robustness of the units is inherited from their contents, in terms of, for example, knowledge or dialogue, but involves solely a temporal description for their initial purposes. This involves the assumption that such units exist, that they can be identified and that they can be correlated with particular user types.

This type of unit should enable the designer to describe the particular fashion in which the user perceives the task as a real-world interaction descriptions and hence elucidates the particular potential behaviours which may be exhibited by a given user group or vice versa. It would seem that the onus in terms of design activity lies in the identification, extraction and realistic

representation of such behaviours. The reliability of such units depends on the quality and reliability of the techniques used to carry out these activities and hence, this reliability ultimately also relies on the accuracy and relevance of the chosen statistical controls. One of the features which suggests the existence of such units are the threshold temporal limits which users apply when performing a given sub-task. Accessing information from an on-line dictionary will be subject to an upper limit in terms of time dependent upon its role in the task. Resolution of the individual behaviours within this sub-task can be variable and this affects the total sub-task execution time, however the unit remains an integrated whole.

These units differ from Card, Moran and Newell's 'Unit - Tasks' as this tool aims to specify integrated and discrete units in terms of a number of media and to use this to the possible orders of behaviour within each unit, rather than merely attach a temporal value to a supposedly objective unit. It is necessary to identify the actual units employed by the user, along with the components and the interdependencies between those units and components, if a realistic model of human temporal behaviour is to be formed. An example provided in later chapters resolves these units as knowledge states and hypothesises that the hierarchical structure is temporally constrained. Hence, it phrases potential constraints of the following type: 'is it possible to know x. before knowing y in the context of Interaction Device Knowledge?'

Users perception of progress during interaction will be based on the achievement of given task 'goals' which will be definable in terms of these generic interaction behaviours. Performance of a given repertoire of these behaviours will evoke a sense of completeness for the user as all of the sub-goals are resolved for the achievement of the task goal. This can be described by the psychological concept of 'closure'. Information and knowledge requirements can be derived from the ongoing state of generic activity and features can be implemented to satisfy those requirements at given points in interaction. Observation of the utilisation of a combination of a set of the known generic task features for that user, which corresponds with the users goal being achieved, will define the precursors for the evocation of closure in that user.

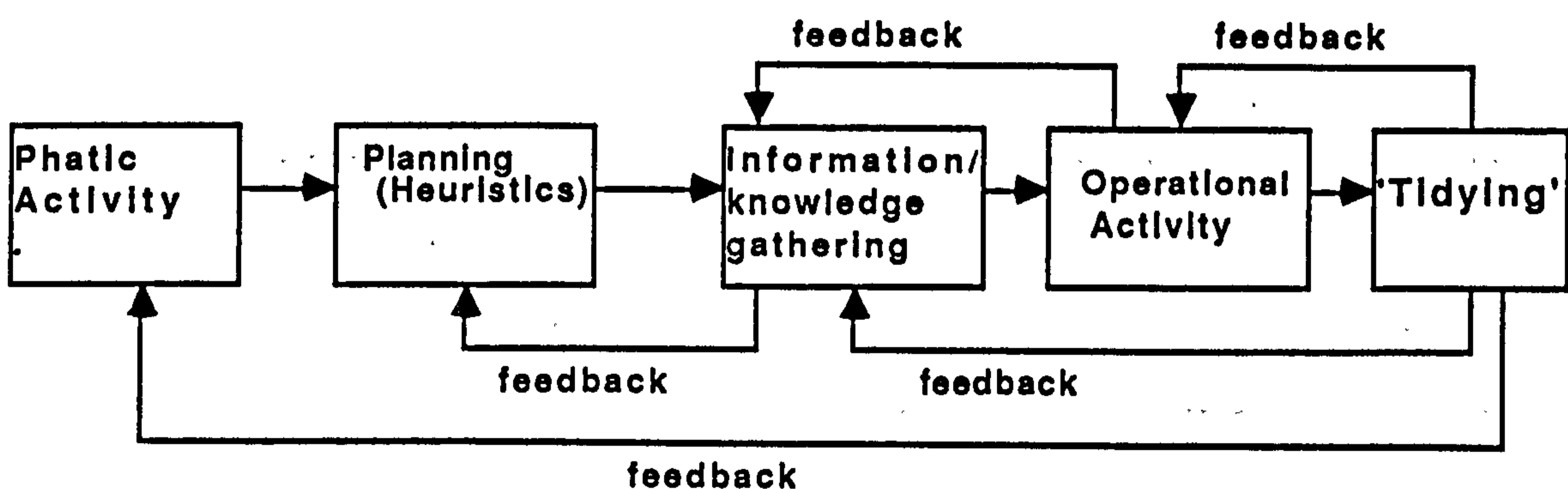


In conclusion, the features which can best describe the interesting aspects of temporal issues are those which can be derived from a human perception of the task structure. These features can in turn be described by modelling generic interaction behaviour on a number of levels and rationalising the appropriate units.

### 9.7 A STANDARD TASK-PATH DESCRIPTION TECHNIQUE

A useful tool for the HCI designer would be one which enabled them to describe the approximate temporal constraints which exist on the possible display of types of interaction activity. These can be in the form of 'hard constraints' wherein 'LOGIN' must be displayed at the start of interaction. Alternatively, they can be in the form of 'soft constraints' in that certain types of activity are more likely to appear at given points on the interaction path, and furthermore that certain types of activity are likely to follow or proceed other given types of behaviour, such that the quality of the results obtained when reaching the desired goal is governed by a given ordering.

A general model of the given phases of interaction has been formed, as a result of experimental work carried out during this work, within which known constraints can be said to apply:-



**FIGURE 7 THE STANDARD TASK-PATH DESCRIPTION TECHNIQUE**

The above figure represents example general categories of activity which can be constrained with respect to position on the interaction path and provide a potential framework for the analysis of

interactive computing behaviour. This model is task-independent as it does not refer to the task content of each unit, merely its syntactic structure. Barnard's (1988) notion of an 'approximate user model' supports this type of model. The integrity and self-containedness of the generic units of behaviour described later in the thesis in combination with the approximate, general temporal framework gives realistic model which allows comparison between task-paths allows manipulation of generic behaviours within the relevant temporal stages.

The assumption that task constraints exist on some level is an essential precursor to the coherent study of HCI, as behaviour would otherwise be modeless or indivisible and a coherent structured interaction environment could not be provided. The temporal stages may not be explicit in the context of some applications, however they can be deduced from variations in the type of operation which occur and the above model presents a general framework for this process. Modifications of this framework would make the structure appropriate for any application.

The action phases within this model are resolved as a series of cycles in which feedback is obtained from later activity and a gradual refinement of the structure is effected. Introduction of sample behavioural units into these cycles would require units which resemble this process. That is, these units should be progressive, regressive and/or recursive transitions leading to incremental models of the ongoing task-state of interaction.

This is an approximate model which could be broken down into specific types dependent upon the actual purpose of the interaction. This process would enable incorporation of semantic material from given types of task. If a series of meaningful frameworks could be formed to cover a wider range of interaction types they could comprise the basis of a standard framework for the recording of behavioural data about interaction which would allow comparisons on equivalent terms with recourse to the standard task description. There is a significant need for such a tool in the field of HCI to co-ordinate the data extracted during many different studies for its incorporation into a general model. With a standard framework only novel or salient aspects of task descriptions need

be stored uniquely with existing data forming a reliable kernel of behavioural primitives. Rapid comparative evaluation could be carried out between systems in a specific 'point-to-point' fashion. Standardisation can be maintained by referring to the objective, functional task description.

## 9.8 BEHAVIOURAL DATA EXTRACTION TECHNIQUES

The foundation of valid work in the study of HCI is in essence, the quality and realism of the behavioural analysis techniques utilised in its primary stage. Reliable complex models of behaviour can only be established if the underlying behavioural representation is accurate and the premises on which the behaviour is reproduced are sound. Representations need not necessarily be precise or at a very low-level of analysis as long as they are consistent, reliable and realistic. The available range of tools for executing such studies are currently limited and do not always generate the type of material which would be of most use in HCI design.

The essential difficulty which exists in devising methods for employment in this area is that the behaviour or phenomena which must be modelled are rarely explicit or observable and, conversely, the explicit behaviour exhibited invariably does not relate directly to the phenomena which must be modelled. Hence, the focus of current usability assessment techniques and their underlying data capture facilities is towards observable behaviour such as keystrokes or utterances from which simple action descriptions are formed. Interpretation against a hypothesised task structure offers another qualifying perspective, however this task may not be based on real-world interaction behaviour and may or may not relate to the actual strategies used.

Use of 'Protocol Analysis' allows the experimenter to question the user on their goals, motives and methods. This can be done during interaction in which case the interaction is influenced by the interrogation and the method may vary. Alternatively, the analysis can be retrospective, via perhaps a medium such as video, in which case the subject may not be able to provide an accurate record of the behaviour used as the subject may have forgotten exactly what was in his mind. This type of

technique may not be as valid as was previously assumed due to work which has shown the relative unreliability of users reports about their own behaviour, goals and motives (Rich 1983).

An alternative to the above technique involves attempting to elucidate some of the discourse and interactive problem-solving protocols of human behaviour by asking two individuals to solve a given problem interactively (with a given set of instructions). This can be face to face or via limited communication channels, such as the telephone (Evans 1974); the latter more accurately simulates a computer. This technique is known as 'constructive interaction'(O'Malley 1984, Miyata 1979), however it is often difficult to maintain the relevance of such simulations in a HCI context.

Hence, the activity involved in 'behavioural data extraction' can be divided into two distinct phases: data capture and data structuring. Data capture methods in HCI include videotape, system transcripts, questionnaire, written transcripts, audio transcripts and local interrogation.

The author has identified some potential techniques for behavioural elicitation. These involve a more structured approach based on actual action representation rather than user reports. Essentially, any structured model should derive from known existing structures about users actions, however the dilemma this presents is how to obtain such a model before elicitation begins. Formation of a fairly robust and reliable model of behaviour based purely on observation then the qualification of this model with retrospective user feedback should be possible as although humans may not be able to produce spontaneous reports of protocols they may be able to distinguish between the right and wrong ones (used and not used). This model can be continuously refined until it predicts behaviour reliably (with actual machines) and then extrapolated toward similar situations to form a more comprehensive model. This could be combined with unobtrusive data capture techniques (covert video-recording) such that the user is unaware of the study until the feedback phase. Interrogation of users for the purposes of gaining user feedback would be via a 'structured interview' technique to control for experimenter bias due to the presentation of loaded queries.

This type of study could be supported by 'constructive interaction' type studies which actually require users to interact with a simulation perhaps in terms of a model of the 'Virtual Objects' of the system. Hence, no influence is made by another party. Feedback could be simulated by the experimenter reporting the results of their actions on the virtual objects according to pre-defined rules (those of the proposed system) and this may reveal the user would learn the system. This process would then not be a reported simulation but an action simulation. The study of users interaction with simple and paper protoypes is a very useful source of intial data on task behaviour which has hitherto been utilised sparsely. The above techniques are explored in the experimental study offered later in the thesis.

Other proposed techniques include software for the automated collection of data via questionnaire or by offering selected tasks to the user to see how they are performed. Specific questions on the subjects perception of system objects could be put for the purposes of forming a perceptual model. Matching against known problem solving protocols could be attempted to elicit a sample strategy. Explicit record should be made of error behaviour even in the most primitive of models in a standard format related to the task.

The final area of potential would seem to be in recording implicit behavioural cues; frustration responses or hesitation would indicate areas of difficulty, as would anxiety values, whilst studying user gaze would identify areas of attention at given points in interaction.

This chapter has described the types of temporal calibration which are possible for HCI. The concept of calibration scales for particular groups of users is discussed and this is explored later. The contribution of calibration scales to design has been examined. Finally, ethological techniques for data capture are discussed and these are employed in the empirical work.

## **10. INTERACTION STATES AND MULTI-MEDIA DESCRIPTIONS**

This chapter introduces the notion of discrete interaction states and describes the process of modelling users in terms of such states. Possible constraints and variations on interaction states are described in terms of the evolution of the user during interaction and salient changes in the interaction model are identified as the points where changes between states occur. A mechanism for the evaluation of interaction is described by comparing the task descriptions generated by the model with the users task requirements.

### **10.1 IDENTIFIABLE DISCRETE INTERACTION STATES**

The previous chapter discussed the notion of discrete temporal user states, which were to be identified in terms of generic interaction units. This chapter goes on to elaborate on the nature of such units in terms of a particular potential format, namely discrete knowledge units, and suggests how these might be used in the definition of discrete user states. Discrete user states are those segments of user behaviour which are relatively self-contained and integrated and which display an identifiable and limited repertoire of generic behavioural units related to the users cognitive model at a given point in the interaction-path. Such discrete states will be identifiable by a statistically similar exhibition of generic behaviours.

It has been suggested in previous chapters that one of the essential precursors for the coherent study of HCI is the assumed existence of discrete and identifiable user, machine or interactive states which can be categorised, specified and described. If it is assumed that the process of HCI is merely a stream of indivisible actions, reactions, percepts, assumptions, contextual cues, data, information and knowledge then a structured analysis of the interface would prove to be extremely problematical. If HCI is assumed to be a homogeneous stream of information flow which cannot

be elementalised then it cannot be described and evaluated meaningfully and conclusions cannot be drawn from the structure. It is true that the potential variety of types and modes of information, cognitive state and type of action are vary great in number, hence when they are combined the number of potential states is also very numerous due to the possible combinatorial permutations.

Many theorists, however, have identified a number of possible finite or discrete states for HCI and these are discussed in the following sections. Such states can be identified with some reliability and hence they facilitate a useful 'approximate' (Barnard 1988) description of the interface at an appropriate grain-of-analysis to be of use in HCI design. This thesis asserts moreover, that much of the discernible structure of the human interface can be derived from the nature of the user and the task. Essentially, the interface is the task, or rather the interface is the means for the user to perform the task.

HCI representations can be stored in the form of a variety of media. The choice of possible media for the implementation of the model described in the previous chapter is numerous. An exemplar medium is required for this study which demonstrates the potential applicability of the model and which outlines the potential expressive power of many of the possible media. The medium required for the specification of HCI must be one which enables the accurate and realistic description of interaction units and states. The ideal position would be to produce a model which in conjunction with task descriptions could evaluate instances of real-world interaction behaviour.

Insightful use of behavioural elicitation techniques such as those described at the end of the last chapter enable the experimenter to capture data in a variety of media so the choice of notations is numerous. The integration of material resolved in a diverse range of media facilitates evaluation from a number of perspectives and this is essential in capturing the dynamic aspects of any interactive situation. The underlying phenomena which drive behaviour are difficult to address by the utilisation of only one form of description. The use of alternative forms of description which can be generated makes underlying events explicit: for example, a given HIP strategy, invoked in a user due to knowledge of a given domain or a particular sub-task, could be initiated by a dialogue

event in an interactive situation. These connections can only be identified and specified using a multi-media model. The multi-media model provides an overview of the interactive environment which enables simultaneous and parallel events to be identified.

A medium which would seem to be useful in terms of the exploration of the potential of the various media, is that of a knowledge description. This medium has been identified as a 'powerful and task-independent medium for realising descriptions which can account for behaviour at a number of levels' (Johnson 1986). It is important in the modelling of HCI that the model structure reflects the inherent structure of the user and does not impose an arbitrary framework onto the interaction. Knowledge models can possess this property and can generate many instances of task-behaviour from one model. The interaction structure is determined during the implementation of the knowledge model and hence, this composes a flexible tool for the representation of user state. The particular subset or dimension of knowledge which is to be explored is that of Interaction Device Knowledge as this relates closely to the functionality accessible to the user.

Data extraction techniques suitable for Interaction Device Knowledge elicitation are particularly reliable, as they are based on explicit behaviour, so this approach would seem to be appropriate: it can be assumed that the user 'knows' about a particular interaction behaviour if it is displayed consistently in its correct format and hence the users knowledge state can be derived, to a certain level, from his behaviour (Johnson 1986). A deterministic model of the user can be formed and behavioural instances predicted and realised from the knowledge model. The users interaction knowledge is, therefore, identifiable and descriptions produced from combinations of the generic descriptors by Johnson's Task Analysis techniques can be said to be "well-formed, unique and discrete instances of behaviour" (Young 1984).

### 10.1.1 Task sub-goals or states

A number of forms of task-sub goal or discrete interaction state have been identified by various theorists as potential features of design tools which would enable the rationalisation and description



of interaction. These could be described as sub-goal or state descriptors which have been shown to be relevant in given types of application to store representations of the said applications definitive or salient interim goal states.

### 10.1.2 'Finite' states

This terminology is derived from the work of Parnas (1975) and this is the seminal work in this field. This work aimed to generate a method whereby activity could be expressed by a 'finite -state grammar'. Situations could be expressed in terms of a series of legal, finite sub-task states, transitions between which were specified by given operations. Hence, systems could be formally described in terms of the set of states, and the operators which were required to move between them, to reach a given goal. This was the original conception of the 'discrete, identifiable' state which could be identified in situations and exploited to produce a simple but functional description of interaction. This paradigm suggested the identification of a set of interface states for any given application which could be specified in terms of the presentation aspects and their relation to the underlying logic. A command language could then be constructed from the legal input strings which drove the transitions between states. In this way any application could be developed and a simple interface offered to the user which could offer the user choice between the states available in a given traversal sequence.

### 10.1.3 Dialogue States

A typical way in which discrete states can be realised in the design of computer systems is in the form of a dialogue and this has arisen as a concrete approach to interactive systems design (Gaines and Shaw 1984). Here, an interactive sequence is specified as a number of interactive states which are connected by dialogue statements. The states formed are, of course, an approximation to possible examples which exist in non-automated human problem-solving (Newell and Simon 1962), however they represent the generic set of statements that are necessary to be able to perform a task with the aid of a computer. Hence, a working representation of the task is obtained in terms

of the dialogue it involves and this acts as a framework or template for the design of the eventual system: dialogue statements may be interpreted as commands which allow the user to bring the machine from one state to another.

The structure of human communication is such that this metaphor maps naturally onto the dialogue process. Work on Multi-Party Grammars (Shneiderman 1982) enables the designer to predefine the dialogue structure in such a sense that it is automatable for a number of parties.

#### 10.1.4 Knowledge states

Whilst the dialogue metaphor is a natural extension of the finite-state description and it represents a convenient and useful design paradigm through its economy of description it offers limited resolution in the representation of user state and real-world interaction. A dialogue description may be sufficiently application independent such that it can generate a number of task-descriptions, however even generic dialogue models are limited in their descriptive power for the specification of interactive situations and they will offer a limited view of the underlying structure of interaction. The modelling of dialogue states is an essential process in system specification, however on its own it may be insufficient as a complete model of HCI. An explicit treatment and examination of task-attributes is desirable if the states which determine dialogue are to be addressed and accounted for in design.

A possible alternative and complementary state description medium is that of knowledge and this form of description has been utilised successfully to generate practical models of HCI (Johnson et al:1984). As has been suggested in previous sections, the knowledge model is a powerful representation which enables a number of possible tasks and implementations to be modelled.

It would appear initially that were any knowledge model to be imbued with finite-states its applicability and independence, and hence its power, would be reduced. There may, however, be temporal constraints on the application of items within the knowledge model at given points during

interaction. Users will learn during interaction so a different model would be required at the start of interaction to that at the end: this model would contain extra items. Clearly, the user cannot be modelled at every point where new facts are inferred so a set of discrete, finite knowledge states must be identified to represent the important changes in the user. These knowledge states should express the significant changes in state which affect behaviour in the performance of the task and would offer a general set of conditions within which behaviour could be predicted.

Finite knowledge states could be expressed not only in terms of the users knowledge of interaction but also in terms of the users possession of necessary knowledge from the application domain required to perform the task. This could be defined as the users level of experience within the user group and the formal specification of this knowledge would allow the designer to make assumptions about domain knowledge when defining the interface. This is a very important source of information for the application designer.

The principle task in both types of knowledge domain, however is the identification of reliable and meaningful knowledge states that the user can be said to possess for some significant period. To facilitate the identification of significant states for the purposes of design some reference may be needed to a functional description of the interface. It can be assumed that users exist who have no specific knowledge about either interaction or their given domain. This could be termed the 'naive' knowledge state. Then it could be assumed that there exist a group of users who have fundamental knowledge of a domain or the interactive system but cannot perform specific tasks associated with an application. This could be termed the 'novice' state. Next, users who have a grasp of the basic functions which enable a typical general or clerical subset of the task domain repertoire to be performed but have no specific expertise in any one area: this level of expertise could be termed 'working' knowledge. Finally, users who have knowledge of a large set of functions within one particular subsystem and where those functions cannot be employed in other modes. This could be described as the 'experienced' state.

The above represent a number of example general states and it is believed that other more domain

specific states could be identified within given domains or applications. This, however, remains a useful set of states and it is these states which will be explored later in the thesis. The states described above are generic states which apply to the majority of application domains. Other generic states may exist, in addition to, a variety of non-generic or 'specific' states which apply within given applications. Identification of such states would form part of the task-analysis process in much the same way that dialogue states are identified. The modelling of behaviour using generic descriptors would facilitate the identification of such domain dependent and independent states as common behavioural elements which could be identified and statistically analysed.

The designer needs to know about the range of behaviour within the user population. The finite states described above offer a convenient basis for categorisation of the skills inherent in users of various levels of experience so that the interface can be designed with a clear user model.

## 10.2 'EVOLUTION' OF THE USER

Progression of users between states can be described as a process of 'evolution'. Evolution in this sense refers to changes in cognitive and behavioural state due to learning, forgetting and interactive experience however the knowledge model is assumed to reflect those progressions and hence, can be used a medium for representing such changes.

Evolution can be categorised in two forms: accretion of new knowledge to add to the total body of existing material, and removal of redundant or superseded material, or alternatively restructuring of existing knowledge (Bruner 1974) to accommodate or assimilate new material or to redefine the context of the existing model. Both of these processes involve the user deciding which material is relevant, and hence should be stored, and which material can be discarded. This must, to some extent, be determined by behavioural or task considerations with the priority being given to behaviourally adaptive knowledge with all knowledge being processed in terms of its potential use in determining behaviour. Hence, it is reasonable to suggest that the knowledge acquired about interactive devices in particular is task or behaviour related.

It has been suggested in previous sections that some usability problems will arise because new knowledge which the user is expected to acquire is incompatible with existing material. So, if an interface adopts an inconsistent or diverse strategy in terms of what has occurred in the users previous experience he will have difficulty in assimilating new material and usability will be limited. The users ability to acquire, extract and infer new knowledge about applications and interactive systems is a desirable feature of HCI. The user is the open part of the system and often any variation in the functionality of the system will be due to user interpretation of system facilities (Edmonds 1986). Hence, the knowledge which the user brings to the machine when interacting is a vital area of study if the state of the interface is to be assessed.

The interplay between the users existing knowledge and the assumptions made in the design of the machine define the users available repertoire and characterise the dynamic profile of the system. The user is engaged in dynamic hypothesis formation based upon their particular experience and their perception of the system (Schank 1982). Hence, the interface should ideally allow the user to learn in the appropriate way for the optimal exploitation of these facilities and the user should be supported in any exploratory activity which they might wish to undertake. This support can be offered by offering users such facilities as reversibility of actions (Shneiderman 1986).

### 10.2.1 Knowledge acquisition and learning

So, the process of user knowledge acquisition during interaction may be determined, to some extent, by behavioural factors such as the extension of the repertoire of knowledge for the control of the interactive process for the purpose of satisfying their goals.

Acquisition of knowledge can be viewed as an active process in which the user seeks a particular solution for the purposes of solving a given problem. This process involves identifying the specific demands of a given problem, devising a method by which a solution can be sought, accessing that material and assessing whether it is the correct solution (Newell and Simon 1972). This enables the

user to identify a specific type of solution to given types of problem which they may encounter in their task domain and hence is, to some extent, a unique set of methods for that users requirements or experience. Facilities could be offered for this type of exploratory behaviour in interface design and support could be offered to a users particular application domain to make this safe and easy. Alternatively, learning can be seen as a passive process in which the user has no explicit goals in terms of the knowledge he wishes to acquire.

Knowledge about the interface can be captured merely by interacting with the machine to determine its functionality, that is by exploring given non-specific task related behaviours which would seem to offer some utilitarian value to the user in hypothesised future interaction. This type of learning can be supported in interactive systems design by the provision of consistent interfaces which enable the user to make rational assumptions about interface behaviour. Users form models of the aesthetically correct patterns and forms found in a given systems (Malone 1984) and can distinguish these by the application of implicit knowledge. The same is true of system behaviour in terms of appropriate system response times, for example. Techniques may be needed to represent the processes by which this implicit knowledge is stored and the types of pattern isolated by the user so that models of the appropriate behaviour and appearance of systems can be made to facilitate optimal learning strategies. This can be effected in the form of a generic temporal model which reflects users preferences.

Another interesting distinction which can be made in the learning process is that between declarative and procedural knowledge. A distinction which can be made between the two is that the latter form refers to behaviour or methods by which given goals are achieved whilst the former is a descriptive representation. In practice the latter type of knowledge is used to implement strategies based on the formers knowledge of facts. Hence, the two forms of model are connected by a set of procedures for mapping declarative descriptions onto procedural solutions (Johnson 1984). It can be useful to produce separate descriptions of the two types of knowledge in the design of interfaces in order that the procedural repertoire provided by an interface for a given user can be evaluated against the set of declarative descriptions which the user would want to implement to perform a given task. This is

the method adopted in the evaluation described later in the thesis. Discrete states are identified in terms of both of these types of knowledge and the resulting set of generic descriptors evaluated against each other. In actuality, the two forms of knowledge are often defined in terms of each other so the distinction is difficult to make but would be useful for the purposes of design activity.

The final distinction which would be useful is that between knowledge and information. HCI is typically a process in which knowledge is used to manipulate some form of information to reach a known goal. This information can exist in many forms such as text, images or integers and although it can be modelled in relation to knowledge states plays a different but equally vital role in the interaction process and should be modelled separately. It may also be possible to identify discrete information states, in terms of packages of information which can be related to discrete states in the knowledge model or behavioural repertoire.

Interfaces should therefore offer users facilities to retrieve specific items of information readily during interaction as well as providing a general and appropriate passive learning environment.

### 10.2.2 Regression of the user

One of the salient features of the learning process is the individuals ability to reject or discard material which is irrelevant or which has been superseded. This enables the user to maintain a useful working knowledge kernel which does not incorporate redundant material and facilitates practical decision making (Schank 1982). Hence, some of the knowledge which would appear in a naive user knowledge model would not appear in later models or would have been transferred into another more appropriate form. Hence, we cannot assume that users possess the same knowledge model of interaction at its end as they would at the start and similarly we cannot assume that the user will know the same things about interaction at the start of a new session that they did at the end of a previous one. Hence, knowledge can be seen as a set of constraints on interaction which are relaxed as the users repertoire becomes larger or more appropriate. Therefore, the type and amount of knowledge which a user possesses can define the total functionality that the user can access.

This abandoning of user knowledge can be termed 'regression' as the user is actually discarding rules which have hitherto determined behaviour. This type of activity must be accounted for when designing interactive systems if the optimum learning environment is to be provided and this is especially pertinent to adaptive systems in plotting the state of the user. Formal models are required to identify the types of item which are abandoned, restructured or forgotten. This can be effected in the perspective of the particular types of item rejected by given user groups (in a task-related context) (Schank 1982). Items could be classified in terms of morphology, semantics or task relevance and related to user group demands. Items typically rejected from the generic knowledge model could be utilised to evolve that model and could be seen as only possible members of the model. This type of model is investigated in later chapters.

### 10.3 MODELLING KNOWLEDGE STATES

An initial step in the modelling of knowledge should involve the identification of the discrete, salient and generic entities and operations which combine to describe a users total task repertoire (Johnson 1984). These can be expressed in the form of declarative entities and procedural operations. The capture of knowledge states concerns this study, and it would therefore be appropriate to record instances of users actually interacting with computers to try to identify real-world states. Johnson's techniques offer a powerful tool for this exercise.

The method for describing an initial (naive) user state is described later in this chapter and once this model is formed then subsequent models can be described in terms of discrepancies from this one. Only the salient changes in the knowledge state need be recorded in terms of addition, restructuring or regression. This description may approximate to the evolution process of the actual user as users are believed to build upon a working kernel of knowledge as their experience is increased. This kernel is unaffected by subsequent movement and is very resistant to change.



Eason's results (1977) show that users tend to adhere to a consistent and incomplete set of commands even when performing new tasks which suggests that the working knowledge state persists for an almost indefinite period or until the user can no longer progress with the existing subset. His work shows considerable robustness to exist in these knowledge states even when they are insufficient and supports the notion that the user will, in many cases, preserve an idiosyncratic working command set which the designer should exploit to its full potential rather than providing excess functionality. This suggests the need for substantial experimental work into the nature of users biases, such as the nature of these command sets, along with the human bases for acquiring and exploiting knowledge about how to interact with computers.

As the technique, which is described later, shows, given states can be modelled initially by establishing users experience levels and correlating this with the type of knowledge that they manipulate. Their particular subset of the total generic knowledge kernel for a set of tasks can be specified and the discrepancies between users of different experience can be identified to account for the evolution process: this can describe the specific task-related differences between users of different levels of experience for a given interface. The validity of the users reported experience levels may be variable and this suggests that experience should be quantified from a number of perspectives (for example, in terms of a fixed scale such as 'time' and a user determined scale such as 'effort').

The distinction between declarative and procedural knowledge enables descriptions of users knowledge of interaction devices to be modular and hence lends a clarity of description to the process (Johnson 1986). A set of procedural knowledge units can be derived along with a set of generic declarative interactive knowledge units. The former may include such descriptions as 'point' and 'select' which can be realised behaviourally as 'position' and 'click' and can be further built upon to produce complex descriptions such as 'drag' and 'double click'. Hence, a kernel of operations available at any one level can be described.

The technique described is based on Johnson's work concerning task-related knowledge structures and deals with the task of interacting with computers in a wider task context. Johnson (1986) offers a technique for the identification and capture of a set of task elements which can subsequently be reduced to a generic form. This form reveals the fundamental set of objects and operators utilised by a user population. Johnson advocates the use of a range of data capture techniques to gather data for this activity and these include direct observation, interviewing, questionnaire, concurrent protocol analysis and retrospective protocol analysis. Johnson also offers a number of candidate techniques for the structured analysis of the data and these include, Repertory Grids, Card Sorting, Rating Scales and Frequency Counts. These can be added to the techniques outlined in the previous chapter.

Data is analysed for the existence of identifiable objects and actions within the task. Data capture techniques such as structured interviews can be targetted towards identifying genuine actions and objects. The data analysis techniques listed above assist in the identification of genuine objects and actions: hence, if frequency counts are used it can be assumed that the more often an object is referred to in the description of a task the more likely it is to be a generic object. Importance of objects and actions can be assessed via a number of criteria which Johnson refers to as Centrality, Representativeness and Genericity. Centrality refers to how important or central the objects is to the performance of a task. Representativeness, however, refers to how representative a given action or object is of the performance of a task. Representativeness and Centrality can be assessed using a number of techniques: the former can be assessed using rating scales, frequency counts or card sorting whereas the latter can be measured using rating scales, card sorting and memory tests.

Generification of the objects and actions can be carried out in a number of stages. Firstly, all the actions and objects are listed in two separate lists. Secondly, the lists must be reduced to a form where each element appears once. Thirdly, threshold of occurrence across subject and task groups must be set according to the situation to determine Genericity. Alternatively, the items can be grouped according to their similarity, as assessed by a number of independent judges. Correlations between these assessments will determine which are generic items. Generic labels can then be

agreed for each of the actions and objects. Finally, the generic elements are validated by separating each object from its generic category and asking subjects to identify which group the object belongs to. If this task proves impossible then a new group must be named by the task performer.

Johnson goes on to feed these descriptors into generic procedures, plans and goal structures also derived from the study of users performing tasks. Descriptions are used in a different way in the study to be described. Kernels of knowledge descriptors are used to characterise group behaviour and whilst Johnson (1986) refers to the validity of this activity he offers no procedure for performing this analysis. This study aims to equip the interaction model with a elementary temporal dimension by describing the sequences involved in the performance of tasks using Johnson's generic actions and objects. The study described here also advocates the use of such task-structures for describing states in terms of other media (such as dialogue).

In the execution of a task the declarative description will outline a set of conditions to be satisfied by the procedural strategies as in this example in which the volume control is altered on the Apple Macintosh:

DEC: POSITION (APPLE MENU 4(CONTROL PANEL(VOLUME CONTROL))).

PROC: POINT (APPLE MENU).

PROC: POINT (CONTROL PANEL).

PROC: SELECT ( CONTROL PANEL).

PROC: POINT(VOLUME CONTROL).

PROC: POINT (DRAG( VOLUME CONTROL)).

PROC:POINT(CLOSE BOX).

PROC:SELECT(CLOSE BOX).

Each of these is an generic descriptor which can be used to describe knowledge about users performance of a variety of other tasks even with other machines. Complex actions also can be described from simple entities:

PROC: SELECT (CLICK('DOUBLE-CLICK')).

So, if a complete, relevant and accurate generic knowledge kernel can be described using the above type of descriptors for each user or user group, and a set of knowledge *requirements* can be isolated for each interaction type, then a specification of a proposed or existing machine can be evaluated against them by assessing whether the users knowledge kernel could perform a given set of tasks. Hence, real descriptions of user group interaction can be captured and built upon to form a comprehensive model. This forms the basis of the informal evaluation carried out in chapter 14.

Declarative descriptors can include knowledge about entities such as icons, documents or menus, and the location of given functions. These can be resolved as 'file menu contains print function' or 'backspace key deletes a letter backwards'. They can also be used to create complex descriptions such as: 'spelling checker is at position 5 on search menu' or 'option plus '3' key echoes '#' character'.

### 10.3.1 Evolution of the model via salient changes

The overall knowledge kernel can be partitioned to identify given users group. A metric such as 'experience with computer x' (in hours) is used to subdivide the users and the knowledge elements which apply to that group are partitioned off. This forms the naive kernel at a level set by the experimenter. Once an initial total model has been formed and the naive user state has been identified then evolution can be modelled. The naive state may contain substantial device knowledge or nothing at all so this process is variable. Feedback from users would enable the designer to form preliminary models at a number of levels to represent given levels of experience. Changes from the initial model could be of the three types mentioned. These are: addition of knowledge, rejection of knowledge or restructuring of knowledge. The latter activity could be described in terms of the former two, as restructured knowledge could be seen as 'new' knowledge, however it would be useful to make this distinction. All of the changes could be

specified as manipulations of the total knowledge kernel.

So, the evolution process could be expressed as:-

User Model<sub>1</sub> + Additional device knowledge - loss due to regression +/- change due to restructuring => User Model<sub>2</sub>. (1)

This represents the abstract change in the users knowledge form the initial state, however it does not refer to the temporal component which defines where the change occurs. This component is essential for designer to locate evolution. This could be represented as :-

User model<sub>1</sub> + additional device knowledge etc..... => User model<sub>2</sub>.  
 (time t<sub>1</sub>). (time t<sub>2</sub>). (2)

This affords some perspective as to the temporal framework and may be qualified using further temporal subdivisions to refer to specific aspects of evolution in a more precise timeframe (time t<sub>1a</sub> for example.) The use of single time values to represent evolution is, however, inappropriate as evolution may occur within a range of temporal values within a given user group. A better form of the equation would be:-

User model<sub>1</sub> + additional device knowledge etc.....=> User model<sub>2</sub>.  
 (time t<sub>1-t2</sub>). (time t<sub>2-t3</sub>). (3)

If a particular subsection of a user group is to be examined to analyse their specific evolution behaviours perhaps in the design of a specific system feature the model can be qualified to refer to a given subgroup. Hence:-

subgroup a.

subgroup b.

User model  $_1$  + additional device knowledge etc..... => User model  $_2$ .

(time  $t_1-t_2$ ).

(time  $t_2-t_3$ ). (4)

The final type of information which can be included in the formation of the user models for use in generating task descriptions is a structured model of the error behaviour typical to one subgroup. This would allow a realistic task description in that all the extra activity necessary due to errors can be categorised and implemented at each level. Experienced users make mistakes - they just make different ones to naive users. So, the final model would be:-

subgroup a.

subgroup b.

User model  $_1$  + additional device knowledge etc..... => User model  $_2$ .

(time  $t_1-t_2$ ). +Error model data.

(time  $t_2-t_3$ ). (5)

This model represents all the relevant factors which might represent the evolution of a user in the design of an interactive system.

#### 10.4 USERS INFORMATION AND KNOWLEDGE REQUIREMENTS

As has been said, it is possible, by outlining a groups generic interaction repertoire, a machines functional capability and knowledge requirements required to perform a set of kernel tasks, to evaluate a machines ability to support a user in the performance of the task. In theory, requirements

need not be in the knowledge format but in any format required to focus on a particular issue (dialogue, information are useful alternatives). It is this flexibility in terms of choice of interaction medium which makes this technique so powerful. The evaluation framework (as described in later chapters) remains the same however many media can be used to fill in this framework and hence, many aspects of HCI can be focussed upon.

The users requirements represent a set of functional and informational conditions which must be satisfied to enable the user to bring the machine into their desired goal state or a set of requirements which enable the human and computer to work together. The role of the interactive systems designer is to aim to meet those conditions in whatever form they may occur. The flexibility of this tool enables designers to meet all of the important requirements for their system.

Computers are designed to solve human problems and therefore this should be the predominant design perspective. Technical or functional advancement cannot be said to be purposeful unless it is specifically designed to meet a specific requirement. Work on improving interaction media or computer power does not facilitate better HCI in itself. These are only means for implementing better interfaces, not solutions in themselves. Hence, any technique which can describe users requirements flexibly and accurately is useful.

So, techniques are essential for specifying users requirements in many forms and the more flexible and powerful the techniques the greater the likelihood of identifying new and unresolved requirements which could improve usability. The framework offered here is a structure within which information on all types of requirements could be modelled simply by focussing the analysis on different levels of interaction (and the different dimensions within those levels).

In the 'interaction device knowledge' example user requirements can be specified in a similar fashion to the user descriptions, the kernel of requirements of a set of tasks, or the particular requirements of an individual task. For instance, the 'Login' requirements for a given machine:-

DEC: POSITION (ON SWITCH).

PROC:SELECT (PRESS).

DEC:POSITION(DRIVE).

PROC:INSERT( BOOT (DISC)).

DEC:FORMAT(LOGIN).

DEC:USER NAME(USER #).

PROC:TYPE (STRING).

DEC:FORMAT(PASSWORD).

DEC:PASSWORD(BLAH).

PROC:TYPE (STRING).

This can be easily evaluated against the kernel that user groups may possess or used to generate task instance descriptions which can be matched with descriptions produced by the user kernel.

Information requirements are inherent in this description, in terms of declarative knowledge; for example, a legal password or user name is required.

## 10.5 TYPES OF EVALUATION

Evaluation can take place on two levels within this method. At the knowledge-level, in which case the users generic knowledge structure is mapped on the knowledge requirements model. Any discrepancies would indicate deficiencies in the assumptions of the designer (as embodied in the knowledge requirements model) or would highlight need for training in the user (as embodied in the generic knowledge structure). Care must be taken here, however, to avoid a self-proving model: Ideally, the two models would be formed independently using the same method to ensure that an experimenter does not unconsciously generate a knowledge requirements model which reflects the known structure of the users generic knowledge model.

Alternatively, evaluation can take place at the task-level in which case the users generic model is tested to ascertain whether it encompasses a task description generated by the knowledge



requirements model. Not all task-instances will be describable using the generic task-model and user feedback will be required to ensure that the task instances generated for the purposes of evaluation are accurate, realistic and meaningful.

## 10.6 DECISION SUPPORT

Specific computer systems may require a specification which focuses on user information requirements. These demands may be as vital in the performance of the task as knowledge of interaction strategies. An example of these requirements can be found in information systems which allow users access to a large body of information, perhaps for the purposes of gaining assistance in making decisions.

To realise the users information requirements it would be useful for the designer to have an explicit model of the typical information access patterns exhibited by the proposed user of the application. Since information is an abstract commodity the only way in which such demands can be expressed formally is via a behavioural study of the contextual task requirements behind information use. The ability to exhibit the generic activity associated with the handling of information is a necessary component of many systems. Strategies for classifying, prioritising, categorising and processing information by source and other criteria, and incorporating realistic prioritisation and filtering in these activities, are necessary. The designer requires a method by which these strategies could be reliably and accurately detected and formally specified for incorporation into an application. This would ideally include prioritisation, scheduling and ordering strategies inherent to that particular application.

An example of this is 'dealing room' activity, such as that encountered in the Stock Market where specific structured information is required for dealers to make rapid decisions on market activity. These requirements are usually context, experience and task-related and involve information from given sources in given order or format for action to be taken. Hence, a model which takes into

account a users knowledge of the task-environment is required. Additionally, a hard time constraint is often encountered with the meeting of that constraint often determined by the order of activities. Therefore, it would seem useful if some abstract evaluation could be performed on the proposed temporal strategies before their implementation. This would be possible with the proposed temporal calibration scale. The dynamic use of information or knowledge in the performance of tasks in given environments is what characterises the above situation. This thesis offers techniques for the capture, structuring, ordering, application and evaluation of such information or knowledge especially in the design of interactive computer systems.

### **10.7 PHYSICAL REQUIREMENTS**

Another interpretation of the matching of an interface design ergonomically to the state of the user could be in terms of meeting the users physical requirements. These can be incorporated into a model alongside the knowledge or information requirements.

These may not be describable in terms of universal physical or cognitive human properties but perhaps representative of a particular user group, for example the blind or the deaf. Their requirements are perhaps not quite as obvious as they seem. For example, a blind person can obtain absolutely no visual feed back so even the most simple feedback message (implicit and explicit) must be transmitted through alternative channels such as audition or touch. There is a lack of tools which are sensitive to the identification of such physical or 'haptic' requirements in the design of interactive computer systems. Such requirements are common to all users, to a greater or lesser extent, and this thesis offers a framework whereby this type of problem can be addressed and the underlying requirements identified.

### **10.8 A FORMAL DEFINITION OF USABILITY IN TERMS OF USER REQUIREMENTS**

An Interface can be said to display usability if *it offers such features that all of the users relevant*

*explicit and implicit requirements are satisfied for the purposes of performing a given task.*

Identification of relevant features is the key activity in this form of analysis. Generic features which exist across user populations and user task strategies are the clearest, however this also includes dynamic features of HCI such as the application of knowledge or interaction between user and system features. The user is the open part of the system and is sensitive to many types of stimulus which can affect task performance (Edmonds1986) hence, many types of requirements need to be addressed if the users model of the system is to be formalised.

These possibilities can only be explored by examining the many unique and individual types of feature that users employ in solving actual problems during interaction. These features must be reflected in the capture, modelling and the resolution of given combinations of features within the interface. The tool offered sets out to display this capability.

## **10.9 STRATEGIES FOR OPTIMISING USER PROCESSING CAPABILITIES**

Meeting the requirements of the user does not necessarily entail giving the user what they ask for. The optimum configuration can only be achieved if the designer offers facilities which enable the user to optimise their existing functional or problem-solving capabilities or to exploit the inherent strategies which already exist for optimising them (mnemonics, for example). If this is to be effected an accurate model must be elicited of the optimal behavioural strategies used to exploit these abilities by identifying the optimum generic task behaviours. Subsequently, these natural strategies or familiar optimal ones must be incorporated into the interface such that the requirements necessary to utilise them are satisfied by the interface.

The study of articular types of use however would yield minimal benefit in the design of a general access information system. Professional users may show more rapid evolution than casual users as they can expend more time in identifying optimal methods of interaction due to the greater benefits this may yield and because they are more able to exhibit exploratory behaviour and recover with grace from any error. They too, however, maintain a working kernel of knowledge which

transcends other knowledge acquisition. The kernel of a professional user is, however rarely relevant to the design of optimal computer systems. This is due to the fact that the professionals conceptual model is different to that of a typical user. Hence, the assumptions which can be made by interface designers in serving this type of user are both more numerous and less relevant.

This chapter has introduced the notion of interaction states and discussed how these states might change during interaction. This lead on to a the description of method for evolving a user model due to changes during interaction: this form the basis of the empirical model. The ability of the functional description to support the users knowledge requirements is also introduced as one of the evaluation methods to be used later.

## 11. DYNAMIC HUMAN-COMPUTER INTERACTION

This chapter presents the objectives behind the work described in this thesis. The chapter also presents the case for the study of the dynamic aspects of HCI and discusses those aspects of HCI which need to be addressed and accounted for in the evaluation of dynamic Human-Computer Interaction. In addition this chapter discusses *why* it is so important to account for these features in the design of Human Computer Interfaces. The chapter presents some relevant examples of HCI behaviour to illustrate the importance of dynamic modelling. The techniques offered within this thesis are placed in the context of number of current equivalent techniques and the unique contribution of this work is identified. The requirements for both the theoretical framework and the exemplar design tool are summarised and the desired empirical tests and proofs are outlined. Finally, the approach adopted for the empirical work is justified in the context of the stated requirements.

### 11.1 DYNAMIC HUMAN-COMPUTER INTERACTION

The principal objectives of this work are:

1. To provide a practical framework for the generation of design tools for the integration of dynamic features of HCI into the design of computer interfaces
2. To demonstrate such a design tool and hence offer support to the notion of dynamic HCI

The framework for dynamic HCI essentially contributes in two ways to HCI

1. It allows the generation of design tools which bridge the gap between static description and prototype - action models of interaction which help designers build prototypes
2. It provides a simple framework for the analysis and understanding of complex HCI

behaviours to allow them to be integrated into the design process

In accounting for these two aspects of HCI it is believed that this work contributes significantly to HCI design practice.

### 11.1.1 Describing dynamic situations

This section presents an account of the HCI behaviour for which the framework for dynamic HCI has to account. The dynamic activity which occurs at the Human-Computer Interface can take a great number of forms. All of these types of activity would be accounted for by an ideal framework for dynamic Human-Computer Interaction. This section presents a taxonomy of the types of dynamic interaction behaviour (see figure 8) which the author has identified in the literature.

Dynamic Model Type	Dynamic Behaviour
Information Models	Implicit knowledge models Domain Specific Information models Large or Complex Information models
Action Models	Goal Driven Models Physical Interaction models 'Doing' Models
Attribute Interaction Models	Experience Attributes Environmental Attributes Open system Attributes Individual Attributes Ethological Attributes
Holistic Models	Synergy Models Holistic Models Interactive Models
Pragmatic models	Heuristic/ Pragmatic Models Real-time Models Adaptive Models Utilitarian Models Motivational Models Interrupt Models Error/ Forgetting Models

**FIGURE 8: DYNAMIC BEHAVIOUR CATEGORIES**

This discussion of dynamic HCI, and the evidence to support it in the literature, is approached using this structure, rather than in the perhaps more conventional narrative fashion as the author believes that the application of a concrete structure to this very complex area helps to scope and present the area more clearly. The taxonomy provides a theoretical framework which accounts for the behaviours and theories found in the literature survey. The above categories refer to the types of HCI which cannot be accounted for well using current specification techniques. It is postulated that this situation obtains because current techniques do not address the dynamics of behaviour. The taxonomy breaks down the dynamics of HCI behaviour into its fundamental categories and types.

This section presents some example interaction behaviour for each of the types of interaction and an accounts for the importance of each of the types in design.

### 11.1.2 Dynamic information models

The first type of dynamic model is the dynamic information model. This is a model of a construct formed by the user to enable them to understand and exploit complex, large or deep information structures. It was hypothesised that any model of the way in which humans process large amounts of information needs to be dynamic to reflect the inherent processes which humans undergo to handle information. This area has been addressed in Chapter 7 and a range of evidence was produced to support this type of dynamic model (Newell 1965, Selfridge et al 1960).

This first type of dynamic model within this category refers to the implicit knowledge which underlies information models, which was discussed in Chapters 7 and 10 (Johnson 1986). An understanding of the way in which users employ knowledge in the performance of tasks has been shown to be useful in the design of user interfaces and a number of techniques do exist for the modelling of users knowledge, however current techniques do not attempt to address the dynamic application of knowledge in the performance of tasks. By examining the influences that changes in user's knowledge exert upon their task behaviour it would be possible to design interfaces which meet user's behavioural needs by looking at the knowledge model. This is a dynamic model as it examines the changes in the users knowledge model and the complex interaction of that knowledge with task

performance. This area is addressed by a design tool which has been developed within this work and significant variations in behaviour have been linked to known variations in the knowledge model. The evaluation of this tool is described in Chapter 14.

Domain specific information can be relevant in the design of a computer interface. Chapters 5 and 6 looked at the provision of interfaces to support specific tasks and user groups and the knowledge of work domains which needs to be modelled to provide a useful interface. This extends to the provision of domain related information models for decision support systems. Tools are needed to model the application of users knowledge of the work domain in the performance of tasks if interfaces are to be provided which fit in with those domains. If this can be done then the user can exploit knowledge of their own work domain to make decisions about how to operate an interface. For instance, an accountancy interface can be presented in the form of a number of ledgers and the users can navigate through these according to their own experience. If the designer is aware of this domain knowledge then it can be exploited in the design of the interface. This is a dynamic model as the designer also needs to be aware of how the user dynamically interacts with his domain so that it can be simulated in the virtual machine and additionally, the model has to cope with a changing and wide-ranging body of knowledge.

Thirdly, the user can manipulate large amounts of complex information by applying their own active processing, or coding schemas to the data. Here, users build dynamic information models of relevant data (Newell 1965). These models are constantly updated and amended to include new categories to handle new information. Modelling techniques to address this type of behaviour would allow the users inherent coding schemas to be exploited in the design of the user interface. The provision of configurable 'hypertext' systems is one way to allow users to personalise their own stores of data so that it remains accessible. There are, however, no current modelling tools to assist in the production of individual or co-operative hypertext systems and this work aims to assist in the production of such techniques.



### 11.1.3 The action models

The second form of dynamic model is the action model. These are dynamic models, of the users desired goals, their physical or procedural interaction repertoire and their perceptions of the task which they believe they are performing. The user's model of their desired goals changes dynamically to allow them to rationalise and amend their goals throughout the task. This allows users to change their strategy according to their experience of what is possible at the interface and pragmatic task considerations. For example, a user might set out to create a complex document within a given word-processor, however whilst doing this they may discover that the interface does not offer sufficient tools to allow them to create such a document. The user would then amend the goal model to account for this. If interface designers are aware of such goal models then the interface can be designed to allow the users to see the tools available and match their goals to the tools.

Secondly, the user will have a dynamic interaction repertoire which is changing through experience and understanding of the interaction devices available. This may be stored in procedural memory, as is discussed in Chapter 7, however an understanding of how this model is formed and dynamically adapted would be useful in allowing interface designers to present interaction devices to users, make them easy to learn and understand how the user might exploit the devices. For example, if an action model of how a users action repertoire with a mouse is constructed this would enable designers to design new devices which complement the mouse in terms of supporting different pointing tasks and yet perhaps build upon the physical interaction repertoire of such existing devices. This is very much supported by the calls for exploration into the use of interaction devices found in the literature (Buxton 1986) and, indeed, it is impossible to extend the range of applicability of physical interactions and the devices which support them unless there is a clear understanding of the processes which underlie such interaction.

Thirdly, a dynamic model of how users perceive the tasks that they are carrying out would enable designers to present feedback to the user which maximises comprehensibility. For example, if a designer wants to make a WYSIWYG (What you see is what you get) system then an appreciation of how the user interprets the feedback given to them, in the task context, would explain many of the

user's assumptions. Understanding what users think they are doing is difficult to identify from explicit behaviour however an understanding of how users perceive interface stimuli would contribute to the design process in allowing designers to offer feedback which is meaningful to a range of users. This type of model is necessary due to the application of 'active perception', discussed in previous chapters. Users see what they want to see, or rather what their experience tells them to see, and it is only by understanding this dynamic application of knowledge to the perception of interfaces that appropriate feedback can be offered (Green and Doubleday 1988).

#### 11.1.4 The attribute models

The third class of dynamic model is that which looks at the interaction between interface attributes. This type of model reflects the diversity of attributes in the users environment which can have an impact upon interaction (Van der Veer et al 1983). This form of analysis of HCI has been acknowledged to be a powerful conceptual framework for understanding interface behaviour (Shackel 1986). Were this type of model to be realised dynamically it would enable designers to understand and conceive the effect of the interaction of the attributes of any class of users with any type of system and environment by studying the process of interaction.

Firstly, there are the experience attributes brought to the system by various users. An appreciation of how users apply existing experience in interaction with new devices would allow designers to configure devices to optimise transfer of knowledge. A dynamic model would show how users develop their experience and change this in response to new interaction heuristics. Hence, this model differs from the knowledge model described above in that it refers to the users total experience of the interaction environment rather than simply the knowledge of tasks. For example, desktop interfaces present consistent visual cues across all applications however a dynamic model would show where consistency could be violated without any loss of comprehensibility. Similarly, the user represents the 'open' part of the interactive system (Edmonds 1986) and if interfaces are to be developed to exploit the experience of the new range of users then a dynamic and ongoing model is required of the new body of experience. For instance, future users might be more computer literate than the existing population so designers may be able to make more assumptions about the users familiarity with

computers.

Secondly, there are the environmental and social attributes which impinge upon interaction. These attributes have a dynamic effect in that they can radically affect the quality of interaction whilst they play no direct part in the interactive process. For example, the ability of an interface to support social facilitation, such as the ability of a real-time teleworker to log-off and talk to other workers, can significantly affect the performance of the user. Alternatively, the users immediate work environment may display significant attributes which affect the performance of the interface. For example, a booking clerk may record details to be input onto paper before input and amend the details on the paper according to the state of the interface or the customers requests. Hence the paper system represents a dynamic memory which forms part of the work system. A dynamic model of the interaction would account for the changes in the recorded information as well as the attributes of the system, the user and the customer and would provide a valuable record of the changing information within the worksystem as a whole.

One way to approach this type of model is to use ethological techniques to analyse the user *in situ* and to record a significant sample of the behaviour of the user and the environment in which they work. This would equip the designer with sufficient evidence to build the dynamic model. This is the type of technique used in the empirical study. Use of *any* technique which allows designers to build models which bear closer resemblance to what happens in the real-world is a desirable activity as it ensures that the design of interfaces meet the dynamic requirements inherent in user's actual behaviour.

Another type of attribute model is that which looks at the individual attributes of users and examines their unique requirements for interface design. Any model which seeks to address individual's requirements needs to be able to react dynamically to individual attributes. This is addressed in the discussion of 'cognitive style' (Robertson 1984) in chapter 6. Hence, for many types of system a generic model suffices to meet the needs of a range of users types however, certain attributes of a system however can demonstrate usability advantages by meeting individual usability requirements. Allowing users to build customised file structures with no hierarchical constraints is one example of

meeting users individual needs or offering user's a range of input styles to meet their preferences is another. A dynamic model which focusses on the specific areas in which customised interfaces could offer usability advantages would be useful. The model would need to be dynamic as there would be no predefined set of individual attributes which could be defined, hence the model would need to adapt dynamically to new constraints. This type of model again recognises the user as the 'open' part of the system and would imply a degree of flexibility in any interface to support that model.

### 11.1.5 The holistic models

The fourth type of model of dynamic HCI is that which describes holistic aspects of the system. The traditional approach to the analysis of HCI takes a reductionist perspective and breaks down the interactive process into discrete notional units from which a coherent model of behaviour can be built (Card et al 1985). An alternative, and complementary approach to this is the use of a holistic model of interaction. Here, the interface is evaluated as a discrete entity. This is usually achieved using subjective reports of usability in response to attitude questionnaire or interview (Poulson 1987). These techniques could be supplemented by a dynamic model of the users perception of the whole interface. This would provide a model of users reports concerning all non-specific aspects of the interface. This model needs to be a dynamic one to cope with the new criteria upon which such reports could be based. For example if the colour of the computer exerted an effect on usability - if it is the same colour as a system which the user has found usable in the past - then a dimension would be built into the model to account for colour. Hence, the model would be continually changing and adapting to encompass new measures to ensure its validity.

Another type of holistic model which needs to be accounted for is that which describes the 'synergy' of HCI. Synergy describes the way in which the user and computer can work together to achieve more than the sum of their individual output. Current tools exist to examine 'function allocation' (Fitts 1986) and determine whether the user or the computer should be allocated a task. A dynamic model offers a tool for examining those tasks which are best performed co-operatively between the user and the computer. By forming a dynamic representation of those interactions which by the process of interaction are able to improve upon the performance of the user or computer alone it

would be possible to account for those aspects of the interaction which governed the increase in performance and apply them to other systems. Hence, this is a dynamic model as it looks at the 'process' of interaction.

The changes which occur within the user's knowledge state during, and as a result of, interaction would be addressed by a dynamic model: this is termed an 'interactive' model within the taxonomy. These changes are interesting to designers as they would form the basis of an adaptive system which could present an appropriate interface to the user dependent upon the users level of experience. These changes might consist of the user using a command-key form of input rather than using a menu after the menu becomes too time-consuming. In this case the user model is different at the end of interaction than at the start and hence it is a dynamic model. This aspect of dynamic interaction is the basis of the design tool which is presented at the end of this chapter. Changes of this type can also occur due to the user forgetting a command sequence during interaction and reverting to previous behaviour. These changes are also the subject of an experimental study later in the thesis (Chapter 15 -Study 1).

### 11.1.6 The pragmatic models

The final class of dynamic or behaviour which needs to be accounted for are the pragmatic aspects of HCI. Perhaps the most complex yet illuminating type of dynamic interaction which must be addressed is that provided by the users adaptation or response to pragmatic aspects of the task or interaction using 'heuristic' knowledge (Malone 1984). In spite of this the pragmatic models remain relatively poorly explored in the current literature and this is one of the reasons why this aspect of dynamic behaviour forms the basis of a number of the empirical studies.

The most realistic model of interaction is a pragmatic one which enables users to respond to changes in the interface state, initiate behaviour without a logical basis for action and use heuristic knowledge of the experience of interfaces in the formation of interaction strategies and the solution of problems. This type of model, however, presents many difficulties to the modeller as it refers to poorly-formed cognitive constructs and a very large range of non-specific stimuli which can act as

pragmatic cues to behaviour. A dynamic model of pragmatic aspects of the interface would, however provide a rational basis for the analysis of a large range of important interaction behaviours. The work described earlier looking at task-paths, and offering tools for their specification, is a possible pragmatic model of interaction: it provides a framework for examining how users make pragmatic decisions at given points in interaction by the use of heuristic reasoning.

The performance of any task which has real-time constraints on the behaviour of the user could also be addressed by dynamic modelling. This is exemplified in the discussion of real-time tasks in chapter 9. The designer of an interface to support a user in the performance of a real-time task needs to be informed as to the constraints of the task, the cognitive style of the target user group and the likely interaction of those attributes. Real-time tasks are often safety critical and hence if the performance of the user is to be guaranteed a realistic simulation of the interaction needs to be explored before the system is implemented.

The use of utilitarian decision making in the determination of interface behaviour would be addressed by a pragmatic model. By describing the ways in which users perform cost-benefit analyses of actions it would be possible to design systems which encouraged exploratory, learning behaviour which has been shown to form the basis of rapid learning of an interface (Kennedy 1977). It would also be possible to present functions in a form which allowed the user to determine correctly which was the appropriate command for their desired activity. The discussion of 'accessibility' in chapter 6 also points to an application for this type of model: by understanding how users perceive the value of commands it is possible to determine their required accessibility. Deciding which facilities to offer to the user in an immediate form (perhaps as an ICON), and which to embed within the system (perhaps as a menu), is a major design activity: this type of model would be invaluable in making such decisions.

The role of motivation in determining interface behaviour also falls within the pragmatic class of behaviour. By examining the types of interface which motivate or demotivate individuals in the performance of given tasks it would be possible to present a system in a way which optimises interface behaviour. Motivation is determined by the interaction between the users ongoing

motivational state and the appeal of the cues provided by the interface. It has been shown in previous chapters that by presenting interfaces with a high appeal motivation can be significantly affected (Malone 1983) and an understanding of the dynamics of which interface type appeal to given user groups would provide key information for the design of user interfaces.

All users make errors and the coherent study of these errors would be the subject of a dynamic analysis. The study of errors is useful to designers in designing 'foolproof' systems which prevent users from making costly mistakes. An example of this is the systems prompting of the users to save changes on closing a file. A dynamic model of error behaviour would relate the type of errors made by different types of users to given tasks: the relationship between task type, error type and user group is a dynamic one. This model is dynamic as error types may be apparently unrelated to task behaviour and a 'deep' analysis of the precursors of behaviour would be required to facilitate its understanding. This type of analysis is the subject of an experiment presented later within the thesis (Chapter 15 - Study 2).

The final pragmatic model of dynamic behaviour is that provided by interruptions. This behaviour does not fall into the environmental category as the effect of an interruption is dependent upon its interpretation by the user: hence, it is a pragmatic feature. Again, interruptions are non-specific stimuli which can only be addressed within a dynamic model of interactions. The design of interfaces to support interruption would allow the user to rapidly suspend the task state and resume after the interrupting stimulus has been handled. The effect of interruptions upon interaction is investigated in an empirical study which is described later (Chapter 15 - Study 4).

#### **11.1.7 Application of the theoretical framework**

The theoretical framework highlights some the conceptual tools which already exist to support such dynamic models and outlines how dynamic issues are already being dealt using appropriate representational techniques. For example, the complex dynamic flow of information through the interface can be rationalised and understood using one of the range of 'layer' models. The reason why such a model is necessary is that there are a great number of simultaneous information flows

occurring and the layer model compartmentalises these flows into segments that the designer can analyse and use. The theoretical framework presents a platform upon which new tools of this kind can be built.

Chapter 13 provides a number of principles which characterise the behaviours described above. This assists in the application of the behaviour in design. Needless to say, it is not suggested that the principles account for all of the dynamic behaviours in the framework: they simply represent the behaviour of the dynamic behaviours to be investigated. Such a task is well outside the range of this study and is indeed an ongoing research task. The principles do, however bring the huge amount of dynamic HCI behaviour into a manageable form for use by the designer and provide the basis for the design of experimental studies to test the theory.

## 11.2 CURRENT HCI DESIGN METHODS

One of the fundamental problems which this thesis seeks to address is the lack of available notations, representation tools and specification techniques for the description of dynamic interaction. Dynamic interaction cannot always be accounted for in the design process and hence a wealth of valuable information is lost. In moving from the paper specification to the first prototype much of the information about how users work in the application domain is lost as the tools available for the expression of behaviour do not account for the dynamic interplay between user, system and environment.

The capture and representation of these dynamic features would eliminate the need to iterate the prototype in the early stages as it would anticipate many of the simple interaction problems which a sophisticated prototype is designed to identify. A dynamic model would provide guidelines to the developer in the production of the first prototype. In addition to this, the use of an analytical formal notation would provide concrete diagnostic feedback as to the source of the interaction problem whereas a prototype may simply indicate that the problem exists.

If the designer is to understand the dynamics of behaviour at the interface they require a set of tools



to capture and represent the behaviour which exists, and to apply it to design problems. This work seeks to provide a conceptual framework for dynamic phenomena along with the basis for the development of a set of design tools.

### 11.2.1 Frameworks and methods

The dynamic interaction framework represents a 'methodology' in that it offers a theoretical framework within which range of design and evaluation tools can be developed. It is not, however, a structured methodology in the current sense of the term, as it does not offer guidelines for the application of a set of existing tools, rather it assists in the identification the dynamic issues within Human-Computer Interaction for which evaluation tools can be developed in the future.

It is interesting, therefore, to contrast the framework described here with current techniques for specifying HCI. This facilitates the identification of the areas in which current notations are successful and perhaps highlights the deficiencies in those notations for the specification of dynamic HCI.

In SSADM (Structured Systems Advanced Design Methodology) (CCTA 1986), for example, HCI is represented by the data flows across the system boundary. This approach involves the specification of the human interface using a formal notation to describe the movement of data through an application. This is common of many tools of this type which are very popular in the design of computer systems - Yourdon, Merise. The descriptions of dataflow inherent within the notations in this method are rich and in cases could be said to be dynamic - the Entity Life History notation specifies the temporal behaviour of data objects and outlines some of the prioritisation and scheduling constraints upon that entity. The method contributes to the development of prototypes by specifying the screen items which must support the input and output of information.

The method is limited, however, in the way that it addresses the behaviour of the user. The ability of data notations to capture and represent the work practices of the user, the social interactions within the workplace and the physical interaction repertoire of the user is limited. The model which is formed of

behaviour is often a poorly-formed representation of the users actual or desired behaviour and this does not provide the designers of information systems with information of sufficient quality to enable them to make decisions about how to design a computer interface which reflects and supports the behaviour of the user - hence, these interfaces can show poor usability.

The author contends that the deficiencies within these notations arise largely because they fail to address the dynamics of the users behaviour within the context of interaction with the computer, the environment and the task. Comprehension of the social, environmental and physical precursors which underlie the users behaviour at the interface requires an understanding of the implicit knowledge employed by the user, their goals and intentions and the user's physical interaction repertoire, to name only a few attributes. If a range of design tools were developed, based upon these precursors then the current body of techniques for specifying HCI would possess another dimension.

Many task-analysis techniques do provide a sophisticated model of the task, and the knowledge underlying it, and make steps towards a dynamic representation, however their representations often remain largely nominal descriptions of task events rather than flexible descriptions of the interaction between task, system and user attributes. A number of the notations used in task-analysis label behaviour and do not examine, for example, the way users actively interact with or manipulate application or system entities. A number of techniques do, however, provide attribute based models of the user and system (Johnson 1986, Payne 1984) to allow the production of dynamic models, and the framework described here hopes to accelerate the development of these and other techniques. By the provision of a structural tool for the categorisation of dynamic phenomena, such as that described within this work, a conceptual framework exists to enable HCI workers to comprehend the area and develop techniques in a methodical fashion.

### 11.2.2 Deficiencies of static evaluation techniques

This section discusses further what dynamic models offer over and above their static counterparts which facilitates a more accurate, realistic and useful evaluation.

The notations which are used to implement static models of interaction are not sufficient for the modelling of dynamic situations. This class of notation does not contain the vocabulary or possess the analytical power to capture and represent a number of key aspects of dynamic HCI. These aspects include:

- Parallel activity
- Time constraints on users performance
- Complex cognitive operations
- Action descriptions of users
- Individual user groups

There are many other aspects of dynamic HCI which cannot be addressed by static notations and these can be derived from the taxonomy described above. It is, however, worth examining some examples in more detail. One of these examples is the deficiencies which static models display in the lack of support for the description of changes in the users interaction knowledge during and between interaction.

Static models have been the subject of considerable discussion in previous chapters and one element of the definition of such models has been arrived at due to the fact that they only propose a unitary model to represent the user throughout the course of interaction. Hence, they present a general model of user cognitive attributes which does not represent the variations in user behaviour due to experience. This can be attributed to the fact that such models have no explicit representation of temporal issues or the changes in users which arise as a result of learning and interaction. These models have no facility for describing variations between users task behaviour from different groups or levels of experience as it would occur in a given task.

It is possible that models of user behaviour can be derived from a collection of general cognitive attributes, however since no record is made of the applicability of given attributes in particular contexts or user groups this model may not be reliable. Static models do allow an amount of

empirical data concerning universal human cognitive attributes to be referred to in the design of interactive systems and this data is invaluable. These attributes will apply to users at any time during interaction and help supply normative parameters concerning the speed of processing of given aspects of the HIP system, however the primitives are limited in scope if they are to be combined to describe the actual behaviour of the user. Predictions can be made concerning performance of the user in terms of the summation of a string of cognitive primitives but the actual behavioural repertoire of a given user during a given interaction cannot be deduced from this.

The properties of the HIP system, for example, do not determine whether the actual interaction strategy or 'cognitive style' applied by the user is sound or optimal for that task. Ease-of-use is best determined by the analysis of the behaviour displayed during the execution of real-world tasks utilising actual user methods. We need to know about the user or user groups behavioural and procedural repertoire of the user if accurate and realistic models of the user are to be formed. Dynamic models seek to model real-world task behaviour in this way and by focussing on a limited user and task domain, rather than offering a general processing model, can represent ongoing interaction and the changes in the user during the execution of tasks. The dynamic model also represents error-behaviour as a natural component of group and individual activity.

Static models exploit a hypothetical, optimal task structure which may or may not be utilised by users and which often involves expert or error-free performance. This task structure may bear only nominal resemblance to an actual strategy. The techniques offered seek to represent a limited, or definable set of attributes of the behaviour of users performing tasks (Carter 1986) by focussing on the capture of behavioural data and the analysis of that data in a structured fashion. This approach advocates the study of limited user populations which are amenable to controlled statistical analysis.

Concentration on an abstract task model leads to the modelling of task defined primitive behaviours and whilst this offers the designer a useful framework from which to assess interaction it does not comment on the actual behaviours which are displayed by or appear significant to users. This latter type of behaviour would seem to be the type of description which would be essential in describing HCI for the design of interactive systems if an accurate and realistic model of interaction is required. Task

independent primitives need to be defined which are derived from an empirical examination of user or user group behaviour (Johnson 1986). Whilst user behaviour is generally related to the task-structure the particular cognitive and behavioural interpretation strategies employed by users to optimise interaction behaviour mean that formal task-behaviour is often not displayed. It is the user defined task elements and knowledge requirements (Johnson 1986) which need to be identified, modelled and utilised in design along with the methods by which these elements are employed.

Static techniques utilising the assumptions described above are not necessarily invalid as a result of these deficiencies as they can be applied effectively in situations which meet these constraints and in evaluations where comparative values of optimal performance will suffice. User behaviour in the use of interactive systems does not, in some cases, meet those constraints and optimal performance values are often not representative of 'average' user behaviour.

In summary, the designer must decide whether a static notation meets the requirements of their application and often such a notation will suffice. This does not, however mean that designers should adapt their work to use only static models. Such models do not represent the full richness of the information which could be available concerning the users interaction behaviour as a static notation cannot capture or represent dynamic aspects of behaviour.

### **11.3 THE CONTRIBUTION OF THIS WORK TO HCI**

The work described within this thesis offers two major contributions to HCI.

First, there is a theoretical framework for dynamic Human-Computer Interaction. The validity of this framework is established by means of selected experiments and support from the literature. The framework is presented as a number of principles which provide the basis for the development of a range of novel design tools to evaluate dynamic Human Computer-Interaction. These design tools address a large body of behaviour which has hitherto remained untapped in the design of computer interfaces.

Second, a design tool is developed and presented to illustrate an example design tool which could be developed within the framework. This tool is assessed by performing an example evaluation and testing the consistency of constructs upon which the tool is based. Additional support is offered by looking at the expressiveness of the tool in indicating usability differences between two interfaces. This tool allows changes within users interaction knowledge during interaction to be accounted for in design. This can exert a direct impact on the design of adaptive systems and provides a concrete example of a dynamic design tool.

The role of this work within the HCI community is that by establishing the validity of the dynamic HCI framework other designers will be encouraged to develop evaluation tools based upon this framework and employ them in design.

#### **11.4 THE CONTRIBUTION OF DYNAMIC HCI TO DESIGN**

The majority of current systems development projects now involve the construction of a prototype to determine whether a proposed computer application is viable as practical tool and whether it meets the needs of its target user population (Whitefield and Long 1986). The prototype will typically be developed from an interim specification or operational requirement. It is very difficult to ascertain, however, from the specification whether the chosen implementation of the specification - and there are often very many alternatives which can meet a specification - will be the best one to:

- explore all of the design options
- offer an interface which fits with the users existing task or interaction repertoire
- provide an interface which reflects the users changing perception of the task by providing various tools

The work described here seeks to help designers in obtaining this type of information. A number of techniques exist for 'bridging the gap' between specification and prototype:

- a. **Usage Scenarios (Young 1986):** Designers select a number of typical tasks that the interface

will need to support and task-part in a task-walkthrough to hypothesise how the interface might address the users requirements for those tasks

**b. Constructive interaction (Miyata 1986):** Two subjects are asked to solve a problem using a set of instructions provided by the experimenter. These instructions can be used to simulate the operation of the interface and the suitability of a configuration can be assessed

**c. Paper simulations:** A paper simulation can be built of a desktop system and the users physical interactions with the objects can be recorded

The role of the theoretical framework offered within this work is to assist designers in the development of tools like the above to help them explore the dynamic context of interface designs. By specifying the dynamic context clearly it is possible for future developers of tools to address specific aspects of dynamic behaviour by designing specific instruments based on principles and attributes derived from the theoretical framework.

## **11.5 THESIS STATEMENT**

The principal objectives of this work are:

1. To provide a practical framework for the generation of tools for the integration of dynamic features of HCI into the design of computer interfaces
2. To demonstrate such a design tool and hence to offer support to the notion of dynamic HCI

A number of premises have been identified which express the assumptions made before this work could proceed. These premises are:-

1. That interaction is dynamic

2. That it is an expedient design strategy to treat it as such

3. That a dynamic model can be formed to demonstrate that expediency

## **11.6 REQUIREMENTS FOR STUDIES FOR THE EVALUATION OF DYNAMIC HUMAN-COMPUTER INTERACTION**

This section outlines the requirements which must be met if a design tool for the integration of dynamic HCI is to be evaluated. The section also describes how the theoretical framework for dynamic HCI should be evaluated.

### **11.6.1 Requirements for the design tool**

Firstly, it is essential that the tool to be evaluated is an analytical tool as it is required to provide a coherent framework for the modelling of HCI. A procedural or action based tool, for example a prototyping tool, would be unsuitable as this would not comment in coherent way on the integration of dynamic HCI into design.

Secondly, there is consideration of the adequacy of an observational study of the use of a design tool. This would provide some evidence in favour of the tool but would offer little rigour. The alternative would be an empirical study of the use of the tool. The use of an experimental study to assess the design tool would be unsuitable as it is impractical to design an experiment which could adequately control the design process to enable true testing of the experimental variable. The chosen study is an empirical study which involves the use of the design tool to perform an evaluation of two interfaces. An empirical study using software designers employing the tool in their everyday work would have been more valid however this proved to be impractical.

The model described in the evaluation is not a specification tool as it does not specify a design for a computer system. The tool provides design information for the creation of a system hence it is a



design tool, however it also provides the capability to model interaction hence it can be referred to as a modelling tool. The tool does not, however, just seek to evaluate the interfaces involved hence it is not an evaluation tool. The evaluation exercise is performed merely to assess the model as a design tool.

The proofs which would be required to assess the validity of the notation could take a number of forms. The utility of the tool in a design is the most valid form of proof and this is assessed in the performance of an informal evaluation. Statistical proofs of the validity of the model are also a useful form of evidence and these are addressed in the form of correlation and variance tests. Thirdly, subjective assessments of the validity of the tool are also useful and this is provided in the form of feedback from the user.

Finally, the type of results to be produced by the tool must be considered. Nominal results would be inadequate for an evaluation as they would provide no assessment of the value of the tool. Cardinal measures are appropriate for this type of evaluation as they refer to a comparative value of good and bad for the two interfaces and hence suggest the measuring ability of the tool. Ordinal and ratio measures are beyond the scope of this study and indeed not appropriate for most studies of HCI (Chapanis 1982).

### **11.6.2 Requirements for the theoretical framework**

A full assessment of the theoretical framework for dynamic HCI is clearly well beyond the scope of this work. In order to test all of the types of behaviour described above and outline their role in design would require a detailed experimental programme for the evaluation each type of behaviour followed by design studies of the tools which arose from the framework.

A key test of the framework is in its ability to generate design tools which are useful in the integration of dynamic HCI into design. This is addressed in the evaluation of the design tool, however a longitudinal study of the use of the framework to consistently useful tools would be desirable. This is impractical within the timescale here although future work is planned to address this issue. It

would also be desirable to extend the study into the evaluation of the utility of the design tools in developing useful prototypes. Again, this is beyond the scope of this study and it must be assumed that the information provided to the designer would complement their existing experience in allowing them to design prototypes.

There is a clear role for both observational and empirical studies here in generating evidence for the validity of the model. Observation studies could provide a body of example HCI behaviours and the ability of the framework to categorise and account for the range of these behaviours would help validate it. This would provide useful support for theory, however a more rigorous empirical study would offer more concrete and meaningful feedback. A number of experiments based on the principles underlying the model would test the validity of the principles and could therefore imply support of the theory.

In summary, the ideal proof offered in support of the theoretical framework would again be design based, however since this is addressed by the evaluation of the design tool experimental proofs of the principles would best support the validity of the theoretical framework. The approach which must be therefore adopted tests a discrete number of the key principles underlying the framework.

## 11.7 SELECTED EVALUATION METHODS

This section describes the development of the techniques used to evaluate both the design tool and the experimental framework. The section also justifies that choice.

The default method for evaluating a piece of scientific work is to employ a standard benchmark method so that comparisons between work can be made on an objective basis. Unfortunately, no consensus exists as to the ideal method for evaluating design tools. Evaluation methods have ranged from no formal assessment whatsoever (Roberts and Moran 1984) to subjective assessment of tools by designers (Fowler et al 1989) right through to controlled experimental studies of the use of design tools and the effect they exert upon software quality.

The opportunity to perform a controlled study of design is very rare and the imposition of such controls would be impossible in all but the most rigorous of design environments. The use of less controlled, subjective methods of evaluation is a practical alternative; asking designers to report on the use of the tool within their work. This requires the presence of a number of well-motivated designers who understand the goals of the evaluation. This has also proved to be impractical in this case, however future work, planned to develop the external validity of the tool, will adopt this technique and this is described in Chapter 16.

### **11.7.1 Rationale behind the development of the selected evaluation methods**

This section outlines the rationale behind the development of the tool and framework and describes the metrics with which they are assessed.

The example design tool is one involving the modelling of changes in user behaviour with a TKS-like (Johnson 1986) method. The tool provides methods for the ethological capture of human behaviour, analysis of the behavioural data and modelling with the data to report constructively on the differences in interaction knowledge displayed by given user groups within the sample. The model is described in the following chapter along with a description of the testing procedures used. The tool is described fully in chapter 14.

The method selected to evaluate this tool is one of objective empirical evaluation using feedback from the users to validate the contents of the model. This feedback showed that the tool was capable of capturing and representing user behaviour accurately. A statistical analysis of the constructs in the model was performed in order to assess their robustness. This demonstrated the ability of the tool to represent meaningful groups of users. Finally, the performance of an informal evaluation using the model to make performance predictions against known criteria. This indicated the power of the tool to contribute meaningfully to design by commenting on the worth of each design. This battery of tests, whilst it does not offer the external feedback of a study using software designers provides a strong internal validation of the tool against relevant performance criteria.

This particular design tool was selected for development as it demonstrates one of the key principles concerned with dynamic interaction: the propensity of users to change as a result of interaction. The provision of a design tool which can statistically model the difference in users behaviour according to experience and group membership is seen to be directly applicable in the development of interactive systems. The tool allows designers to identify which attributes characterise particular user groups and account for these in the design process.

The choice of this design tool was also partly determined by the scope of the project: the ability to construct a tool from established techniques (Johnson 1986) brought it more within the range of the Phd work than developing a tool from first principles. It may have been expedient to explore a number of design tools on a much shallower level, however a full empirical analysis of one tool provides more coherent evidence of the value of the tool. The design tool is also based on a design cycle which shares many common elements to that used by many HCI studies and this suggests the appropriateness of this approach.

In summary, this design tool has been selected as it addresses some of the key deficiencies in current techniques and present a more realistic and accurate model of one aspect of dynamic user behaviour.

The tool does this by :-

- Incorporating a temporal dimension into the interaction model to identify how users change during interaction
- Focussing on limited user groups
- Eliciting natural user task behaviour (ethological)
- Performing a statistical analysis of the behaviour to validate the user groups

The overall objective of this evaluation is to show the utility of the proposed tool within design.

The method selected to evaluate the theoretical framework uses some of the key principles which support the framework as the basis for experimental studies of its validity.

The rationale behind this choice is that the literature offers concrete support for the majority of the principles however certain key principles remain largely unsupported. It is hypothesised that the performance of controlled studies of the remaining principles would complement the evidence in the literature to provide a full range of support for the theoretical framework. By restricting the studies to key principles it is believed that the empirical studies could be designed to offer significant support to the framework.

The author has considered the use of an in-depth observational style of empirical study in which examples of HCI behaviour are elicited to test the theory. Although this type of study will form a large part of the future work to follow this study it is not reported here. The experimental work would provide more tractable results in support of the theory and supply focussed and directed feedback which will enable the principles to be proved and updated.

The use of experimental studies which exploit the same user groups as the evaluation of the design tool allowed data on the tendency of users to commit errors with, and forget, given types of command. Because of controls over the user groups this data could be fed back into the empirical model of the design tool to extend that model. This would provide a significant contribution to the validity of that model.

The experimental stance of the work described here involves controls to enable the work to contribute to the body of knowledge on HCI. The phenomenae could be said to apply to any situation within which the conditions hold and hence the results could be extrapolated into other areas of HCI. This is a significant advantage of experimental studies.

In summary, the evaluation of the theoretical framework has been selected as it addresses the following issues:-

- It complements the evidence already in the literature
- It provides rigorous evidence of the soundness of the framework

- The feedback allows critical review and updating of the framework
- The study provides empirical data to amend the design tool model
- The study allows extrapolation of results to other areas of HCI

The overall objective of this work is to provide support for the theoretical framework for dynamic HCI via the experimental testing of key principles. The experiments are described more fully at the start of Chapter 15.

## **12. THE EVALUATION OF DYNAMIC HUMAN-COMPUTER INTERACTION**

This chapter describes a design tool for modelling dynamic changes in users. First, a design cycle is described to provide a framework for the construction of the model. Second, the model is described along with the empirical tests which exist for its evaluation. Third, the model is placed in the context of the theoretical constructs described in previous chapters - such as calibration scales. Finally, the grain-of-analysis and the validity of the model are discussed.

### **12.1 A DEFINITION OF EVALUATION**

The dictionary definition of evaluation (Chambers 1983) is 'to determine the value of' and this is, in essence, the purpose of this thesis: the work described aims to demonstrate the value, as a design tool, and the validity, as a concept, of the study of the dynamics of human computer interaction.

Within the HCI community a more specific definition of evaluation can be found and this typically refers to that set of practises and procedures which can be used to assess user behaviour, machine performance or man-machine interaction (Kidd 1987). The author has identified a typical design cycle for the field of HCI from a number of sources in the literature (Shneiderman 1986, Kidd 1987, Thimbleby 1985) and this forms the basis for the empirical evaluation to be described later. Some techniques use only a few of these stages however what follows is the complete design cycle:-

1. An observation phase in which a study is made of given human or interactive phenomena and effects and relationships identified which might be integrated into the design of a system.
2. An elicitation phase where a technique is developed for the formal or informal capture of the behaviour observed in the first phase and a body of relevant data on the situation recorded. For

example, videotaping of interactive behaviour.

3. A representation phase in which a formal model of the situation is formed which describes in a structured form all the interesting aspects of the interactive situation such that it is economical and offers easy access for the purposes of reference. For example, a task description.
4. An evaluation phase in which a potential design is assessed, with reference to an objective criterion, according to how well it supports the optimal performance of the observed behaviour. For example, a predictive evaluation using Reisner's formal grammar.
5. A build phase in which working prototype of the refined design specification is implemented.
6. A re-evaluate phase in which the prototype is evaluated formally using controlled studies and formal evaluation tools. This might consist of a controlled psychological experiment.
7. A final build phase in which all the data from the formal analyses and feedback from the evaluation of the machine is integrated into an 'optimal' design.

An alternative or complementary approach to design is the evolutionary approach. This involves the immediate construction of a prototype and the iterative evaluation of that prototype. Using this type of method, however, the initial stages of the lifecycle may be carried out implicitly and unconsciously incorporated into the prototype. There is, however no paper representation of the model. Clearly, the two techniques are complementary with some guidance being offered to the specification of prototypes by paper-based specification and evaluation techniques and vice versa. This approach has not been used within this work although it could no doubt produce useful results. The prototyping approach has not been selected as it represents a less analytical approach than that required within this work, and one which would not support the coherent development of tools for the generation of prototypes from formal specifications.



This design cycle forms the basis for the methodology which is to be used in the empirical evaluation. The next section describes the model which will be developed within that framework. It should be noted that this chapter more closely describes a design tool, along with a method for its evaluation, rather than an evaluation technique for user interfaces - although, since one of the methods for the assessment of the tool is an informal evaluation, there is an evaluation aspect to the work.

## 12.2 THE DESIGN TOOL

A design tool has been derived from one of the principles of dynamic HCI (Chapter 13). The example principle to be addressed in the design tool is that of 'change within user behaviour due to':

- i. Experience of interaction
- ii. Membership of a user group

This principle will be further explored by the modelling of 'short-burst' (Roberts and Moran 1984) interactive behaviour within the user groups.

The design tool is realised in the construction of a dynamic model of the changes in users behaviour. The structure of the model is outlined below.

## 12.3 THE DYNAMIC MODEL IN THE EVALUATION

Assessment of the paradigm of 'evaluation of dynamic human-computer interaction' is addressed by building an example 'three-dimensional' model of the knowledge required to operate two popular interactive systems. This model represents a design tool which is used to perform an informal evaluation of the two interfaces.

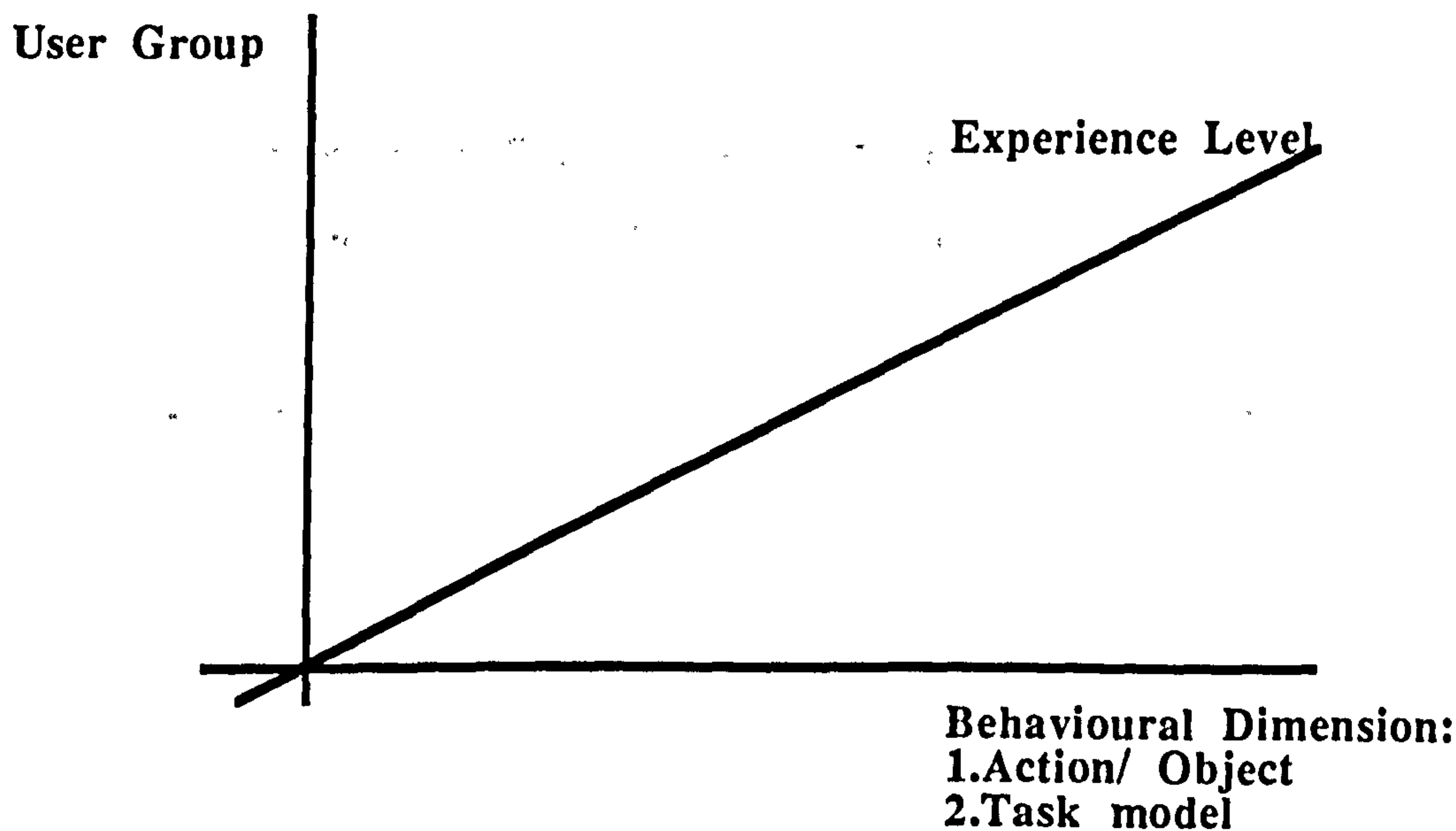
### 12.3.1 Representation of the model in the exemplar

The model was formed by recording users performing natural tasks with the two target systems. Carters (1986) classification scheme was then used to identify the interface objects and actions they employed and Johnson's (1986) techniques were used to genericise that behaviour. This provided a kernel of interface objects and actions for the whole experimental population. This kernel was subsequently divided according to a subjects:

- Experience
- Membership of a specific user group

Subsequently, further analysis of the data was undertaken to identify common strings of up to three action-object pairs and the occurrence of these pairs was again tested for significance within user and experience groups.

The three-dimensional model therefore looked like this:-



**FIGURE 9: THE EXEMPLAR 3-D MODEL**

The first example of the behavioural dimension accounts for whether the behaviour is procedural or declarative (action or object). The second example of the behavioural dimensions specifies a more complex task model based on the short strings of behaviour prevalent in each group. This dimension may in future be extended to include longer interaction sequences when these can be reliably codified.

### **12.3.2 The group model**

The group model is a model of the behaviour of user and experience groups constructed using analysis of the behaviour which they displayed in the evaluation. This model is tested in a number of ways to assess its validity.

1. Structured feedback from the users to determine whether the assumptions made during the construction of the model were correct: that is that the behaviour observed by the experimenter was that actually displayed by the user.
2. Statistical testing of the model. Three basic types of statistical test were used:
  - Tests of the significance of the occurrence of a behaviour as condition of membership of a given group (Sign Test, T-test) compared to that of the whole sample
  - Correlation between the frequency of occurrence of types of behaviour between
    - i Members of the same group
    - ii Members of different groupsto identify similar behavioural repertoires (Kendall's-t)
  - Distribution of behaviour within various groups. Regularity of occurrence of behaviour

among group members and the extent to which it characterises that group (Variance-ratio (f) test)

3. Informal evaluation of the two systems by examining the extent to which the knowledge model could support the performance of tasks.

### 12.3.3 The task model

The task model is a similar statistical model to the above, based upon short strings of task behaviour rather than simple actions and objects. This model is tested in a similar way to the above.

### 12.3.4 Robustness of the model

The three types of test described above combine to offer a range of evidence to validate the dynamic model. The feedback from the users validates the contents of the model and hence validates the data capture and analysis techniques which were used. The statistical tests validate the groups and levels of subjects which were assumed within the design. Where significant similarity between the behaviour of members of the same group can be shown then the group can be said to be valid. Finally, the value of the model as a design tool is assessed by demonstrating the type of performance prediction which it can make and comparing these predictions with the known attributes of the systems. Using the three types of test described above the contents, structure and utility of the tool can be assessed.

## 12.4 MODES OF EVALUATION

This section presents the two fundamental modes of evaluation. The empirical study looks at both types of model to identify to what extent they can contribute useful information to the design process. The type of model formed plays a role in determining the realism of any dynamic

evaluation. As has been said, the formation of dynamic models of working interfaces is made simpler due to the availability of a prototype with which users actual interaction behaviour can be assessed whilst abstract systems offer more problems.

#### **12.4.1 Abstract models**

An abstract model is a model of a virtual system which is based upon the requirements inherent in the application environment. Creating and evaluating models of the dynamic aspect of a proposed design is a useful activity to be undertaken using this tool as the dynamic model formed here would be of significant use in suggesting design options. This type of model is explored within the empirical evaluation by forming a paper simulation of a word-processing system and asking users to simulate the performance of a word-processing task.

#### **12.4.2 Working models**

A working model is a model of some dynamic aspects of a real system and is based upon observation of users interaction with that system. This type of model can show greater realism than the above type of model, however it is of less direct relevance to the design process as it requires the existence of a working system. This type of model is represented in the empirical evaluation by models of the Macintosh and UNIX systems.

### **12.5 FOCUS OF EVALUATION**

This section describes how some of the constructs described earlier in the thesis are realised in the dynamic model:

### **12.5.1 The introduction of a temporal element**

A model can be formed which attaches nominal temporal values, or ranges of values, to quantify possible combinations of generic interaction behaviour. This quantification is similar to the process which occurs in current tools (Card, Moran and Newell 1985). The model described here, however is based upon realistic generic-interaction primitives which have been derived from direct observation and which encapsulate some of the temporal aspects of real-world behaviour.

### **12.5.2 The introduction of a calibration scale**

A model can be formed to enable relative temporal modelling of generic interaction items allowing for identification of scheduling and ordering criteria as they exist in the real-world and prioritisation due to real-world strategies. This is realised in the empirical evaluation in terms of the experience levels. This is a longitudinal calibration scale.

### **12.5.3 The introduction of an interaction path**

The next level of description involves the generation of a real-world interaction path which can be modelled to outline the relevant temporal and generic aspects of an interaction sequence. This is a model of real-world behaviour with a given interface. This is realised within the evaluation as a set of short strings of behaviours. The strings of behaviour are short due to the unreliability of anything more than descriptions of 'short-burst behaviour' (Roberts and Moran 1984).

### **12.5.4 The introduction of decision points and knowledge states**

Once the general knowledge model of interaction is formed significant states and changes can be identified to provide further and more meaningful calibration. This can allow domain specific changes and states to be modelled. The knowledge states referred to here are realised in the

empirical model as the domain knowledge of the groups of subjects. Here domain dependent differences in interaction behaviour can be built onto the basic calibration scale. Decision points are not addressed within the empirical model although some evidence as to the nature of these points can be found in an experiment looking at user decision making which is described in chapter 15.

### **12.5.5 The introduction of user experience or context**

User experience can be described in terms of the specific states and changes which combine to define users in particular domains. An ongoing description of the significant constraints, assumptions and percepts associated with generic interaction behaviour in a given domain could be derived. This would further develop the set of knowledge states along other dimensions and define a further set of decision points such that an exhaustive set of feedback rules could be devised. An explicit model of all the relevant aspects of user experience can be formed and the results of all the generic interaction behaviours defined. This type of model is again not realised in the empirical study. The study of experience or context is seen as one of the areas for the development of this work and it is discussed in chapter 13.

## **12.6 THE GRAIN-OF-ANALYSIS OF THE METHOD**

This section describes how the grain-of-analysis of the empirical study is controlled to provide the required amount of detail within the model and the results. The level of analysis relevant for a particular evaluation will necessarily vary according to the requirements of the experimenter and this should be allowed for in the design of HCI evaluation tools. The tool described above possesses another distinct advantage in that dynamic modelling can be carried out on a variety of levels. This diversity of levels is represented within the current work by two types of model:

1. The user group model
2. The user task model

The first of these represents the large scale evolution of the user over the course of interaction and hence, specifies changes between novice, naive and experienced users. This level of evaluation is most likely to produce concrete results when applied over a number of interaction sessions and this number may be indeterminate if experienced users are to be modelled. The user task model represents the change within the user within one session and records fluctuations in behaviour as a result of the performance of a task within such a limited time frame. This is represented within the empirical study as chunks of 'short-burst behaviour'.

As a result the former type of evaluation facilitates analysis and representation of the changes in user group - or the discrepancies in changes between user groups - behaviour due to dynamic factors over a prolonged period. The second approach refers to changes in individual (or single group) behaviour down to the level of specific task elements over one session.

The units of evaluation in the first model are the user group models and hence the sample size, trial length and grain-of-analysis is fairly large. The units of evaluation in the second study are the task elements and behaviours and hence the sample size, trial length and grain-of-analysis can be said to be fairly small. The generic interaction primitives used for the basis of the model are self-contained and well integrated and hence they can be used to construct models of various sizes. The primitives subsume a set of contextual constraints pertaining during their capture and hence, can be said to represent the fundamental unit of behaviour. The user group model is concerned with the manipulation of large groups of such primitives and comparison of those groups, for example the total interaction kernel or a subset for a given group. The user task model is concerned with the manipulation of unitary dynamic task descriptions and the modelling of the changes of given users during the execution of these tasks.

In summary, the first type of model focuses on the dynamic variations within behaviour at an organisational level, showing variance between given users whilst the second type of model focuses



on the task-level and provides a specific analysis of dynamic changes in task behaviour within a narrower group. It is possible that the first type of model can be placed on top of the second type of model with the second model providing a finer perspective on group behaviour. It is this flexibility in terms of the level of analysis, along with the flexibility offered in the choice of media which allows this technique to be so powerful.

A number of general criteria exist to help in determining which level of approach is appropriate. If the evaluation is to account for organisational concerns then a large grain of analysis is required. Constant study at the finest level of detail would usually include much redundant detail and would often make descriptions opaque for the designers purposes. Whilst this thesis advocates a broad approach to interface evaluation via the consideration of a wide-range of material the level of analysis need not always be at a very acute level as long as all interesting events can be identified on the various levels, in order that comparisons can be made. In some cases, however an acute level of analysis will be required to identify particular local considerations which may not be clear from a general event model.

In many cases the appropriate level of analysis cannot be determined until evaluation has proceeded and results have been examined to see if various target behaviours have been captured and modelled. Hence, an evaluation may be an iterative one in which the grain is adjusted after each trial to capture the appropriate behaviour (Card, Moran and Newell 1982). The grain of analysis does not refer to level of evaluation but to the smallest entity or event that the assessment can detect. These entities or events are not bigger or smaller at various levels rather they occur in different formats.

Grain-of-analysis should be adjusted to account for the natural variation in the sampled behaviour and this can optimise reliability: this process can utilise statistical analysis to determine the required sample size to capture the required effect. Measures can be taken to capture behaviours which fall outside the normal range but this normally involves the capture of much redundant material unless a specific study is made of one behaviour. This type of phenomenon can only really be determined

during evaluation as the significant non-standard behaviours will not arise until evaluation has been attempted (Card, Moran and Newell 1982).

In conclusion, grain-of-analysis can be said to refer to an assessment tools ability to model and evaluate entities or events at given levels of resolution.

## 12.7 THE VALIDITY OF DYNAMIC EVALUATION

Following the above discussion of evaluation methods it is appropriate to extend some criteria under which the validity of the adopted techniques can be assessed. This section presents a discussion of those criteria. Some of the ideal properties of the process of the evaluation of dynamic HCI have been described and it is necessary first to examine the criteria chosen to represent dynamic HCI and second to examine the assumptions which must be met to ensure that the technique measures what it purports to measure.

It has been said that there is no exhaustive set of criteria to define the dynamic component of a HCIF and indeed part of the emphasis inherent in this paradigm is that approach from a number of perspectives is necessary to capture all relevant aspects of individual and complex application environments. It is believed, however that the criteria offered represent a sound coverage of the types of criteria which would be relevant. The set of principles selected provided the basis for the design of this experimental study. The study of the literature allowed assumptions to be made in terms of some of the constructs used - multi-level models, for example - and the useful results arising from the exemplar evaluation validates these assumptions, to a certain extent. The additional principles outlined were the subject of independent experimental study and the results produced not only displayed the existence of a dynamic effect in interaction they also provided error and regression data for use in the main evaluation. Overall the criteria outlines formed the basis for a successful experimental study and a comprehensive evaluation and facilitate the use of existing findings concerning dynamic phenomena - multi-level models, for example - in the evaluation of

HCI.

The three dimensions of the interface are described in terms of time, level and dimension. Some account is taken of the individuality of users to include in the evaluation some consideration of real-world user discrepancies and this is qualified, to some extent, by information on a given groups error and regression characteristics and particular perceptual biases. It considers both local and global effects to a greater extent than is found in the current metrics. Finally, it considers change in the user in the perspective of the other criteria and this plays a large part in describing the dynamic effect. These properties therefore represent a large proportion of the major factors which can combine to represent the dynamic aspects of interaction.

A discussion of the conditions of the application of the metric is appropriate here. With reference to the criteria utilised earlier, the content validity of this metric would refer to the metrics ability to describe a representative sample of behaviour and assess its dynamic component. So this would involve both the capture of data at a relevant level, the application of a process to extract the dynamic aspects of this sample and its evaluation in meaningful terms for the whole interface. In the exemplar evaluation the sample of behaviour captured reported at a useful and practical level on the range and variation in behaviour between users in various users groups and allowed comparisons to be made and statistically significant discrepancies to be identified. It proved possible to capture a sample of the user knowledge of interaction devices which showed some reliability of application and this was tested by comparing the range of predicted user methods against user feedback on their actual knowledge. This sample was used to draw some statistically significant conclusions in terms of various aspects of the dynamics of the user behaviour and to form interesting temporal and knowledge models of the interaction being studied. These models could not, in this case, be tested in the design of an actual interface, however they proved successful in predicting behavioural differences between the two interfaces as displayed by actual users. Content assessment in a full application of this model would involve modelling of the user from a number of different perspectives and this measure would be determined by the experimenters prudent choice of criteria to

assess a given dynamic aspect. Exhaustive treatment using a multiplicity of criteria would of course capture many of the relevant behaviours but would be a time-consuming process.

Assessing the construct validity of this metric would entail an analysis of the synthetic constructs involved in representing the user. The validity of the constructs in terms of their appropriateness for storing data to be used in a dynamic evaluation would be based on the realism and utility of the user models formed in Chapter 10. These have proved to be effective both in giving a full description of user behaviour and one which accurately predicts subjects subsequent performance. In theory the model should remain in isolation from the media chosen for the representation of the interaction as it refers to the temporal aspects of the behaviour of the user and not to its resolution in any one form. Hence, the construct should be applicable in any circumstances with some validity.

Consistent performance over a number of domains would be required to maintain the reliability of a dynamic evaluation technique. Hence the tool should be sufficiently objective to produce consistent reports on given dynamic behaviours from various domains and user populations. This should prove to be the case in the exemplar evaluation due to the objectivity of the knowledge descriptions and the calibration scales. The subjects performance on both machines is examined in objective terms with controls for user variation and discrepancies predicted on subject performance. This is a major determinant of the applicability of any tool as it must describe user behaviour reliably between domains and account for subject variation. A fuller test of reliability would be obtained by testing the performance of the tool with a wider or different sample including a different range of users. This has proved impossible within this research project, however any subsequent use of the tool would provide some useful data.

Maintaining internal consistency within a given application is a difficult process. Given symbols or processes occurring at different points of the interaction should be described in a consistent fashion. This is controlled for in the objective task description provided by the calibration scale: the generic functional and knowledge descriptions formed in the experimental study descriptions should allow

objectivity to be maintained. The statistical controls provided for user experience along with the real-world interaction behaviour displayed by the user maintained an accurate description of interaction when the task was reproduced from the generic kernel. Feedback from the user on given tasks and subtasks used allowed internal consistency to be maintained and the model actually reported with substantial accuracy performance discrepancies between user groups in the exemplar evaluation described.

The metric described seems to perform adequately on the objective criteria described above for the assessment of measurement tools and the criteria selected seem to offer a treatment of many of the types of criterion which can help to define dynamic analysis. The technique would require substantial testing in the actual implementation of systems for a full evaluation to be effected, however the exemplar evaluation does offer some support for the validity and applicability of the tool. The essence of a genuine dynamic evaluation tool can only be referred to in terms of the extent to which it categorises or describes the 'dynamic interactive experience'. That is it places a value or a symbol on what it is like to actually use a computer and allows predictions about the behaviour of users with hypothesised systems. Whilst this tool does not purport to account for such factors it offers a initial methodology on which such a tool could be built.

This chapter has presented a tool for exploring dynamic interaction and has outlined the methodological framework used within this thesis to do so. The basis of the dynamic model has also been outlined and the application of the model in design has been discussed.

## **13. THE APPLICATION OF THE THEORETICAL FRAMEWORK FOR DYNAMIC HCI**

This chapter presents a discussion of the application of the practical framework for the integration of dynamic aspects of HCI into the design of computer interfaces. The framework allows the generation of design tools for the evaluation of dynamic HCI and consists of a taxonomy of dynamic interface behaviour along with a number of principles which characterise some of those behaviours. The principles provide the basis for an empirical evaluation of the framework, which is described later, and for the assimilation of novel behaviour and phenomena which support the theory of dynamic HCI.

Appropriate areas of application for dynamic analysis are proposed and the focus of evaluation of the framework is presented in the context of given applications. This chapter suggests that the consideration of dynamic factors in the evaluation of HCI is essential, outlines how the consideration of such factors allows characterisation of task behaviour in 'real-world' interaction and shows how future developments of this model can lead to powerful predictive models of interaction behaviour.

### **13.1 A DEFINITION OF DYNAMIC HUMAN-COMPUTER INTERACTION**

Human-Computer Interaction is a process, the nature of which is determined by the relationship between attributes of the user, task, tool and computer (Van der veer et al 1985, Shackel 1986). The interface is essentially a notional entity which is the product of the interaction between two or more parties and is inherently related to and proscribed by the attributes of those parties. Furthermore, the interface does not exist until an individual user and a given computer are engaged in interaction, therefore the nature of that interface can be deduced by analysing the interaction between the

attributes of user and computer (Watkinson 1986). Rationalisation of attribute combinations which constitute the interface is, however, necessary if interaction is to be studied meaningfully as the possible number of relevant attributes is enormous. As a result dynamic human-computer interaction can be said to be the process of the interaction between user and machine attributes and the mediation of that process by task and environmental factors.

The design tool described in the previous chapter aims to provide a framework for the modelling, evaluation and statistical control of the interaction between interface attributes and for categorising these combinations according to their usage. The remainder of this chapter discusses some of the issues which the *theoretical* framework of dynamic interaction would hope to address and also discusses the concept of dynamic HCI and its viability.

## 13.2 PRINCIPLES OF DYNAMIC HUMAN-COMPUTER INTERACTION

The theoretical framework for the analysis of dynamic HCI consists of two main components:

- The taxonomy of dynamic behaviours
- A set of principles for the instantiation of those behaviours

The taxonomy of dynamic behaviours is discussed in Chapter 11 however, the successful addressing of the dynamics of HCI within design relies on the reduction of the problem-space generated by complex interactive systems into a set of concrete, testable principles or hypotheses. This section presents such a set.

It would be misleading to say that such an exercise could be undertaken without compromising the integrity of the dynamic interface as an holistic and integrated entity. It would also be adventurous to claim that the full richness of the interactive process could be expressed in a limited number of principles. Indeed, the author would claim that there exists a vast number of relevant dimensions,

the identification of which remains an ongoing research task.

The author has, however identified a number of salient dynamic properties of HCI both from examples in the literature and via anecdotal experience. These represent behaviour or attributes symptomatic of dynamic systems which would be evident were HCI to be considered such a system. Whilst none of these attributes on their own denote a dynamic effect their summation would offer a practical basis on which to determine the validity of the dynamic evaluation paradigm. This is by no means an exhaustive set and, indeed, the very principle of this research paradigm is that no such set could be practically identified within this work: any relevant criterion inherent in an application environment needs to be accounted for. These criteria do, however, offer a practical theoretical structure on which to base the experimental phase of this work which accounts for the selected categories of dynamic criteria in HCI. These criteria are resolved as a set of assumptions concerning the behaviour of users and interactive systems, which state that:-

1. Changes occur in user, machine and task state during and between interaction with respect to motivation, goals and memory
2. Changes in user state may be due to interactive effects which arise as part of ongoing or previous interaction
3. Interaction occurs in real-time, subject to user inference, experience and task context
4. Interaction involves individual users and machines which differ significantly
5. Interaction is a multi-level process in which users address task issues at a number of levels.
6. Interaction is a multi-dimensional process



7. Interaction has a heuristic or pragmatic basis in which users are involved in utilitarian decision-making
8. Interaction possesses holistic properties
9. The perception of the interface is an active process
10. Error behaviour is natural and error-handling is a dynamic process
11. Users regress, in terms the loss of items from their interaction model, during and between interaction

The selected principles form the basis for the experimental work described in chapter 15. These are the principles which are not well supported by the current literature.

### **13.3 ATTRIBUTES OF DYNAMIC HUMAN COMPUTER INTERACTION**

In addition to the fundamental principles of dynamic HCI expressed above a number of characteristic and salient features of dynamic interactive computer systems are offered which scope the problem area for future research. This section presents a brief recap of some of the categories in the taxonomy.

#### **1. They are open systems**

If an interactive computer system is to be useful then the user must have control over the state of the machine as the user represents the open part of the system (Edmonds 1986). Hence, the action, cognitive and knowledge repertoires of the user should be accounted for in design.

## **2. The evaluation of dynamic systems requires a ecological and ethological approach**

The user forms a part of their environment and that environment can exert a major influence on the nature of interaction (Shackel 1986). Influences from the environment can play a role in the dynamic interaction process and hence the interaction between user and machine attributes, and environmental attributes needs to be studied (Van der veer et al 1985).

## **3. It is interested in the interaction between the user and the environment -**

### **Physical context**

One particular form of interaction which dynamic evaluation seeks to address is the users physical interaction with both the machine and the environment. Modern computer systems involve physical action as a large part of the interaction metaphor ('mousing' and handling of paper, for example) and these factors are difficult to describe in a static context (Buxton 1986).

## **4. It is interested in the way users apply existing knowledge in the use of computers - Cognitive context**

Users of one type of computer system may apply cognitive and physical principles in the operation of novel equipment: and this manifests itself as transfer of training between computer systems. Dynamic evaluation would seek to codify and rationalise the processes by which users store and employ knowledge about the dynamics of the interaction process and apply this within design. This is the subject of experimental work described later in the thesis (Study 1 - Chapter 15).

## **5. Dynamic HCI displays 'synergy'**

The combination of human and computer interacting can clearly produce better results than the

human or computer could on their own in many cases. Work has been done in classifying the tasks at which humans and computers can excel individually (Fitts 1986), dynamic evaluation would look at what the combination of the two were good at and how the dynamic interactive process supports those particular tasks. This involves the study of parallel, co-operative problem-solving between user and computer. This knowledge can again be used in the design of computer based tasks.

#### **6. The interface can be seen as being holistic**

Dynamic evaluation would seek to develop measures for the usability and effectiveness of the interface which are not based upon 'reductionist' paradigms (Malone 1982). Attitude measures are one of the few techniques which evaluate the interface as a whole. Dynamic evaluation would aim to calibrate the users more general percepts of the machine in the task and environment context and assess their accuracy and relevance for the task, and in the uptake of new computer systems.

#### **7. There is a high active and interactive component**

Interactive computer systems are typically highly active and interactive in their operation. The benefit which is derived from these systems is in that users can display natural active behaviour similar to that they would use in the real-world and in the rapid exchange of information in a configurable, structured format (Buxton 1986). This takes advantage of both natural interaction metaphors and well structured tasks. Dynamic models assist in the development of metaphors and the design of useful interaction strategies.

#### **8. Dynamic HCI involves a rich interplay and manipulation of large amounts of information in complex types**

Systems which require the handling of large amounts of information also require special strategies for information categorisation and retrieval and for the prevention of cognitive overload

(Shneiderman 1986). Dynamic models are useful here as they allow users to adopt flexible classification strategies for information and they allow the monitoring of the ongoing information load in real-time in response to a combination of task-factors.

Presentation and transfer of large amounts of data are also key dynamic topics: highlighting of key aspects of large bodies of information relies upon the active perception of the user whilst data transfer relies on dynamic knowledge of the formatting requirements of both the transmitting and receiving parties (Innocent 1984).

### **9. Dynamic HCI involves the manipulation of complex esoteric or domain related concepts**

One of the areas in which dynamic factors play a key role is that of professional or domain-specific systems. This is because such systems often rely upon some existing domain knowledge being held by the user (Self 1984). The application of this knowledge in the performance of tasks and hence the design of those tasks is a dynamic issue. The interaction between the task structure and the users existing knowledge, and the changing nature of the users knowledge base are all issues which can be addressed using dynamic modelling.

### **10. Dynamic systems exhibit behaviour which can only explained by 'deep' or pragmatic reasoning**

Users typically will exhibit some behaviour which often seems unexplainable in the current task context. For example a user may persist with a practice developed when using another machine which is irrelevant in a novel task context. This behaviour can only be explained and accounted for in design by developing 'deep' models which take context into account (Barnard 1988). Such models need to examine the role that the context plays in determining how the user performing the task and this is a dynamic relationship which can vary in each example.

### **11. Dynamic interaction involves the interaction between individual computer and user attributes**

One of the areas of dynamic interaction which has received some attention (Van der Veer et al 1985) is the interaction between users and computer attributes during interaction. An example of this is how experienced users perceive extended system response times: the attribute of experience interacts with that of system delay to cause frustrations. This is also addressed by the illustrative design tool developed within this thesis.

### **12. Dynamic systems can be interrupt - driven**

Realistic models of HCI need to account for those random events which can interrupt interaction and which can effect the users later operation of a system (Cypher 1986). Most of the activity in our lives is interrupt driven. For instance a user might be interrupted in the creation of a document by the arrival of a mail message. The system needs to allow the user to leave the task, read the message and return to the task with as little effort as possible. Hence a model is needed of those aspects of the task which must be suspended and re-initiated when the user returns to the task. This is the subject of novel experimental work described later in the thesis (Study 4 - Chapter15).

### **13. Dynamic systems can be subject to hard time constraints**

A number of systems are subject to very strict time constraints on the behaviour of the user. These are typically 'life-critical' systems such as defence systems or civilian air traffic control or those which involve taking 'risky' business decisions concerning large assets. In both types of system a temporal or dynamic model of the tasks must be available for the designer as the time constraints may represent a key design goal. Information access time and system delays all need to be accounted for in the model to ensure performance of the system to the user requirements.

#### **14. Dynamic HCI is interested in what users are actually doing and want to do**

Finally, and perhaps most interestingly, there is a need for realistic action descriptions of users both in terms of how they perform current tasks and how they would want to perform future tasks; this would include specifying what the user wishes to gain from interaction. Using formal methods to describe how users manipulate data structures or primitive task-analysis techniques to specify physical actions or procedural task segments does not tell the designer what they want to know. Informing designers about what users do and want to do with computer systems is the principal purpose of the study of HCI (Johnson 1986). This would be more readily achieved with design tools which take into account context, user and machine attributes and specific task requirements within design.

##### **13.3.1 The generation of novel design tools**

Chapter 11 describes the requirements which a novel design tool must meet if it is to successfully address any aspect of dynamic HCI. It would be useful, however to show how the principles shown above can be used in conjunction with some of the attributes of dynamic HCI to offer a practical means for generating potential design tools.

The above discussion of the dynamic attributes of interfaces cites the interface as an 'open system'.

This can be translated as the users ability to:-

- Affect the state of the system in a novel way
- Bring their knowledge to bear in the use of the system
- Configure the system to their own requirements

These requirements upon the system can be the subject of testing to ensure that a computer offers an

effective dynamic interface. Firstly, the ability of the user to affect the system in a novel way should be examined. This involves the specification of system states such that an evaluation of any novel state reveals that the machine is not functionally or informatically equivalent to its previous state. This could lead to the development of a grammar which would be sensitive to novel operations by the user. By testing the logical equivalence of the users range input with the legal syntax of the machine it would be possible to assess the flexibility, or 'openness' of the system in accepting user inputs.

Another possible design tool would be to test the way in which a user might bring existing knowledge to the system. A notation would be required for describing the users existing relevant knowledge and evaluating how this might interact with the current system attributes. Hence, a knowledge and functional representation tool are required along with a mechanism for rationalising how the two might interact.

Thirdly, the ability of the user to configure the system to their own requirements could be tested by forming a requirements model for the user to describe their configuration and then examining how far the user can go towards meeting those requirements. This requires a sound method for expressing requirements and an experimental method and set of measures of 'reasonable effort' to assess the systems ease-of configurability.

These tools are derived by taking the fundamental set of requirements which the system would need to possess to facilitate a particular type of dynamic behaviour, as specified by the principles, and recruiting tools for the measurements of systems and user performance.

### **13.3.2 Rationale behind the development of the selected design tool**

The example design tool described in chapter 14 is one which involves the modelling of user behaviour with TKS-like (Johnson 1986) primitives and the statistical analysis of those primitives

using correlation and variance tests. This particular design tool was selected for development as it demonstrates one of the key principles concerned with dynamic interaction: the propensity of users to change as a result of interaction. The provision of a design tool which can statistically model the difference in users behaviour according to experience and group membership was seen to be directly applicable in the development of interactive systems. The tool allows designers to identify which attributes characterise particular user groups and account for these in the design process.

The choice of this design tool was also partly determined by the scope of the project: the ability to construct a tool from established techniques brought it more within the range of the Phd work than developing a tool from first principles. Time-taken for the performance of the exemplar study amounted to some four weeks involved in obtaining subjects, running trials and gaining feedback. The time-taken to analyse all the results and form the generic models amounted five and a half-weeks. This would seem to make this a useful developmental tool within the timescale of an actual research project. Rapid iteration is necessary in most projects and the ability to vary the sample size helps to allow this whilst the validity of the model is maintained. The example evaluation in this study managed to produce a useful model on a rather small sample.

In retrospect it may have been expedient to explore a number of design tools on a much shallower level. The depth and rigour of the analysis does not necessarily lend support to the validity of the tools during design.

#### **13.4 APPROPRIATE AREAS FOR DYNAMIC EVALUATION**

All of HCI involves the display of dynamic behaviour, to some extent, as it inevitably involves learning of the interface configuration, for instance. Some types of system, however, require the user to exploit the dynamic aspects of interactive behaviour more readily than others and it is these systems towards which this design technique is particularly targetted.



The first example of these types of system are those systems which involve the manipulation of professional or domain specific knowledge to any extent. In such conditions a known domain or a set of specific conceptual constraints can be identified and particular events on the task path can be attributed to given types of mental activity. Second, systems which require creative, specialist or novel behaviour can benefit from dynamic analysis as they involve task descriptions which may not be describable in other terms. Third, applications which involve a hard or soft time constraints are appropriate as the explicit representation of human temporal factors enables them to be optimised in the final design. Fourth, applications which lend themselves to adaptive implementation would also benefit from the temporal description as the adaptive sequence could be formally specified in temporal sense. Fifth, applications which need to communicate with humans at a level of performance approaching that of intra-human communication can be designed by exploiting the actual communication strategies exhibited by users. Finally, applications which are particularly prone to erroneous behaviour or where such behaviour can be specified well in a formal sense can be described using this type of technique as it can account for pragmatic issues and can incorporate specific empirical error-data associated with a given user group performing a given task.

A comprehensive, dynamic evaluation can only usually be justified where the benefit to be gained in terms of the identification and modelling of dynamic considerations is sufficient to mitigate the detailed level of analysis required. This could be costly both in terms of time and resources and the design of deterministic systems where interaction is driven totally by the machine may not be influenced by the dynamic behaviour which can be exhibited by users.

Particular examples of deterministic systems which would be inappropriate include static menu driven systems or form filling interfaces or those systems which offer little user choice in the flexibility of the dialogue or offer very short or simplistic dialogues. The tool offered could, however, contribute something in terms of a description of the temporal aspects of any system. In the design of information systems, for example, the tool could support the designer in specifying the optimum ordering of interactive operations and the relative information demands at given points.

The area where this technique shows no scope for application is in the implementation of systems which have no interactive component and where the above issues determining dynamic interaction do not apply. It is very rare, however, to find a system which has no interactive component at all as the interface usually involves providing feedback to the user. If such feedback is minimal, however, it may not offer scope for increased performance due to dynamic factors.

Some examples of applications which demonstrate the above behaviours, and hence would benefit particularly from dynamic modelling, are outlined below:

- a. Management Information systems
- b. Real-time control systems
- c. Professional, specialist or domain specific systems
- d. Integrated organisational computing systems

The first example here involves the 'manipulation of large of amounts of esoteric information' and hence the development of this type of system would benefit from modelling tools which can account for pragmatic and esoteric types of information and present them in way which can make them accessible to users.

Real-time control systems present one of the most clear cases for the modelling of dynamic HCI as the constraints which are applied to the users performance - for instance Air Traffic Control - could be directly controlled for and anticipated in a dynamic model of interaction. This would present a very real benefit in predicting performance with these systems.

Professional systems again contain a large amount of specific knowledge and practice which could be best codified with a dynamic model of knowledge and interaction which takes into account users experience and preferences for interaction styles.

Finally, integrated organisational computing systems would benefit greatly from the group modelling tool developed within this work to account for the pool of experience and help rationalise the requirements of a number of user groups. The use of this technique in CSCW applications is to be the subject of a future paper and this is described more fully in Chapter 16.

#### **13.4.1 Potential benefits of the evaluation of dynamic Human-Computer Interaction**

The focus within dynamic evaluation is on real-world situations such as those that would occur during interaction and this technique seeks to model them in a realistic fashion. Dynamic methodologies state user requirements in meaningful and realistic terms, that is as any type of attribute which the user might need to display to be able to carry out a given task. In essence, dynamic evaluation aims to specify realistically the complex interplay between user and machine which combines to determine interaction and to describe the methods, flows of information and strategies which define human-computer interaction.

This technique also aims to allow the designer to impose controls over the interaction of all the salient aspects of machine and user, which occur in given interactive environments, and rationalise them in the design process. Modelling of interaction through a variety of media and the user groups which inhabit a given environment enables the designer to characterise the interactive behaviour of given environment (Shafer 1988) and capture the local usage patterns which form a salient model of that environment. Designing interfaces to exploit the users repertoire in terms of a the full range of Human Computer Communication skills (as expressed in terms of the types of interaction attribute)

and to reflect the preferred styles of interaction of the known user group allows interactive systems to be customised to an actual interactive situation. This thesis indicates how the dynamics of behaviour can be accounted for in the design process by identifying the users assumptions or knowledge which allow 'deep', pragmatic or implicit dynamic issues to be explained and controlled for. This evaluation is effected largely by comprehensive and complete modelling from many perspectives, dimensions and levels to combine to form a diagnostic model which can isolate and describe specific interactive phenomena. This enables the designer to attempt to predict some of the relevant factors associated with users operating a computer in a given environment. Dynamic evaluation formally describes the type of 'predictive empathy' displayed by ingenious examples of interface design which perceptively exploit known factors or good predictions of human behaviour (Reisner 1984).

This type of technique allows a designer to separate the relevant, desirable, salient, definitive, important or interesting aspects of the successful execution of a task due to the intelligence or style of users and the adaptivity of behaviour to computers and other devices. The framework would also allow the designer to exploit the power of computers and their combination with other computers and users, and exploit what each of them is best at, by characterising the mass of information or knowledge flow, in the design of future interfaces.

Dynamic evaluation also allows the designer to identify, model and implement human problem-solving strategies derived from real-world behaviour or those which are compatible with human faculties or behaviour (Newell and Simon 1962) and it allows the designer to 'empathise' with the user and find out what they need or is important to them at a given juncture.

Finally, dynamic evaluation allows the designer to specify what is the essence or quintessence of successful, stylish HCI and store it for future use in the design process. The above features are largely possible because this technique focuses on capturing and representing attributes of actual human behaviour and strategies rather than simply describing human cognitive attributes. Edmonds

(1988) suggests that the study of *ease-of-use* of computers may be defunct and that study should be concerned with *ease-of-learning* or *ease-of understanding*. This statement may be misleading as whilst the study of the latter two phenomena is much less problematical more direct 'user targeted' benefits are more likely to arise from the former. This may be due to the fact that systems which are successful elementally do not necessarily work holistically or operationally.

### 13.5 ASSUMPTIONS NECESSARY FOR THE EVALUATION OF DYNAMIC HUMAN-COMPUTER INTERACTION

Comprehensive and complete evaluation relies on a realistic and accurate model of the user to account for all relevant behavioural and cognitive phenomena in the design process. Before an evaluation can be performed fundamental assumptions must be made about the nature of the user and interaction. For example, it is usually necessary to assume that human behaviour is purposeful otherwise it is difficult to represent interaction (Moran 1984). A great deal of work has been carried out to establish whether human behaviour is purposeful or whether it is driven solely by environmental factors, however it is believed that very little of human interaction with computers is truly without purpose, rather that many behaviours have an implicit purpose which is satisfied during the process of interaction (Malone 1982). Such descriptions are largely meaningless unless human goals can be identified and the behaviour and actions of the user related to them as it is otherwise impossible to hypothesise about the human rationale behind HCI: this is one of the target features of this work.

The very act of using a computer affects subsequent performance with that interface and this is expressed in terms of what the user learns about the interface (Carroll 1982). Current techniques, in attempting to produce application or task independent descriptions, have largely ignored or failed to address the recursive effect that interaction can have upon itself or the changes in the user of interactive state which occur during and due to interaction. These descriptions therefore do not account for interaction as it is experienced by the user throughout the course of interaction. These

factors are typically identified in current studies by evaluating and assessing a prototype and controlling for them in the re-build phase however they represent aspects of the system not captured in the formal analysis. Prototypes are tested by users with a range of experience levels to account for changes in interaction requirements rather than identifying and describing these changes in formal manner and designing an interface to meet the changes in the user. This can only be facilitated with the aid of some sort of interim representation of the interaction process, and the dynamic factors which need to be considered, before the working prototype is built.

The interface can be seen as arising from the combination of user and machine attributes over time and hence it fluctuates as those attributes fluctuate:

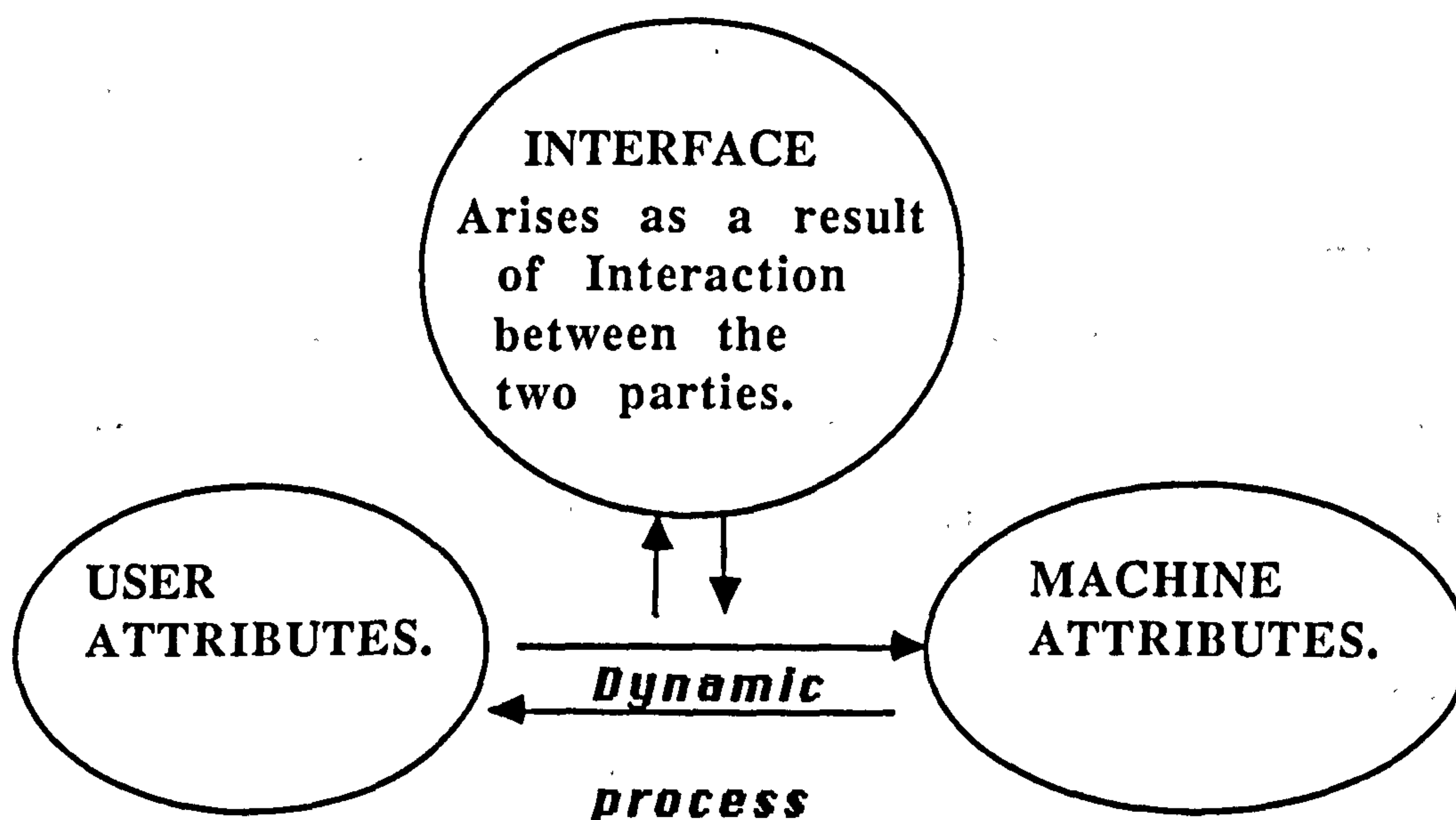


Figure 10 The interface as the product of dynamic HCI

Effectively, there exists a different working interface at each stage in interaction as different attributes are employed or discarded. Recursive processes, such as learning, which occur over time are derived from the inherent states of users and machines and if these states can be realistically identified the interactive effects and changes can be modelled and accounted for within the task framework.

The properties of interfaces which enable them to exhibit such recursive and flexible behaviour can be categorised within the term 'dynamic'. The term dynamic embraces the changing state of the user and interaction state which results from the causal effect that the exchange of information, acquisition of knowledge and perception of the interface (and the delays involved therein) exert on each of the interactive parties. This effect derives from the flexibility of the human information processing system in coping with different situations in the world, to some extent, and describes the power behind the human communication facilities. The human is able to focus on the interesting and task-relevant aspects of the interface, at a given time, and to utilise and learn from this experience in future interaction. The inter-human communication process is a good analogy for this and this factor has been exploited in a number of experimental studies (Evans 1974, Chapanis 1983).

It would seem an attractive prospect to be able to describe or account for some of these dynamic factors, where relevant, in the design of interactive systems. It has been difficult to identify a concrete research strategy, to date, to allow workers to describe the changing user. A frame of reference is required against which discrete changes in given user or interaction attributes can be evaluated. Since change in the user occurs over time and no factor in the interface can be assumed to be constant for any given period a realistic temporal or task-path framework may offer the only calibration scale against which change could be assessed. With sufficient empirical study the types of unit or scale actually employed by the user to provide a stable and objective view of interaction could be identified and this would act as an appropriate frame of reference for storing changes: this thesis describes a method by which such a scale could be identified.

No predefined set of criteria exists which can be identified or specified to provide a set of metrics or measurables which would effect a comprehensive dynamic analysis. Relevant criteria may vary according to tendencies inherent in the application, user or task domain which determine the dynamics of interaction. Hence, any criteria which can be identified to be pertinent to interaction in a

given situation should be modelled. Dynamic factors do not exist in their own right, they merely form an alternative perspective on the criteria to be modelled. Accordingly, dynamic HCI is merely a convenient vehicle for expressing and highlighting those issues particular to a given interaction which would facilitate a realistic and meaningful evaluation of humans behaviour with computers.

### 13.5.1 The viability of dynamic evaluation

It could be supposed that the number of salient factors relevant to users concerning interaction grows exponentially as interaction time increases linearly. As various factors are introduced into the interaction path the body of relevant information can potentially increase in size by a further power. In actuality, however, it is believed that the amount of information relevant to the user remains relatively constant as certain information becomes irrelevant at the same rate that new information becomes relevant (Christie 1985). Human beings have only a limited capacity for handling and storing information deemed to be relevant or important at one particular time which is related to the buffer or processing size of the short term or working memory (Miller 1957). Hence, this limited set is amenable to representation using a formal description technique.

Users overcome memory limitations by applying existing knowledge that they possess of similar situations or contexts (Baddeley 1986) or they may condense the information already encountered during an interaction into a more compact and accessible form in a process known as 'chunking' (Wicklegren 1953) to make room for further input. These strategies, however, involve a certain number of assumptions that may not necessarily apply in any given interaction path and which might not affect the total capacity of the model in the working memory. Information is simply stored in meta-level, or more information rich, units rather than a larger number of units (Newell 1965). Interfaces often do not respect the absolute level of information storage capacity in the Short Term Memory and despite the most ingenious strategies as described above some relevant information is inevitably lost due to inaccurate assumptions: these limits must be respected if the system is to be used to its full potential and an accurate model of the ongoing state of the users information or



knowledge model is necessary to match the interface with the users requirements.

The features which would seem to be of most relevance to the interface designer are thus not necessarily the universal strategies for information processing at any one time (these are useful but they should be used as a normative constant) but the particular strategies that define a users interaction in a given context, such as the prioritisation and scheduling strategies employed by the user. These strategies enable the user to identify relevant information items and therefore manipulate only one type of information at one particular time. It is this process of identifying relevant items of information or knowledge which forms a vital part of the analysis of a set of users or an interaction context. These areas can only be identified by the capture and modelling of actual behaviour in a task and user context as the possible strategies for prioritisation are innumerable and diverse. Capture of the implicit strategies employed by users enables important behaviours, information or knowledge to be demonstrated in practice and reflected in the design of interfaces. Tools should exist for modelling this aspect of interaction and this thesis aims to offer such tools.

A genuine dynamic approach embraces all the diversity and complexity which is spawned from real-time studies and all the vagaries of multi-party, complex interaction, however this strict analytical approach may be less fruitful than a less rigorous but more structured approach. The latter approach would require a greater number of assumptions on the part of the experimenter, however these will almost certainly be essential if a concrete and focussed evaluation is to take place and would be no greater than are currently assumed. The 'approximate' model (Barnard 1988) of interactive behaviour proposed in this thesis is not a rigorous mathematical model as this type of model cannot accurately describe the flexibility inherent in human behaviour rather it is a model which allows the specification of a given level of analysis of interaction which generates useful models of behaviour for the purposes of designing interactive systems: for example, interaction can be modelled as a series of stable states which approximate to the state of the user at a given point.

The experimenter needs to identify the points in interaction where a given user or user groups interaction strategy or knowledge of the interaction devices and procedures changes to a significant extent and this may be indicated by the use of novel behaviours or an alteration in the general interaction style. This could be termed an evolution point in terms of the users interactive behaviour. Discrete states which have been identified in learning both computer and natural languages are subject to concrete, sudden and significant changes (Kennedy 1979): these are the points where interaction should be modelled, as the changes in behaviour identified will allow designers to make predictions about future behaviour. Interaction strategy and behaviour is, to some extent, ongoing and continuously evolving, however some of these changes are not always significant to the designer at any useful grain-of-analysis as they do not represent changes to the approximate models of the user which are sufficient for the purposes of interface design (Barnard 1988).

Hence, part of the process of evaluation involves the identification of an appropriate grain-of-analysis which retains the salient qualities of interaction but allows the experimenter to represent interaction in an economical and coherent form. This topic is discussed more fully later in this chapter however a useful example would be that evolution points may be identified in the real-world by a conflict between important goals which require a re-organisation of knowledge to make a decision on which goal to prioritise (goal-level). These would be identified by means of the techniques described in the previous chapter.

### 13.6 THE THIRD DIMENSION

Many current techniques describe interaction in terms of a number of levels (Moran 1984 et al.) and a number of techniques address interaction in terms of a number of dimensions (Johnson 1984), however these models exist in a type of atemporal state and it is the temporal discrepancies between the occurrence of behaviours which would allow experimenters to elucidate users strategies for handling these parallel activities. Dynamic interaction proposes a 'third' and additional temporal

dimension to user activity and proposes to bring the three-levels of analysis together into a single model based around the performance of tasks.

The work described here offers two types of three-dimensional model which operate at :-

1. The Task level
2. The User Group level

The first type of model refers to an individual users performance of a task and models the execution of task elements during interaction on a number of levels and on a number of dimensions within those levels, in the context of the users behaviour at a particular point in the task-path. The last contextual aspect represents the third, or temporal dimension and this aspect is modelled by the capture of the users chosen generic interaction primitives, at different levels of experience, and the order in which they are presented.

The second type of model refers to a user groups performance of a task and models the execution of task elements during interaction according to the users membership of a user group and the users level of experience within that group, in the context of the users position on the task-path. This model allows the identification of tasks and sub-tasks which are related to that user group. This model should not be confused with the total user kernel which expresses the user repertoire but does not record methods used by user groups.

The model described above is realised in terms of strings of generic interaction primitives which can be associated with a given user, or group, during the performance of a given task and these are placed within the context of the phases described in the approximate task-path framework outlined in chapter 9. Existing work indicates that predictive models are only accurate over "short bursts of behaviour" (Roberts and Moran 1984) and this is the level of analysis utilised in the experimental chapters described later. The interaction primitives captured have sufficient integrity and are sufficiently self-contained such that the strings of behaviour can be introduced into the approximate

task-path framework which is appropriate for the type of task being performed and provide a model of the users behaviour at a given phase in the task-path. This process is described in later chapters.

Chapanis (1984) states that "most metrics use ordinal scales, that is, comparative statements of better or worse, and do not make point-to-point elemental comparisons of quality". Such comparisons can be made within the basis of an objective temporal framework and offer the designer more realistic, relevant and meaningful evaluation data which would refer to common and comparable task activity.

### 13.7 THE DYNAMIC INTERACTIVE EXPERIENCE

The true value of a genuine dynamic evaluation tool can only be assessed in terms of its ability to categorise or describe the 'dynamic interactive experience'. That is it places a value or a symbol on what it is like to actually use a computer and allows predictions about the behaviour of users with hypothesised systems. Whilst this tool does not purport to account for such factors it offers a initial framework on which such a tool could be built.

A truly dynamic evaluation tool would allow assessment of interaction from the perspective of a given user or user group and would allow characterisation of their 'dynamic interactive experience'.

This section discusses some of the potential developments of this technique to model interaction as the user experiences it. A tool would be effective in this sense if the actual mental states of the user could be identified and simulated at a given point in interaction and the resultant behaviour predicted from its combination with a given interface and task structure.

This extension to the proposed type of model would be implemented by modelling specific local and global contextual constraints at particular temporal locations and their effect upon behaviour at a given point rather than simply by modelling the knowledge requirements of the user. This would be a 'context' like model. The assumptions behind the activity at particular decision points can then be

elucidated as the context of assumptions will be known. The notion of 'context' seems to embrace many of the qualities which define dynamic interaction: the concept of a constantly fluctuating set of assumptions and conditions related to the users experience, knowledge and to the particular environmental conditions in which the user is operating seems to mirror the some of dynamic changes seen in behaviour which seem to have no explicit cause. Expressing such experience and assumptions and identifying, describing and accounting for only relevant aspects of the environmental conditions would be a difficult process. It is possible that all these factors interact to bias the user towards particular local constraints.

The users assumptions and experience in conjunction with inherent, perceived characteristics of the system, cause them to regard given aspects of the system to be of particular relevance at one time and hence this perceived relevance is what determines context at any one time. A constraint model formed for a user at a given point in interaction would assist in identifying the processes with which a user determined relevance. Such constraints can be realised in the form of different interaction media and their dimensions as is described within this thesis, however this remains an approximate rather than a realistic model. These constraints would be associated with and derived from the users experience of interaction and would define the users assumptions as to the perceived result of a given interaction behaviour. Hence these assumptions could be simulated and very realistic predictive model could be formed. Experience needs to be formally described, however, to implement this type of model and this could be based on a more comprehensive kernel of temporal generic interaction behaviours such as the definitive behaviours which characterise the performance of a given task by a particular type of user of known experience.

The form of these contextual constraints will be variable and they will be difficult to define, however they may be similar to the following example. When a LISP programmer is in a LISP environment and can be said to be writing a LISP program then the term 'evaluate' can be taken to refer to an operation or entity which resembles the predefined LISP function of which the programmer is aware. This would display the set of assumptions which were pertinent at specific points on the

interaction scale, in the timescale of the discrete states displayed by the user. So, given a specific task a specific type of user including his experience of the task, equipment and domain and a pre-determined calibration scale a designer may be able to determine how the user would perceive a particular type of symbol or operation and hence make an opportune choice of symbol for the circumstances.

So, in effect, it would be necessary to know the effect a symbol  $x$  would have upon a users behaviour, under constraints  $a, b$  and  $c$ , at time  $t$ ..... This can be seen as attempting to read a users mind at a particular point in the future. This process presents many difficulties, however in the structured environment of the execution of a computer task and with the limited set of practical responses to the known interface configurations, then a useful predictive assumption can be made for a given type of user. Green's (1988) work would provide useful results here: the use of analogy theory to determine how users perceive displays may provide some general guidelines as to the processes engaged in by the user in the interpretation of symbols associated with an interface.

This is closer to the true dynamic experience: it is the change in the underlying conditions which determine or affect behaviour, such as assumptions, or implicit and explicit device knowledge which are of interest to the designer, not the explicit manifestation of the behaviour itself - although the model must be capable of generating real-world behaviour. Temporal considerations can have a profound effect on the users perceived meaning of symbols and the relationships between those symbols. It is evident from this that the experienced users conception of a system object is different to that of a naive user: tracing the evolution of this perceived image would teach us about the changes in perception of computer objects due to learning and identify some of the underlying assumptions which govern HCI behaviour due to learning. This can be termed 'true' dynamic evaluation as it allows us to deduce the ongoing state of the users interaction model with the machine at every point in interaction.

This technique still does not describe the dynamic interactive experience in its fullest sense,

however it would offer a more realistic and accurate model of user state than the model expressed in this thesis and offers the prospect of forming more relevant and appropriate predictive assumptions.

A novel technology which demonstrates the potential for the prototyping of the 'dynamic interactive experience' is that range of system which purports to offer the capability to simulate 'Virtual Reality'. By developing a testbed for interfaces based on the simulation of a computer and its interaction environment it would be possible to control for all aspects of that interaction environment. In addition, the flexibility which is offered in the simulation of interfaces is practically infinite and would allow the simulation of any interface stimulus.

The prototyping of this type of experience is all the more important where multi-media systems are concerned. This type of system aims to take advantage of the full bandwidth of the human communication channels to simulate interaction in a real-world setting. Usability is optimised by the use of the full range of communications channels and the analogy of the multi-media task to that of the real-world. Multi-media systems offer a new type of interactive experience much like the one discussed here. Here, the performance of tasks is much more of an exercise of the senses rather than the mechanistic reproduction of a paper-based task. The development of such systems has been possible due, perhaps in part, to the conception of HCI as a dynamic experience and this should encourage designers to follow similar methods.

This section remains speculative, however it outlines the potential extensions to this technique and suggests how they may be built on the work described in this thesis. If the full benefits of this work are to be eventually realised then this is the type of model that designers should look to form.

This chapter has presented a definition of dynamic evaluation and in terms of a number of principles of dynamic HCI which form the basis of the experimental studies described later. Some further attributes for dynamic HCI are offered to scope the area for further study. Some of the theoretical background behind the study is presented showing how this brings the area into the realm of

empirical analysis. Finally, some of the appropriate areas for the application of the technique are outlined along with the benefits it can realise.



**SECTION THREE: ORIGINAL EMPIRICAL WORK**

## **14. AN EXEMPLAR EVALUATION**

This chapter describes an evaluation of the design tool discussed in earlier chapters. This example is intended to display the validity of the methodology as a practical design tool. An additional purpose behind this chapter is to offer evidence to show the usefulness of the concept of the dynamics of HCI.

### **14.1 THE SCOPE AND RELEVANCE OF THE EVALUATION TECHNIQUE**

The ideal test of the relevance of any design tool is its employment in the design of an interactive system to determine the level to which it assists in the design process. This test is most valid if two groups of designers can be studied in the activity of designing the same system, one group equipped with the said design tool. The qualitative difference in terms of the assessment of user performance between the two systems indicates the value of the tool. This method of tool evaluation has proved to be inappropriate for the design tool suggested in this thesis due to the lack of availability of a suitable project which would bear results within the required timescale. Also the cost in terms of research time was deemed prohibitive within the framework of a Phd. project, however this is a future objective for this work in order to optimise the external validity of the study. Existing metrics (Card, Moran and Newell's, for example) have typically been empirically assessed away from a concrete project with additional support of this nature sought from later studies (Roberts and Moran 1984).

It is difficult to offer formal proofs of the utility of a design tool from an objective perspective, however desired performance characteristics of the tool can be enunciated and an attempt made to see if the tool actually meets these criteria. Obviously, some of these criteria are set by the

experimenter in advance of a particular evaluation and the level of analysis pre-determined so it is difficult to identify a set of *absolute* performance criteria. The criteria relevant for this tool are outlined at the end of chapter 11. It is hoped that the example offered displays the type of potential results which can be generated and indicates some of the range of analysis which is available. A discussion of the overall reliability and validity of the test will hopefully offer some further perspective on the performance of the test.

If a design tool is to possess some range of applicability across a range of activity it should not be constrained to a given type of implementation or type of activity. Although some specialisation may be necessary so that the tool can comment at some level or relevance on a particular application where this leads to stringent constraints on the possible domain of application the range of the tool may be reduced. Design tools should be independent of media considerations and should be able to analyse interaction in objective terms irrespective of current technology. Functional descriptions may facilitate this independence with implementation details specified later in the process perhaps leading to performance improvements in the ultimate implementation. The design tool should be able to identify the interactive component, specify it and evaluate it without describing unnecessary implementation detail. The grain-of-analysis should allow the experimenter to distinguish between two competing devices but offer no further detail.

So, a new proposed design tool should be able to accurately and relevantly represent the interactive component inherent in any proposed system irrespective of possible media and produce distinctions which allow rapid generation and assessment of potential configurations. An example of the possible distinctions between implementation details is that expressed by the current dichotomy between text and graphics based systems (interfaces are often a hybrid of the two but some are based purely on text). These issues should be transparent to any design tool and they should not affect the tools ability to objectively represent and distinguish between the two types. This

distinction is reflected in the choice of exemplar interfaces.

The ability of a tool to assess working systems is perhaps easier to formally analyse than its ability to produce meaningful descriptions of a proposed system. Again, the only true test of the latter type of analysis is to implement the proposed system and although this has proved impossible in the study under discussion it is hoped that the description produced can be shown to be accurate and as useful as its working counterpart. The tool can be re-used over a number of trials to re-evaluate refined designs incorporating subject feedback and material from the 'working system' analysis and this is demonstrated in terms of the authors proposed word processor design.

It is hoped that the design tool proposed can be used to evaluate any type of interface objectively and the data produced integrated into further studies due to the formal controls proposed and the objective analysis effected. This tool does possess specific areas of application where the descriptions produced may be of particular use, however it should be applicable to any situation. Temporal descriptions represent the novel aspect to be treated but they offer a framework within which many other aspects can be managed.

#### **14.1.1 The system specification cycle**

The author has developed a system specification cycle based upon the user modelling work described in Chapter 10 within which the proposed design tool can be implemented. The exemplar evaluation uses this system specification cycle. The cycle is outlined in the diagram below. The following eight stages represent the total design framework and it is believed that modelling on such a basis would enable a realistic dynamic model to be formed. This is the cycle utilised in the evaluation to be described later.

### **1. The data extraction phase:-**

This phase is divided into two separate types of activity depending upon whether the system to be described is a working or proposed one. These two activities utilise different techniques in eliciting the behavioural repertoire of the user.

The analysis of the working system relies on observational studies of a given types of user in their operation of the machine. This observation can be structured by observing the user performing a concrete set of tasks to which enable them to display their full repertoire and in this case express the full extent of their device knowledge (the commands available to them). The second type of evaluation involves a number of observations using paper prototypes and interaction simulations.

It is essential that naturally occurring behaviour in the target environment is captured as this represents the users actual behavioural repertoires. The user is asked to perform the type of behaviour they would normally undertake with that equipment perhaps as part of their 'normal' working day and feedback is obtained on their actual goals with later interviews. The validity of the data is paramount here as it will eventually be transferred into a generic format before it is analysed and this will impose any structure which is necessary. Users can be prompted with a number of representative tasks derived from the observation of the local user group however it important that the user employs their own particular style. A guide can be constructed indicating rules for generating the typical type of task inherent in an environment. Choice of style is left to the user and this offers a more natural body of interaction behaviour than the structured approach or that offered by other elicitation techniques.

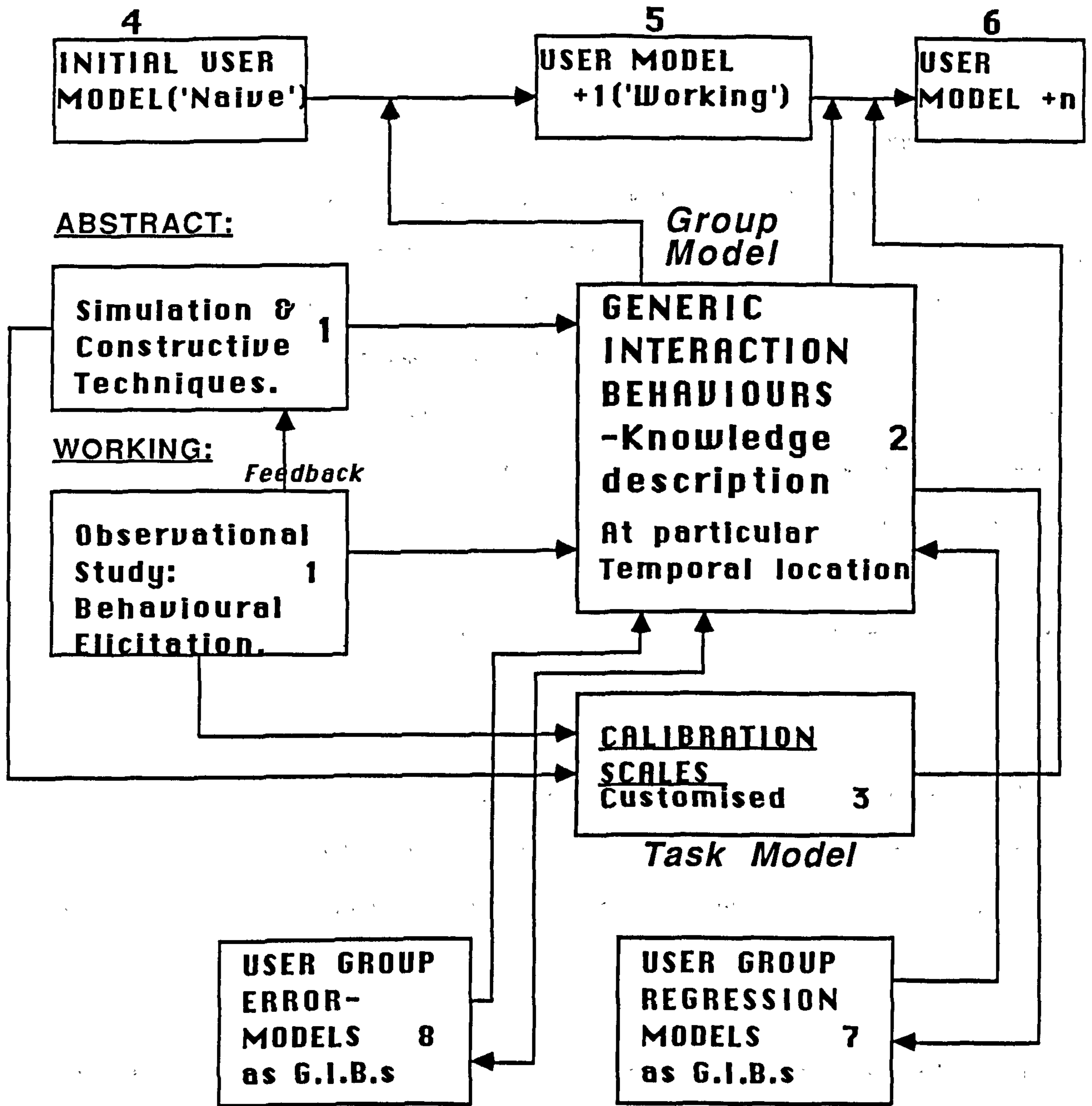


FIGURE 11 THE SYSTEM SPECIFICATION ACTIVITY MODEL

The techniques used for extracting abstract behavioural repertoires are not as valid as those described above as they are only partly based on material gained by studying the users actual behaviour with a machine. Feedback from such studies is obtained to validate the assumptions

made.

Behavioural simulations are used to ascertain the users generic interaction repertoire in the abstract scenario. Subjects are required to interact with a prototype or to take part in a real-world simulation in which objects (representing the virtual objects of the system) are presented to the subject and they are asked to manipulate them as they would in the execution of a given set of tasks. This simulation can be implemented using a rapid prototype in which the interface is presented without the underlying functionality and in which the screens are generated to resemble the proposed system. This simulation will be presented in the users typical environment as it is important that the user obtains the usual environmental task cues. Again, feedback can be provided by the experimenter as to the performance of the screen objects or devices in the simulation and the user can be interviewed as to their initial goals. The exemplar evaluation is of the latter type: the subjects are required to interact with a paper simulation of the interactive system.

A variation of this technique is 'Constructive Interaction' (O'Malley 1984). Here, the role of the second party would be less well-defined and the two parties would be able to express their natural problem-solving strategies. This enables the experimenters to examine the typical problem-solving protocols exhibited by subjects and examine particular issues in detail.

Structured interviews are used to gain feedback and obtain detail on subjects task methods and goals. This enables a structured view of the subjects actual methods to be elicited as amendments to the model formed from the observed behaviour. This feedback is elicited from a number of perspectives and can include the option to place the subject in a given 'hypothetical' situation to determine how they would respond.

Behavioural repertoires elicited from users reports of performance on similar machines could be

extrapolated towards machines of a similar configuration, with the appropriate controls and weightings for user group, to assist in the interpretation of data in a new behavioural model.

Finally, a users specific experience is identified by studying the performance of users with similar background and experience of systems and associating this with their behavioural repertoire in a given context (for instance, text-editing in UNIX). If a robust user group exists and the subject can be identified as a member of it the results from this group can be extrapolated to the new situation. Hence, data from studies of a working machine can be re-used and interpreted in terms of other behavioural situations to form a coherent model.

All of the above methods can be combined to form a comprehensive body of data.

## **2. The modelling phase:-**

An overall model of all of the subjects interaction behaviour is formed based on the data elicited using the above techniques. All of the full repertoire of subjects task actions are represented in the chosen medium: in the exemplar case the model generated is a generic model of the subjects knowledge of interaction and its devices. Each behaviour is represented uniquely in the generic kernel and a sub-kernel is formed for each type of user by indicating which subset of the full repertoire is held by each user group. The behavioural repertoires of members of each group can be correlated with each other and where correlation exists above a given value the user can be said to be a member of the user group. This forms an approximate model of the groups usage of behaviours whilst the next section addresses the sequencing of those behaviours.

This represents the group kernels. This model allows the statistical validation of users groups and experience levels.



### 3. Induction of temporal information:-

Next, specific temporal information concerning the sequencing of behavioural items at each level is derived from the data. Identifiable ordering, scheduling and prioritisation criteria are identified and represented in an overall model. This model is then partitioned according to the use of each of the behavioural strings by each user group and a sub-set of the total strings is identified for each group. These are stored separately as calibration scales for each type of user and experience-level and this provides the temporal information for the later versions of the user models. The types of strings or sequences which are identified are of the forms:-

- a then b then c      *OR*
- a typically occurs before b      *OR*
- a must occur before b

The occurrence of given instances of the above type of string can therefore be related to user groups and can be assumed to form the genetic repertoire of that group. The types of sequences displayed by given members of a hypothesised user group can again be correlated against each other to obtain a value of similarity which would support the robustness of the group.

This represents the task model group kernels. This model can, again, allow the statistical validation of users groups and experience levels, however the principle purpose is these strings is to be inserted into the task-path framework to form a total task model.

#### **4. Formation of the initial user model:-**

Once a sufficiently large amount of data has been collected over a wide range of subjects (this should of course include subjects of varying experience) to gain a representative sample of interaction behaviour for the purposes of the experimenter the initial user model on the overall temporal calibration scale is determined. Sampling posed some difficulties and these are discussed later.

Firstly an initial user state to be modelled is identified. This was taken to be the naive knowledge state in the exemplar evaluation, that is where the user has no experience of the given machine (or comparable machines in the abstract analysis) and this would typically be the initial state. The assumptions and predictions of this subgroup are determined by studying their behaviour as a subset of the main group. This would contain the actual set of behaviours or heuristics exhibited by the naive users in a given group in attempting to interact. The experience level of the subjects is assessed using some objective measures, in this case reports of interaction time experienced with a given task type on a named machine. The overall kernel is partitioned according to these values. This would of course necessitate the inclusion of a significant number of naive users in the original subject population. The naive generic repertoire is then tested for its ability to support the performance of a given set of tasks using the interface. Again, this model is refined with feedback from actual naive users to determine their actual methods for performing a task in order to validate the results.

#### **5. Formation of the second user model:-**

Subsequently, the next significant user state is identified by study of the data and the user groups. Some calibration is necessary to set the grain-of-analysis, however once the naive kernel was

formed the novice level followed on from that. In the exemplar this was a 'working knowledge' state in which the user has acquired sufficient device knowledge to be able to perform a given subset of typical tasks such that a large number of the routine and clerical operations associated with interaction are familiar to the user.

Hence, the total generic interaction behaviour model is studied with the perspective of subjects reported experience and the subset of the total kernel for this level of experience is identified. Behaviours which are not connected with the typical task repertoire are collected in the process which means that the model is not self-justifying: the associated extra stylistic behaviours inherent within a user group at the 'working knowledge' level. Interim stages can be identified between naive and working levels of knowledge but it is believed that these level are not significant in terms of the functionality of the machine. This state represents a range of experience bordering on the novice and various types of expert user (in various subsystems). Data from the independent calibration scale on user temporal behaviour at this level can be introduced here and incorporated along with the generic interaction model in terms of the ordering of the generic behaviours (as say a set of constraints for generating task descriptions).

#### **6. Formation of the nth.user model:-**

The next level of analysis is at the discretion of the experimenter and involves, for instance, the definition of a behavioural kernel for the execution of tasks in a given subsystem or mode. This level of experience was termed 'expert'. This can also be set to examine sophisticated knowledge of the main interface such as the use of customised or 'macro' commands and can capture ingenious use of the interface by users. This can again be identified by user reports of specific experience and can incorporate temporal data for that group which can be qualified using user feedback.

### **7. Induction of regression data:-**

The overall temporal framework has been established and additional data concerning the evolution of the model is now introduced. Empirical studies are undertaken as to the type of item, for example in terms of knowledge which may be lost from the initial model due to forgetting and restructuring and which are not totally reliable members of the interaction kernel.

A formal model of regression is identified for each user group according to typical items lost and this can be related to user group activity and used to define levels of experience. This indicates which users would require assistance in the learning of the interface.

### **8. Induction of error data:-**

Empirical studies are undertaken to determine the typical errors displayed by given groups. This study is carried out in the form of the recording of actual interaction errors for their identification in similar tasks or grouped in to meaningful types according to the conditions or task types which cause them to arise. Different models are formed for different levels of experience and incorporated into the model as necessary to define the amendments which must be made to task descriptions as a result of the errors of users. This enables designers to rationalise error behaviour in the design phase. Such a study is described in chapter 15.

#### **14.1.2 Some assumptions made**

This study focusses on the capture of practical interface knowledge and it is assumed that if users are sufficiently aware of a command for it to be useful during interaction they would display it in these trials when given the opportunity. This is based on the premise that users may be aware

commands but they are only demonstrated to be useful if users are prepared to employ them.

It was assumed that the subjects would behave with reasonable correspondence to their methods in a normal interaction and that they would not be unduly affected by experimental conditions. Every measure was taken to ensure that users followed their own usual and typical task structure and that the experimental conditions were those which constituted their normal working environment.

## **14.2 RATIONALE BEHIND THE SELECTION OF THE EXEMPLAR INTERFACES**

The exemplar Interfaces selected for the evaluation comprised of the UNIX (4.2) operating system and the VI editor contained therein running on a VAX minicomputer. Also selected was the APPLE MACINTOSH (version 2) system interface with the MACWRITE word-processing package running on APPLE MACINTOSH PLUS computer.

The MACINTOSH and the UNIX system represent examples of the most popular interfaces of their type. The former constitutes the pioneering WIMPS interface whilst the second is an industry standard text-based operating system. The author envisages that in the near future the large majority of systems will either be UNIX-based or possess a windowing interface.

The word-processing systems were chosen as subsets of the total system due to the preoccupation of a large amount of current HCI activity with word-processing and indeed this is why it has been exploited in previous studies (Card, Moran and Newell 1982). Hence, this can be taken as a useful analogy to HCI behaviour as a whole. Also this was a common subsystem between the two configurations and allowed objective, direct and meaningful comparison between behaviours in the two systems.

It has been stated in this thesis that a good design tool should treat implementation details as transparent issues and assess the quality of the underlying interface. The objective descriptions should also allow the two descriptions to be compared meaningfully. These two systems are characterised by the media within which the interfaces are realised. The MACINTOSH uses a hybrid of graphics and text to form the interface whereas the UNIX system is purely text based.

The MACINTOSH possesses a set of visible virtual objects and is based on a set of generic action operators such that all commands can be issued using a limited set of actions. These actions include pointing (using the interaction device (a mouse) to place the system cursor on top of a given screen image) and selecting (pressing and releasing the button on the mouse), for example. The UNIX system has a set of implicitly represented virtual objects such as files and directories and again a set of generic actions which can be performed upon these object to execute a set of tasks. The user activity in this case however is purely in the form of the generation of command strings on the command line and representation of the virtual objects is via purely textual representations (such as filenames).

The two systems provide ostensibly the same functionality to the user in that typical interaction behaviour - file handling, for example - and word processing activities are supported by the interface. The way in which these systems are implemented may, however, affect access to that functionality and the tool is able to detect this. The MACINTOSH has been designed with specific reference to Ergonomic issues whilst UNIX does not highlight these considerations as well. The technique offered will hopefully outline the discrepancies between the systems and offer some diagnostic feedback as to the functional differences.

The MACINTOSH explicitly represents and exploits a generic behavioural interaction set such that

all interactive behaviour can be defined in terms of a set of primitive actions stored in procedural memories. The behavioural content of UNIX is merely symbolic and a transitory representation of higher levels. The former approach is believed to be more appropriate for both designers and users and represents a more coherent approach interface design, especially with reference to object-oriented programming techniques, and as a result the focus in this technique is on object-based 'Generic Interaction Behaviour' descriptions. This technique can however represent the generic behavioural component of text-based systems and can, therefore, evaluate them on the same terms.

Choice of interfaces was also influenced by the fact that many of the available subjects possessed experience of both machines which was necessary for the design of the experiment: This enabled a repeated measures design to be used between machines to control for user experience. Subjects were split so that half encountered one machine first and half the other to control for practice effects.

This choice of interface also suggested a relevant abstract proposed design which could be examined. This was a word processing system devised by the author based on the generic objects and actions pioneered by the MACINTOSH. This enabled the author to demonstrate the potential for extrapolating material gleaned during the working study towards the abstract system design, due to the similarity of the interfaces. A full description of the proposed system is contained in the appendices.

The working hypothesis formed before the evaluation, in the context of the argument contained in the earlier chapters of this thesis, is that the MACINTOSH should perform better in an evaluation with this tool if this tool is to be seen to be valid. This is assessed in terms of:-

- the total number of commands supported within each interface
- appropriateness of those commands for the task

- the compatibility with the requirements of the various user groups

The description that follows is of the exemplar evaluation which was undertaken to assess the utility of the proposed design tool .

### **14.3 THE DATA EXTRACTION PHASE**

This elicitation was undertaken in two distinct phases:-

#### **Working system phase**

Forty subjects were asked to perform a number of tasks in their normal style on the two interfaces described. These tasks included the creation and editing of a document and a variety of file-handling and system administration functions (some example tasks were taken from a UNIX tutorial manual (Bourne) 1983 and the MACWRITE system manual.). It should be pointed out here that the exact nature of the tasks is largely irrelevant as it is the underlying behaviour which is being studied. The choice of clerical behaviour did, however, ensure sufficient similarity between users activity to enable comparison between subjects.

Twenty subjects used the UNIX interface and twenty the MACINTOSH interface. Subjects were matched between systems on a number of dimensions:

- the number of hours of experience with the target machine
- total number of hours of overall computer experience
- total number of hours experience of the type of task undertaken (word-processing)



All subjects were videotaped and comments were recorded. Trials commenced with the machine switched-off and users were required to initiate interaction themselves. Subjects were observed in their usual working environment where possible. The sessions recorded took up an average of half an hour depending upon the coverage of tasks. Guidance was offered by the experimenter in terms of task interpretation.

### **Abstract System Phase**

Ten subjects were required to simulate the performance of a set of tasks, derived from the above work, using a paper system presented by the author. Subjects reported verbally on the methods that they would use to perform the task. These reports consisted of how they would manipulate the paper objects given to them. The objects consisted of a set of virtual objects and a written set of operations which could be performed upon them - this system is described fully in the appendices. The experimenter responded to each of the reports by the user to indicate what the response to their action would be. The subject then progressed as far as possible with the task.

A second trial of the same tasks was performed in which two subjects from the same background were required to interact with the paper system and each other. Feedback was provided in the same way as in the first trial. All subjects were videotaped and comments were recorded.

## **14.4 THE MODELLING PHASE**

The videotape data was analysed for the existence of identifiable objects and actions within tasks. The data analysis techniques described in chapter 10 (Johnson 1986) assisted in the identification of genuine objects and actions. The tasks were described using functional descriptions based on those of Carter (1986). Names for the functions were agreed with the user according to Carter's method.

Frequency counts of objects and actions were used and it was assumed that the more often an object was used in the performance of a task the more likely it was to be a generic object. This allowed an initial list of generic interaction objects and actions to be performed.

The validity of objects and actions was then assessed via a number of criteria, namely:

- Genericity
- Centrality
- Representativeness

Generification of the objects and actions was carried out in a number of stages. Firstly, all of the actions and objects were listed in two separate lists. Secondly, the lists were reduced to a form in which each element appeared only once. Thirdly, the threshold of occurrence across subject and task groups was set by grouping them according to their occurrence in a given group context.

Generic elements were validated by separating each object from its generic category and asking subjects to identify whether they had used an element in their behaviour (as recorded on the video).

Centrality was assessed using frequency counts and the results were similarly amended after user feedback. Representativeness was assessed by placing each of the objects on a card and sorting them according to whether they were representative of a particular type of object. The results of this stage of data analysis were used to validate the kernel of generic objects. These rigorous tests allowed assumptions to be made on the accuracy and validity of these actions and objects as elements of interactive behaviour. This is very important in allowing the evaluation work to proceed as it is only by the accurate modelling of behaviour that the statistical tests described later become viable.

The data extracted from the subjects in both of the above phases were now partitioned into four distinct user groups based on subject background. These were naturally occurring groups within the subjects and were computer scientist, linguists, psychologists and artists based on academic background. The videotapes were then re-examined to establish the total behavioural kernel for each user group. Further feedback was obtained via interview for such a model with each of its members. User group integrity was tested by applying a correlation test to the frequency of commands used by different members of a given group and tests of the variance within the commands executed. This was compared with a correlation between different groups.

Hence, interaction behaviour has been described in generic form consisting of action/object pairs identified during the data analysis. Generic entities existing between machines have been recorded and these enable the comparison between equivalent functions on each machine.

#### **14.5 INDUCTION OF TEMPORAL INFORMATION**

Videotapes were subsequently analysed to identify generic strings of action/object pairs and equip the interaction model with a elementary temporal dimension by describing the sequences involved in the performance of sub-tasks and tasks using the generic actions and objects.

Analysis of the initial data was undertaken by the experimenter for the purposes of the forming the generic temporal model. The temporal model was based on the original observed behaviours and included commonly displayed strings of behaviours. Once the generic objects had been identified it proved easy to review the videotapes to look for commonly occurring strings of behaviour and subtasks. A kernel of common strings of behaviour was formed and this was partitioned into user groups. A candidate set of sub-tasks was also identified and users in each group were required to

assess their validity. Video tapes were again analysed to identify ordering, scheduling and prioritisation of these strings and sub-tasks using the following rules:-

- a then b then c indicates user ORDERING
- a typically occurs before b indicates user SCHEDULING
- a must occur before b indicates user PRIORITISATION

Videotapes were thus analysed for specific ordering, scheduling and prioritisation criteria and the string kernel was partitioned into user groups. Order of use of commands in performing subtasks was recorded along with order of subtasks in performing tasks on the user groups subset of the generic interaction model as a set of predefined strings. Behaviours were numbered and generic strings were recorded as sequences of digits for brevity. Correlations in terms of inter and intra-group similarity were performed in the form of knowledge strings of behaviour, as in the formation of the group object kernels.

#### **14.6 FORMATION AND EVOLUTION OF THE USER MODELS**

Subjects from the user groups identified were classified according to level of experience and the data was analysed from this perspective. Subjects were classified according the following criteria:-

- the number of hours of experience with the machine
- total number of hours of computer experience
- total number of hours experience of the type of task undertaken (word-processing)

All subjects were of undergraduate level so intelligence was assumed to be standard for the purposes of the experiment. A portion of the subjects were deliberately selected to be totally naive

to computers as this was the initial model to be formed.

Videotapes were examined to establish a total generic repertoire for each level of experience and this subset was partitioned from the total generic kernel (objects and strings - see appendix 1).

Correlations of inter and intra-level commonalities were carried out in the same manner as were performed on the user groups and the variance of behaviour within levels statistically assessed.

The kernel of generic behaviours possessed by the naive user was then tested against a set of simple tasks to ascertain whether the kernel could support the description of the task. The task description was converted to a generic form and the kernels examined for the presence of the required actions /objects and strings of actions/objects.

#### *The 'novice' model*

Subsequently, a working kernel of tasks which a 'working knowledge' level of user should be able to perform was elucidated and the total generic model was analysed to determine which users could perform those tasks. This was validated using the experience reports provided by the users so that upper and lower limits were formed to represent the range of working knowledge: more than 1 month but less than one year of intensive usage. This enabled chronological experience values to correspond to task ability and provided the basis for inducting further data. Generic kernels (strings and objects) were subsequently outlined for this level.

#### *The 'experienced' model*

An 'experienced' level of interaction was determined in terms of a kernel of tasks which could be performed by a user who had substantial experience of the word processing systems. The kernel

was analysed and a set of users identified who possessed the necessary subset of behaviours to describe that set of tasks. This was again quantified using the reported experience levels and the statement of system expertise: more than one year of intensive usage. Generic kernels (strings and objects) were subsequently outlined for this level.

#### **14.7 INDUCTION OF REGRESSION DATA**

The body of behavioural data elicited in this study was extended using data from subsequent studies to allow the experimenter to draw further conclusions about group and level specific behaviours. The methods used are described fully in chapter 15: the generic error and regression behaviours of each group were identified from the corresponding tasks in the experimental studies. These behaviours were recorded and their correspondence to a given group and level statistically established using correlation tests. This data was incorporated into the model by indicating which data objects and strings subset of the total kernel were subject to error or regression within the context of a group and a level.

This process could be said to be subject to experimenter bias, to some extent, however the only function of the experimenter was to identify and describe the original unique and novel behaviours which were actually displayed and these were subsequently confirmed by the user during data analysis. Experimenter bias, albeit unconscious, is a possibility however users were presented with the opportunity to reject the experimenters choice of items and select their own. It would be impossible with current techniques to simplify or shorten this process and maintain the validity of the generic objects produced.

## 14.8 RESULTS OF THE EVALUATION

Performance of the tool was assessed from a number of perspectives:-

1. Feedback from the users on their actual behaviour was compared with the behaviours which were initially included in the model, in terms of all user groups and levels

2. Statistical tests were performed to establish the robustness of the groups and levels chosen.

These were executed by analysing

- Single behaviours displayed (commands)
- Relational temporal strings (strings of commands)

3. An informal assessment of the performance of the tool in describing usability differences between the two machines

### 14.8.1 Feedback

This section describes how the initial kernel of interaction behaviours was amended during the generification process using feedback from the users to provide evidence of the validity of the data analysis method. Feedback from the users concerned:

1. interaction behaviour known which should have appeared in the model
2. interaction behaviour not known which shouldn't have appeared in the model
3. behaviour which was inaccurately represented in the model

The experimenter viewed the videotapes of interaction with the user concerned and informed the user of behaviours that had been identified: for example the experimenter might say 'we think you were renaming a file here', the user would confirm or deny this and the kernel of behaviours would be amended accordingly. This process allowed the experimenter to amend the model to reflect users actual behaviour and perhaps more importantly to validate the items which had been included and therefore examine the quality of the data capture method. The small number of changes which were necessary suggests that the data capture method was largely successful.

During the formation of the overall behavioural kernel feedback from the users caused 23 commands to be added 5 commands to be removed and 17 to be redescribed: this was out of the total initial behavioural kernel of 85 commands. This resulted in a total overall behavioural kernel of 103 commands. Hence, variation occurred in some 27% of the commands described.

The 'user group' and 'experience-level' kernels were analysed independently to give a clearer picture of the accuracy of the specific descriptions of users interaction knowledge. The computer scientists, who possessed 102 of the 103 commands, were responsible for the addition of 14 new commands which were unique to that group and required 12 commands to be redescribed. The linguists, who possessed 85 of the 103 commands, were responsible for the addition of 5 new commands and the removal of 2 with 2 being redescribed. The psychologists, who possessed 79 of the 103 commands, required that 3 new commands be added 1 be removed and 2 be redescribed. Finally the artists, who possessed 63 of the total kernel, caused 1 command to be added 2 to be removed and 1 to be redescribed.



User group	Commands Added	Commands Removed	Commands Redescribed
Computer Scientists.	14	-----	12
Linguists	5	2	2
Psychologists	3	1	2
Artists.	1	2	1

**FIGURE 12: AMENDMENTS TO THE BEHAVIOURAL KERNEL DUE TO USER GROUP**

Hence, a number of changes were made to the initial kernels, however the model was largely accurate.

The experience level kernels were also qualified by the feedback obtained. The interviews with the users classed as 'naive', who actually possessed 42 of the 103 commands, led to very few changes. One command was added 7 were removed and 5 redescribed. The working level lead to rather more changes: this level possessed some 61 of the total commands and lead to 10 additions, 8 removals and 12 redescriptions of commands. Hence, all of the groups could be said to possess at least a 'working' level of command knowledge. Finally, the consultation of the experienced level, who possessed all of the commands, lead to more changes and as a result 20 commands were added 1 removed and 12 redescribed. The large number of changes in the final group were as a result of the inclusion of the users specialist knowledge of the word processing systems. The significant number of changes in the working level were effected because of the importance of the correct representation of this level.

User level	Commands Added	Commands Removed	Commands Redescr.
Naive Users	1	7	5
Working Users	10	8	12
Experienced Users	20	1	12

**FIGURE 13: AMENDMENTS TO THE BEHAVIOURAL KERNEL DUE TO USER LEVEL**

Again, the initial models formed were largely accurate and accounted for a large proportion of user behaviour. Changes were minimal and largely due to specialist or esoteric knowledge.

Subjects reported methods were compared with the behaviour in the strings kernel via feedback interview and in the majority of cases the kernel was able to predict the majority of the behaviours evoked by the subject (around 80%). The predictions were, of course, over the limited domain of sample tasks included in the exemplar study and further work would be needed to refine the model completely.

#### 14.8.2 Statistical tests

This section describes how each of the hypothetical groups and levels was tested for integrity by comparing performance of users in the same group or level with each other and with users in other groups.

Three basic types of statistical test were used:

- Tests of the significance of the occurrence of a behaviour as condition of membership of a given group (Sign Test) compared to that of the whole sample
- Correlation between the frequency of occurrence of types of behaviour between
  - i Members of the same group
  - ii Members of different groupsto identify similar behavioural repertoires (Kendall's-t)
- Distribution of behaviour within various groups. Regularity of occurrence of behaviour among group members and the extent to which it characterises that group (Variance-ratio (f) test)

Tests were implemented on the basis of the users employment of:

- Single commands (represented as action/object pairs)
- Short strings of commands (represented as (up to) three action/object pairs)

Scoring of the data was done on the basis of frequency: frequency of occurrence of each type of behaviour was recorded for each user.

Correlation tests were performed by recording frequency of command usage for each member of a group. The pattern of command usage (that is the frequency of occurrence of given commands) was then compared with an equivalent subject in another group and with another subject within the same group, in a matched-pairs design. Scores were assessed according to frequency of occurrence

and were subsequently correlated to obtain a value representing the similarity between the subjects. Scores were converted into ranks to enable the test to be performed. Raw data is thus not presented as this normalisation of the data reduces the relevance of the absolute score. This scoring procedure is illustrated in the following example table:

Commands/ Strings	Group A		Group B
	User 1	User 2	User 1
1.	3	4	2
2.	6	3	4
3.	8	6	3
4.	2	5	6
5.	4	8	7
6.	1	2	8
7.	4	4	5
8.	6	3	3
9.	2	7	5
10.	5	5	5

**FIGURE 14: SCORING OF DATA FOR CORRELATION TESTS (EXAMPLE DATA)**

Where the correlation proved to be significantly higher in comparing subjects from the same group rather than subjects from different groups it is assumed that the similarity of behaviour is due to membership of that group. This scoring procedure allowed subjects from the same group to be compared as well as subjects from differing groups. This allowed inter-group effects to be compared with intra-group effects.

User Group	Between Groups Correlation (t).	Within Groups Correlation (t).
Computer Scientists	0.45	0.76
Linguists	0.32	0.65
Psychologists	0.34	0.54
Artists	0.21	0.53

**FIGURE 15: CORRELATIONS BETWEEN AND WITHIN GROUPS (t) USING SINGLE COMMANDS**

Figure 15 shows results of the correlation tests applied to user groups to test their integrity. The values were largely non-significant when behaviour was compared between groups whilst values for behaviour within groups were significant. Whilst the tests in themselves are of limited power in describing similarities between behaviour, comparing the two values gives a relatively reliable view of the difference within and between groups. This suggests that there was some effect due to group membership.

User Group	Between Groups Correlation (t).	Within Groups Correlation (t).
Computer Scientists	0.31	0.53
Linguists	0.20	0.49
Psychologists	0.084	0.22
Artists	0.043	0.20

**FIGURE 16: CORRELATION VALUES FOR USER GROUPS USING STRINGS OF COMMANDS**

Figure 16 represents the correlation values between and within groups assessed using simple command strings. The results are not as conclusive here as for single commands. All of the values were non-significant with the exception of behaviour within the Computer-Scientist and Linguist groups. This result could be explained due to the fact that both groups will have had formal training in the use of computers and the similarities of their behaviour may have been due to their ability to reproduce formally correct strings of command. Even this limited evidence of similarity is therefore useful in demonstrating the sensitivity of the design tool.

User Level	Correlations			
	Single commands		Simple strings	
	Between G	Within G	Between G	Within G
Naive	0.12	0.31	0.09	0.11
Working	0.42	0.52	0.06	0.32
Experience.	0.52	0.57	0.02	0.41

**FIGURE 17: CORRELATION ( $\tau$ ) TESTS FOR USERS LEVELS FOR SINGLE AND STRINGS OF COMMANDS BETWEEN AND WITHIN GROUPS**

Variance of the behaviours exhibited within groups and levels was tested to determine whether the users behaviour was due to a normal distribution or to the effects of the membership of the group. Variance of command usage was assessed by calculating the mean frequency of command occurrence within each group and then examining the variance of the various members of the group from that mean. The variance-ratio test was used in preference to the standard-deviation test as this test gives a value which is useful in the control of subject variability. It was hypothesised that if groups conformed to the normal distribution command knowledge would be evenly distributed

throughout each group and that each group would display the same pattern of command knowledge. It was assumed that where a non-significant f-value was obtained that the group could be said to display a significant similarity in behaviour: subjects could reliably be said to be a member of a given group if the variance remains non-significant.

The sign test was performed by recording whether the mean occurrence of a command within a group was more or less than the mean of the whole sample and allocating a sign as appropriate. The sign test was used in preference to the t-test as the significance of group membership cannot be tested in a true experimental fashion due to a number of factors: membership of a group is not a true independent variable, lack of stringent experimental controls (beyond those of user experience and variance tests) meant that extra-group effects could not be excluded and low subject numbers meant that strict sampling procedures could not be observed. The sign test provides a good comparative measure of any gross effect, as a result of group membership.

User Group	Sign test.			Within Groups Variance (f).
		L	T	
Computer Scientists	1.4	1	9	2.43
Linguists	0.86	11	24	3.28
Psychologists	0.75	4	10	2.15
Artists	0.07	4	21	3.13

**FIGURE 18 SIGN TEST AND VARIANCE VALUES FOR USER GROUPS USING SINGLE COMMANDS**

The above table demonstrates the validity of the groups as assessed using the sign test and the variance-ratio test. In this table and the following tables the L refers to the least frequent sign and T

refers to the total number of signs. The data shows that according to the frequencies obtained there is a significant difference between the groups and this is born out in sign values which are all significant with the exception of the artists.

The variance values show a non-significant value in all cases which suggests that the variance within the command frequencies for each group could have arisen as part of the normal distribution. Hence, this allows for the possibility that there is significantly similar behaviour within the group.

User Group	Sign test.		Within Groups Variance (f).	
Computer Scientists	0.7	1	12	4.23
Linguists	0.2	5	16	3.25
Psychologists	0.09	2	15	2.28
Artists	0.03	0	10	3.13

**FIGURE 19 SIGN TEST AND VARIANCE VALUES FOR USER GROUPS USING SIMPLE COMMAND STRINGS**

The above table represents the sign test and variance values which denote group integrity when assessed as the use of simple command strings. The above table represents values of level integrity according to correlation within and between groups. The above table demonstrates the validity of the groups as assessed using the sign test and the variance-ratio test. The data shows that according to the frequencies obtained there is a more significant difference between the groups than with single commands due to the distribution of signs and this is born out in sign values which are all significant with the exception of the artists.



The variance values show a non-significant value in all cases which again suggests that the variance within the command frequencies for each group could have arisen as part of the normal distribution. Hence, this allows for the possibility that there is significantly similar behaviour within the groups.

User Level	Single commands			Simple strings			
	Sign Test		Variance	Sign test		Variance	
Naive	0.12	L	2	3.1	L	4	1.1
		T	14		T	21	
Working	0.42	L	5	5.2	L	2	3.2
		T	20		T	12	
Experience.	0.52	L	9	5.7	L	2	4.1
		T	22		T	14	

**FIGURE 20: SIGN AND VARIANCE TEST RESULTS FOR SINGLE COMMANDS AND STRINGS AT VARIOUS LEVELS**

The above table demonstrates the validity of the experience levels as assessed using the sign test and the variance-ratio test. The data again points to the fact that according to the frequencies obtained there is a significant difference between the groups with reference to sign values. The sign values which are all significant.

Finally, the variance values show a non-significant value in all cases which again suggests that the variance within the command frequencies for each group could have arisen as part of the normal distribution. This result also allows for the possibility that there is significantly similar behaviour within the group.

### 14.8.3 Evaluation of systems

This section describes an informal evaluation of the two target systems based on the behavioural kernels derived from each system. These kernels relate to the range of behaviour which the two systems supported for each type of user. This allowed some comments to be made on:

- The overall range of behaviour supported by the system
- The appropriateness of a given system for a given user group
- The suitability of a given machine for performing a given task
- Estimated task performance times

The descriptions formed highlighted a number of the inherent differences between the two systems to be evaluated.

The machines were tested firstly by examining the total range of observed behaviours which had been identified by each machine. The MACINTOSH description contained fewer terms (83), than the UNIX description (103). The UNIX description constituted largely of declarative descriptors (objects) (85) with few non-specific procedural descriptors (18). This indicated the lack of exploitation of procedural elements and the greater load that the UNIX system placed upon the users memory. The MACINTOSH kernel displayed a good balance of procedural and declarative elements (35 and 48 respectively).

The MACINTOSH naive user kernel was able to generate a greater number of task descriptions (3) than the UNIX kernel (1) when tested with tasks in terms of the novice kernel and hence this supports the MACINTOSH's appropriateness for naive users. Conversely, the experienced computer scientists showed few performance discrepancies with either machine and actually

performed better with the UNIX kernel in many cases.

The MACINTOSH kernel was able to generate successfully a larger number of the task descriptions (5) than the UNIX kernel (2) with which it was tested. This indicated, to some extent, the appropriateness of the generic descriptors inherent in the system. This evaluation was carried out by generating a number (8) of simple tasks descriptions based on the action-object (Johnson 1986) method and matching the elements within such a description with those in the respective kernels. If all the necessary elements were present in a kernel then a preliminary assumption could be made on the comparative ability of the system to support that type of task. Examples of the tasks are provided below.

Finally, predicted task performance times were significantly shorter when assessed with the MACINTOSH kernel. The mean difference in task time amounted to some 35 seconds over the course of the two tasks which both systems could reproduce.

	<b>Task 1: Editing a Document</b>	<b>Task 2: Moving a file</b>
<b>UNIX</b>	205 seconds	53 seconds
<b>MACINTOSH</b>	156 seconds	32 seconds

**FIGURE 21: TASK TIMES GENERATED FROM BEHAVIOURAL KERNELS**

The sample tasks involved:

1. Starting a word processing function, creating a document, typing in a piece of text provided, saving the document and exiting the word processor.
2. Entering the file structure, locating the document, moving (renaming) the document and returning to the main interface

### **14.9 DISCUSSION AND CONCLUSION OF EXEMPLAR STUDY**

Essentially, three types of test were performed on the data produced by the exemplar evaluation:

1. Feedback was obtained from the interviews with the user to validate the model which had been formed. In this way it was possible to test the integrity of the kernels of user behaviour which had been formed.
2. The data from the exemplar was converted to frequencies to enable a number of statistical tests to be performed. These tests look at the range and distribution of behaviour and examine the significance of given behaviours in the kernel.
3. An informal evaluation was carried out using the behavioural kernel to facilitate comment on the quality of the design tool.

#### **14.9.1 Feedback**

The feedback obtained from users viewing video tapes of their own interaction behaviour lead to

relatively modest changes in the items within the behavioural kernel. This gives some preliminary support to the integrity of the kernel. A number of changes in certain areas are worth commenting on.

The large number of changes required by the computer scientists and experienced users may have been due to their explicit awareness of their own knowledge of computer commands and their ability to articulate them to the experimenter. This is difficult control for in the feedback interviews but reflects the generally greater number of commands available to such users.

A rough analysis of the command strings within the kernel indicated some validity within the strings kernel. The strings identified by the experimenter were validated but this may have been due to the fact that they were obvious patterns of behaviour which were common across all subjects. An example was: SELECT - FILE, OPEN - FILE. Fewer subjects were used in this analysis however initial reports confirmed integrity. The short-bursts of behaviour examined do not comment effectively on the tasks performed by the user as a whole however all the evidence shows that capture of behaviours is only reliable in short-bursts (Roberts and Moran 1984).

The feedback from the user did not only validate the kernels it also naturally provided evidence of the validity of the data capture method. Great effort was made during the generification process to ensure that the behaviours captured were the ones that the users actually produced and the validity of the tools depends upon this assumption. The changes made by the user indicate that the experimenters assumptions were largely valid and suggests that the behavioural capture technique is useful.

Whilst samples of the user population were smaller than would have been desired, and the amount of activity studied amounted to less than would have been desired, it is believed that a workable

sample of user behaviour was obtained. A useful and manipulable kernel of behaviours was captured which enabled the experimenter to model a number of typical situations and this kernel was validated by user feedback. Finite restrictions on sample size were, of course, in operation and it is believed that a much larger sample would have been needed to capture some of the more esoteric behaviours.

Future work is needed to extend upon the kernel generated in these circumstances, however this sample would need to be substantial to ensure the capture of a complete model of behaviour and the experimental condition would need to be duplicated for the results to be applicable: this would enable accurate, repeated predictions of user behaviour over a number of circumstances via the development of a robust formal model. This study has shown that behavioural models can be formed and that they are useful. Further work is also needed to extend the kernel to form larger models.

### 14.9.2 Statistical Tests

To recap, the statistical tests tests performed were essentially looking to evaluate one of two things:

1. The range of behaviour exhibited
2. The distribution of that behaviour
  - a. Within groups
  - b. Across groups

In order to test:

1. The representativeness of particular behaviour for particular groups
2. The accuracy and validity of the behavioural model of interaction

Hence, three basic types of statistical test were used:

- Tests of the significance of the occurrence of a behaviour as condition of membership of a given group (Sign Test) compared to that of the whole sample
- Correlation between the frequency of occurrence of types of behaviour between
  - i Members of the same group
  - ii Members of different groupsto identify similar behavioural repertoires (Kendall's-t)
- Distribution of behaviour within various groups. Regularity of occurrence of behaviour among group members and the extent to which it characterises that group (Variance-ratio (f) test)

'Within group' and 'within level' correlations were significant in the majority of cases and tended to be of greater significance than the 'between group' and 'between level' correlations. This tends to indicate the presence of some group effect. This is due to the fact that the behavioural patterns of users in the same group was more similar than those in different groups in terms of the relative frequency of given command types that they produced. It is difficult to formulate direct measure of similarity using this method, however it is possible to demonstrate a comparative superiority in the 'within-group' conditions which suggests some similarity in behaviour.

Variance of command usage was assessed by calculating the mean frequency of command occurrence within each group and then examining the variance of the various members of the group from that mean. It was hypothesised that if groups conformed to the normal distribution command knowledge would be evenly distributed throughout each group and that each group would display the same range of command knowledge. It was assumed that where a non-significant f-value was obtained that the group could be said to display a similarity in behaviour: subjects could reliably be said to be a member of a given group if the variance remains non-significant. Non-significant f-values were obtained in the majority of cases however variance proved to be not significant in almost all cases due largely to small sample sizes. It is not expected that subjects would all produce identical behaviour, however the variance in this case may have been rather more than one would expect from random variation. Variance was more pronounced in the computer scientists and experience users. Again, this can be largely attributed to their greater knowledge of the command language which enabled them to show more diversity in their behaviour.

The sign test was performed by recording whether the mean occurrence of a command within a group was more or less than the mean of the whole sample and allocating a sign as appropriate. The sign test was used in preference to the t-test as the significance of group membership cannot be tested in a true experimental fashion due to a number of factors: Membership of a group is not a true independent variable, lack of stringent experimental controls (beyond those of user experience and variance tests) meant that extra-group effects could not be excluded and low subject numbers meant that strict sampling procedures could not be observed. The same could be said to apply to the application of the sign test, however since the scoring involved only the recording of whether frequencies were more or less in relative groups the requirements for experimental controls to eliminate slight variations in performance would have been diminished. The sign test does, however, provide a comparative measure of a gross effect, as a result of group membership.



The sign test values proved to be significant in most cases at a low level of significance: this suggests that the frequency of production of the given behaviours was significantly higher in the given experimental group than in the group with which it was compared.

### 14.9.3 Informal Evaluation

Finally, the tool was able to comment on the interfaces being tested to describe comparative superiority in particular elements of the equipment in terms of its ability to support users task-performance. This in turn facilitates some comment on the design tool itself and the type of comment provided suggests that the results may be useful in design. The evaluation was performed by looking at the ability of user to perform tasks on the various machines was tested in four ways:

- The overall range of behaviour supported by the system
- The appropriateness of a given system for a given user group
- The suitability of a given machine for performing a given task
- Estimated task performance times

The MACINTOSH kernel was shown to support more procedural (action) elements and was correspondingly more usable by naive users (their kernel supported more task descriptions in the MAC context). The UNIX interface was shown to support more declarative elements (objects) and was favoured by experienced users. Hence, an absolute value of the superiority of one machine is neither useful nor appropriate. The application of the MAC in physically intensive applications, for example graphics is suggested by its greater procedural repertoire whilst the provision of a greater number of declarative elements suggests the UNIX system may perform better in complex problem-solving.

The ability of the MACINTOSH to support naive users is a useful in design as it suggests that systems targetted at naive users can be more readily exploited if they display a graphical interface. The ability of the UNIX interface to support experienced users reflects a demand for more powerful and more specific commands. The design tools is useful as it is able to comment upon this type of issues and hence allow the designer to make decisions about the appropriateness of a particular interface style for a given user group or task. This shows that the tool could contribute usefully to design however these assumptions are not based on any formal analysis of the users behaviour.

The kernels generated by each machine was tested against a number of action/object based task descriptions and the MACINTOSH was able to describe more tasks. Primitives and strings of behaviours obtained from the behavioural kernels were combined to match task descriptions produced by the author. This may have been due to the ease of modelling of behaviour with the MACINTOSH due to its object-oriented interface.

Finally, predicted task performance times were significantly shorter when assessed with the MACINTOSH kernel. The mean difference in task time amounted to some 35 seconds over the course of the two tasks which both systems could reproduce. This was typically due to the use of fewer operators and to a shorter mean operator production time. This is congruent with the performance of the users during the trials. Timing of the various operators across a number of subjects allowed average values to be established for each command on each machine. A summative model was formed for each of the successful task descriptions described above by adding the time taken on each of the predicted behaviours. Despite the reservations expressed earlier concerning this type of model (Card, Moran and Newell 1982) it does allow a comparative estimate of the suitability of the two machine for the performance of the tasks selected. Task times generated by the summation of temporal values for each operator were subject to many of the

criticisms of current methods however they provided a rapid assessment of the subjects ability to generate the primitives of interaction.

#### **14.9.4 Grain-of-Analysis**

The level of analysis relevant for a particular evaluation will necessarily vary according to the requirements of the experimenter and this was allowed for in this study. The evaluation described above demonstrates another distinct advantage of dynamic modelling in that evaluation can be carried out on a variety of levels. This diversity of levels was represented within the current work by two types of model:

1. The user group model
2. The user task model

The first of these represented the large scale evolution of the user over the course of interaction and specified changes between novice, naive and experienced users. This level of evaluation produced concrete results when applied over the range of trials highlighted the evolution of the users between experience levels. The user task model displayed the change within the user within one session and was represented as a series of strings and scheduling criteria

The former type of evaluation facilitated representation and statistical analysis of the changes in user group (or the discrepancies in changes between user groups) behaviour due to dynamic factors over a prolonged period of interaction. The second approach highlighted changes in individual and single group behaviour down to the level of specific task elements over one session. This was demonstrated by the ability to produce accurate and verifiable task descriptions for a range of users.

The units of evaluation in the first model are the user group models and hence the sample size, trial length and grain-of-analysis was fairly large. These user groups were subsequently evaluated using statistical analysis. The units of evaluation in the second study are the task elements and behaviours and hence the sample size, trial length and grain-of-analysis can be said to be fairly small. The generic interaction primitives used for the basis of the model proved to be self-contained and well-integrated and hence they were useful in the construction of models of various sizes.

In summary, the first type of model highlighted the dynamic variations within behaviour at an organisational level, showing the variance between given users to be the result of group membership whilst the second type of model focussed on the task-level and provided specific instances of task behaviour when evaluated against user level repertoires.

A number of general criteria were used to help in determining which level of approach was appropriate, however the appropriate level of analysis could not be fully determined until evaluation had been initiated and the generic kernel had been examined to see if various target behaviours have been captured and modelled. Hence, the evaluation was an iterative one in which the grain is adjusted to capture the appropriate behaviour.

Grain-of-analysis was adjusted to account for the observed variation in the sampled behaviour and this optimised reliability: this process utilised the statistical analysis described above to determine the required sample size to capture the required effect. Measures were taken to ensure the capture of behaviours which fell outside the normal range but this involved the capture of redundant word-processing objects which proved to be of little use in generating typical tasks.

### 14.9.5 An analysis of the synthetic constructs inherent in the method

The two key synthetic constructs featured within this method which play a significant role in the structure of the evaluation paradigm are:-

- the three-dimensional model
- the calibration scales

The validity of these synthetic constructs is based upon the soundness of the assumptions made when collecting and analysing the data from which they are built.

The assumptions made during the formation of the three-dimensional model are that the user groups, the user levels and the tasks with which the model is evaluated are realistic constructs. The analysis of the user groups and levels described above, in terms of both common strings and common generic objects demonstrate a statistical validity in the vast majority of cases. The robustness of these groups as synthetic constructs supports the validity of the model as a whole. Secondly, the choice of tasks to enable the elicitation of data to construct the knowledge models is validated due to the fact that subjects were encouraged to use natural methods and were given little prompting and by the fact that consistency across tasks was encouraged due to the experimenter interpreting tasks descriptions for the user in the context of previous subjects methods.

The calibration scales meanwhile are given structure and integrity by the validity of the generic units from which they are formed. The assumptions made concerning the self-containedness, robustness, integrity and modularity of these units is supported by the extent to which they are validated by the user during the generification process. The units are not only identified for their genericity but also for the centrality and representativeness and these units are again tested outside

their context to ensure that they are valid.

Hence the existing statistical analyses and the rigorousness of the data extraction methods have previously validated these constructs.

FOOTNOTE: It should be noted that the above is an empirical but not an experimental evaluation of a design tool. Hence, the study has not observed experimental convention rigorously. Whilst experimental techniques are used in the study - statistical analysis, for example - the conclusions drawn are not experimental ones.

## **15. EMPIRICAL WORK**

This chapter describes a number of experimental studies conducted to provide evidence in support of a number of the principles underlying dynamic Human-Computer Interaction.

### **15.1 INTRODUCTORY STATEMENT**

A number of principles have been identified which describe the dynamics of behaviour at the Human-Computer Interface. The principles outline a set of explicit behaviours which dynamic systems should display. The principles can therefore represent a number of testable experimental phenomena from which hypotheses can be generated. Substantial evidence exists in the literature to support the majority of the principles however it does not offer sufficient support to certain specific principles and hence, these merit special study. This chapter describes some simple experiments to provide evidence to support these principles and therefore complement the evidence which exists in the literature. This provides a more complete body of evidence to support the theoretical framework for dynamic HCI.

#### **PRINCIPLES OF DYNAMIC INTERACTION:-**

1. Changes occur in user, machine and task state during and between interaction with respect to motivation, goals and memory
2. Changes in user state may be due to interactive effects which arise as part of ongoing or previous interaction
3. Interaction occurs in real-time, subject to user inference, experience and task context

4. Interaction involves individual users and machines which differ significantly
5. Interaction is a multi-level process in which users address task issues at a number of levels
6. Interaction is a multi-dimensional process
7. Interaction has a heuristic or pragmatic basis in which users are involved in utilitarian decision-making
8. Interaction possesses holistic properties
9. The perception of the interface is an active process
10. Error behaviour is natural and error-handling is a dynamic process
11. Users regress, in terms the loss of items from their interaction model, during and between interaction

Empirical evidence from various fields of study has been cited in the earlier chapters to support some of the above premises and dynamic models of behaviour have been identified in other studies in isolation from HCI. For instance the dynamic nature of intra-human communication has been displayed in both psychological (Christie 1986) and HCI (Evans 1974) studies and this offers support for the broad notion of HCI as a dynamic process.

Multi-dimensional (Principle 6) (Johnson 1984, Kieras and Polson 1982) and multi-layer (Principle 5) (Moran 1982, Clarke 1986) interaction models have been well discussed and these now seem to be accepted, to some extent, as valid design tools for the analysis of some of the dynamic properties of in HCI. The notion of heuristic and pragmatic models of behaviour (Principle 7)



based on goal-driven (Shneiderman 1986) and motivational (Malone 1982) aspects of interaction seem to be well accepted as valid and examinable dynamic properties of interaction. Finally, the existence of dynamic general models of perception (Newell 1964 et al.) would seem to validate, to some extent, the assumption of dynamic or active perception in the study and modelling of HCI (Principle 9) (Green and Doubleday 1988). Additionally, the evaluation described in the previous chapter provides evidence to support Principles 1 and 4: it examines the changes in user behaviour over time and looks at significant differences of behaviour between user types.

In spite of this body of data, and the novel empirical work already described, a number of the principles require experimental investigation. Study 1 looks at principle 8 and examines how the users holistic perception of an interface can affect performance with future interfaces. Study 2 looks at principle 11 and attempts to categorise the types of items which are lost from the memory of given users groups. Study 3 looks at principle 7 and examines how users can make pragmatic decisions about how they perform tasks based on the cues provided by the task. Finally, study 4 looks at principle 2 and shows how an interactive event can affect the way in which users approach interaction.

## **15.2 STUDY 1**

### **15.2.1 Introduction**

One of the dynamic principles identified within this work is that elemental descriptions of an interface are not always sufficient to describe all relevant interactive phenomena. Users seem to be sensitive to some holistic aspects of the interface, an example of such an aspect is the overall appearance or function of the computer (Malone 1984). Hence, a user who perceives a computer or one of its features as being similar to another known computer or feature may assume that a computer being used will perform in similar fashion to one with which they are familiar. The user will therefore make errors in the use of commands based upon their experience of similar systems

and it is believed that the type of error behaviour produced will be consistent with membership of a given user group.

This phenomenon is similar to the notion of 'Interference' (Postman 1961) which affects human memory and causes similar items within memory to exclude one another. This effect can occur due to morphological or semantic similarity between items in memory and effects have been described for each. The effect is postulated to occur due to the structure of memory which only stores the unique items once and shows difficulty in distinguishing between similar items during retrieval.

Hence, the effect due to 'interference' is associated with the correspondence between new stimuli experienced and the existing contents of memory. A users experience or knowledge of a system is determined, to some extent, by the users experience or background hence it would seem reasonable to assume that there should be some similarity between the interference shown by user with a similar background.

It is this effect which allows 'transfer of training' to be applied between systems, however such transfer can only occur where the users retention of learned material and its application to new interfaces is well-understood. In order to examine the nature of the 'interference' process which facilitates transfer of training it is necessary to identify some explicit manifestation of this interference. It is impossible to show that a user produced correct behaviour with a novel interface due to experience but we can study the errors made and hypothesise about why the user made that error.

The study of users errors is also interesting from another perspective: this work supports the notion that error behaviour is a natural phenomenon (Gilb 1977) and that it can be accounted for and protected against in the design process. It is believed that errors can arise as a function of the type of activity being undertaken and the experience of the user. This type of data would form a valid part of any behavioural model of HCI (albeit erroneous) such as that exploited in this work. This would

enable error data to be localised to a given group and utilised to qualify the behavioural model in the description of tasks such that error types can be predicted and designed for within particular target user groups.

Studies have been made of common types of user error (Bailey 1974) and many types of category outlined, however this data is difficult to apply in design without specific knowledge of where such errors might occur. This study aims to identify specific user group commonalities and this offers a basis for the integration of this data in a meaningful sense. Hence, user group a. may tend to exhibit error-prone behaviour in task type b. on machine c. 'Errors' can often not directly be attributed to user action however support strategies must exist for recovery from error situations'. These again can be related to user experience of recovery, task demands and machine incompatibility and it is hoped that these factors can also be controlled for with a comprehensive error-model.

The experiment described below looks at the type of errors made by users when faced with a new operating system and assesses to what extent those errors were due to their membership of a given user group. It is hypothesised that users with a particular background will tend to make errors which are, or bear some resemblance to, commands with which they are familiar. Hence, it is possible to categorise the errors produced by the user into one of five limited 'types' of error according to the degree of resemblance to the predicted (UNIX) commands: UNIX errors, where a legal UNIX command is used instead of the correct MULTICS command; Hybrid errors which contain 50% of the letters of a UNIX command; CP/M and VML errors which are legal commands in other operating systems; Errors which fit into none of the above categories.

The identification of a method by which discrete user groups can be identified is a way in which this work can influence the design of computer systems and training material: if we know the types of errors which particular users are likely to make then we can account for these in the design process or focus on them in any retraining material.

The first experimental hypothesis is that:-

- There will be some 'interference' from existing computer experience in the learning of a new and similar interface, that is that users with UNIX experience will produce significantly more UNIX-type errors when using the MULTICS system than those with no UNIX experience

The second hypothesis is that:-

- There will be some significant difference in the type of errors exhibited by users from the differing users groups or backgrounds but with a similar amount of UNIX experience

### 15.2.2 Method

Eight subjects with experience of the UNIX operating system but no experience of the MULTICS operating system, and eight subjects with no UNIX experience but with some computer experience, were required to perform a number of tasks on the MULTICS operating system. These tasks included file-handling operations (moving a file from one location to another, renaming a file, listing the files in a directory) creation and editing of a document (entering VI, inputting a pre-prepared document, exiting VI and saving the document to a named directory), use of the electronic mail system (entering UNIX Mail, inputting a pre-prepared mail message, sending the message to a named distribution list, leaving UNIX Mail) and changes to the user setup file (changing the password, changing access privileges on named files, returning to the users directory).

Four subjects were obtained from each of the user groups identified in the evaluation study : these were computer scientists, linguists, artists and psychologists. Subjects in the two conditions were matched for levels of experience of UNIX and for general computer experience (both measured in

hours). It was assumed that a subject with more than 50 hours UNIX experience was an 'experienced' UNIX user.

Access to the manual was permitted for the first trial. All the subjects were videotaped, keystrokes and comments were recorded. No assistance was offered by the experimenter. Subjects were required to return one week later and perform the same tasks, however access to the manual was denied for the second trial. All subjects were videotaped, key strokes and comments were recorded. No assistance was offered by the experimenter.

The videotapes were analysed and commands which the users reproduced were recorded. All of the erroneous commands were identified and recorded against the user who had produced them.

### 15.2.3 Results

#### *Resume of Hypothesis*

The first experimental hypothesis is that:-

- There will be some 'interference' from existing computer experience in the learning of a new and similar interface, that is that users with UNIX experience will produce significantly more UNIX-type errors than those with no UNIX experience

The second hypothesis is that:-

- There will be some significant difference in the type of errors exhibited by users from the differing users groups or backgrounds but with a similar amount of UNIX experience

Commands used in the second trial were analysed for resemblance to UNIX commands.

Commands were analysed on the basis of morphological similarity, in that they possessed 50% of the characters of the UNIX command, and semantic similarity, in that they performed a similar function. Of 105 commands used twenty-six were erroneous, to various extents. Of that twenty-six eight were actual UNIX commands which were morphologically similar and two were actual UNIX commands which were semantically similar. Twelve of the remaining commands represented a hybrid of the UNIX and MULTICS formats with at least 30% of the letters of both commands. Four of the errors represented commands which were of neither format and two of these were identified as commands from the CP/M systems and the VML system.

Examination of the distribution of errors between subjects showed some common area of behaviour. Errors of the three types were evenly distributed between the four subjects with two subjects showing two of the first type of error and two subjects three of the first type. The hybrid commands were distributed evenly with three errors each and the 'wildcard' errors were shared between three subjects with subject two making two errors and subjects one and four one error each.

Commands in the second trial were analysed for all errors and the results are displayed in figure 22.

Group		Unix		Hybrid		C/PM		VML		?	
		Subject 1	Subject 2	1	2	1	2	1	2	1	2
UNIX	1	2	4		1		1				1
	2	2	5	1	3						1
NONE	3		1	--	3	1	1		1		
	4	1	2		3	1		2			

FIGURE 22: ERROR DISTRIBUTION IN EXPERIMENTAL TRIALS

Looking at the total numbers of commands produced it is possible to gain an estimate of the significance of the numbers of commands which were produced erroneously due to confusion with other command languages. There was a mean of 63 commands produced over all of the trials and this value was used as the basis for assessing the significance of the error-behaviour.

The UNIX group showed a mean of 3.25 UNIX errors each whilst the non-UNIX group showed a mean of 1.25 UNIX errors. The group with UNIX experience produced a significantly higher number of UNIX related errors when assessed with a student t-test ( $t=2.5$ ,  $df=6$ ,  $p<0.05$ ).

Subject	Total Commands		
	First Trial	Second trial	Overall
1.	64	52	115
2.	73	64	140
3.	59	54	115
4.	72	66	141

**FIGURE 23: TOTAL COMMANDS PRODUCED FOR THE TWO TRIALS**

Frequency of each type of error was calculated for each subject and comparison was made between the comparative numbers of each error type produced by each subject. The scores produced were correlated as matched pairs using Kendall's tau. This is the same technique as that used in the evaluation study. A matched pairs design was used to control for spurious factors. Raw data is not shown here as this is a rank-order correlation and frequencies were converted into ranks before testing.

Correlations between intra-group behaviour proved to be significant ( $\geq 0.5$ ) for three of the four groups (0.8, 0.6, 0.5) according to the types of error displayed when tested with a Kendall's tau in independent pairs. The final group (the artists) showed a slightly sub significant value of tau (0.4). Correlations between Inter-group behaviour proved to be insignificant ( $\leq 0.3$ ) in all four groups using a Kendall's tau in matched pairs between groups. Correlations were performed by listing the numbers of errors of each type displayed for each user comparing this with a.) members of the same group and b.) members of different groups.

#### 15.2.4 Key points from study 1

A number of key points arose from this study:

- Subjects with UNIX experience produced significantly more UNIX related errors than those with general computer experience, as assessed by a student t-test
- Knowledge of an operating system interferes with the learning of a new operating system and this is demonstrated in the type of errors made
- The errors were significantly similar for all of the subjects within two of the four user groups, as assessed by a Kendall's tau in matched pairs

A further discussion of these points can be found at the end of this chapter.



## 15.3 STUDY 2

### 15.3.1 Introduction

One of the principles of dynamic HCI suggests that users interaction knowledge changes and between sessions and that knowledge will change to reflect ongoing task requirements. Indeed, studies suggest that users tend to lose material concerned with how to operate an interface due to forgetting between interactive sessions and due to those items being excluded from the limited memory store by other more novel items during interaction (Carroll 1982).

The process by which this activity takes place could be controlled, to some extent, by the user determining which information is relevant to current needs. The users current needs will typically represent the set of commands, for example, which support the tasks which the user is currently engaged in and this in turn may be associated with the tasks inherent in his or her domain of study (Greenberg and Witten 1983). Hence, by this means, the users preferences in recalling interface knowledge can be deduced and some of the behaviour which arises as a result predicted.

Eason's (1977) work suggests the existence of a typically limited command set which could be extended further to define the particular sets which are associated for given user groups and this set could be exploited in the design of interfaces. Hence, it is believed that there exist commonalities in the way users will lose items from memory due to forgetting between or within interaction and that these commonalities will be related to membership of a specific user group or user experience.

The author refers to the changes in the users interaction model due to forgetting and reorganisation of interface knowledge by the term 'regression'. Identifying a general 'regression' model would be problematical as the factors which may affect the manipulation of interface objects could be manifold, however within limited user groups specific task demands and experience could be isolated which could help in determining the behavioural processes which underlie 'regression'.

'Regression' has important ramifications for the design of interactive systems. Designers will want to provide the most efficient and useful set of commands and the study of those items which the user will most rapidly or easily reject will allow the design of useful command sets. A command set is not useful if it puts excessive strain on the user or is not memorable. The identification of those items which are particularly susceptible to rejection within a given user group allows the designer to make predictions about the types of commands which will be appropriate for given user groups (Carroll 1982).

The experiment described below uses a similar method to the previous experiment, however in this case the behaviours which are being examined are the items which the user cannot reproduce in the second trial.

The experimental hypothesis is:

- That there will be significant similarity between the items lost from memory displayed by members of a given user group.

### 15.3.2 Method

Eight new subjects with experience of the UNIX operating system but no experience of the MULTICS operating system were required to perform a number of tasks on the MULTICS operating system. These tasks included file-handling operations (moving a file from one location to another, renaming a file, listing the files in a directory) creation and editing of a document (entering VI, inputting a pre-prepared document, exiting VI and saving the document to a named directory), use of the electronic mail system (entering UNIX Mail, inputting a pre-prepared mail message, sending the message to a named distribution list, leaving UNIX Mail) and changes to the user setup file (changing the password, changing access privileges on named files, returning to the users

directory).

Two subjects were obtained from each of the user groups identified in the evaluation study : these were computer scientists, linguists, artists and psychologists. Subjects in the two conditions were matched for levels of experience of UNIX and for general computer experience (both measured in hours). It was assumed that a subject with more than 50 hours UNIX experience was an experienced UNIX user.

Access to the manual was permitted for the first trial. All the subjects were videotaped, keystrokes and comments were recorded. No assistance was offered by the experimenter.

Subjects were required to return one week later and perform the same tasks, however access to the manual was denied for the second trial. Subjects were asked to report when they were unable to reproduce a command which they had used in the previous trial. The frequency of these events was recorded. The function of the command was recorded (for example, 'this is a command to change access privileges to a file') and these descriptions were matched to the appropriate UNIX command. All subjects were videotaped, key strokes and comments were recorded. No assistance was offered by the experimenter.

The videotapes were analysed and commands which the users successfully reproduced were recorded.

### 15.3.3 Results

#### *Resume of hypotheses*

The experimental hypothesis is that:-

- There will be significant similarity between the items lost from a memory displayed by

members of a given user group.

Frequency of occurrence of each command forgotten was calculated for each subject and comparison was made between the comparative numbers of each error type produced by each subject. The scores produced were correlated as matched pairs using Kendall's  $\tau$  in the same way as the previous study. This is the same technique as that used in the evaluation study. A matched pairs design was used to control for spurious factors although poor sampling and low  $n$  values caused some problems and this is discussed later. No raw data is displayed as this is rank-order correlation and frequencies were converted into ranks before testing.

Correlations between Intra-group behaviour proved to be significant (0.7 & 0.9) for two of the four groups according to the types of command recalled displayed when tested with a Kendall's tau in independent pairs. Of the other groups the 'linguists' showed a non significant value (0.4) and the artists showed a clearly insignificant value (0.2). Correlations between Inter-group behaviour proved to be insignificant ( $\tau < 0.57$ ) in all four groups using a Kendall's tau in matched pairs between groups. Correlations were performed by listing the frequency of items of each type forgotten displayed for each user and comparing this with: a. members of the same group and b. members of different groups.

A Sign test was performed to see if users of a given group committed significantly more errors than the mean of the other groups. The results of these tests proved to be non significant. The sign test was used as a test of significance in preference to the t-test as the controls over the subject sampling and the experimental variables were not sufficient. No raw data is shown here as scores were normalised into pluses and minuses before the test was carried out.

### 15.3.4 Key points from study 2

A number of key points arose from this study:

- The commands forgotten were significantly similar for all of the subjects within two of the four user groups, as assessed by a Kendall's  $\tau$  in matched pairs
- The tendency of group members to commit similar numbers of errors suggests that error behaviour can be linked to group membership, however these results must be interpreted in the light of a small sample size
- The total number of errors produced by each group was not significantly greater than the mean of the other groups, as assessed by a sign test, hence overall error behaviour did not vary between groups

A further discussion of these points can be found at the end of this chapter.

## 15.4 STUDY 3

### 15.4.1 Introduction

The principles of dynamic HCI suggest that significant change can occur in the mental state of the user due to local task requirements and that this change will be reflected in behaviour: local task requirements can take very many forms and when placed in the context of the possible variations in interaction knowledge the resulting interaction state inherits the ability to be able to fluctuate dynamically (Green et al 1983).

Users are obviously required to respond to changes in their interaction strategy based on the particular demands of the task that they are carrying out but the question which remains is how this process is exploited in systems design. Exploitation is via the prediction of user behaviour in target task situations. Predicting the type of changes which might occur is a hazardous process as there are many relevant factors which can influence the production of interaction behaviour. If, however, the experimenter is provided with a method for the capture and representation of users interaction-device knowledge then it may be possible to hypothesise how he or she will react in given tasks circumstances. This would be a useful process in the design of computer systems as, given the known existing interaction knowledge of a target user group, it will be possible to test the task descriptions which will arise from an interface design. This will allow the evaluation of the various task strategies which a user might adopt during interaction.

Eason (1977) identifies a kernel of working knowledge which users will maintain during interaction. In specific task situations, however, this subset of commands (file-handling, for example) is not sufficient or economical enough to meet the users needs and the user will undergo a process of identifying a more appropriate specialist command. It is hypothesised that in these circumstances user will adapt their interaction strategy to a more economical format such that an optimal strategy is maintained but that a sub-optimal strategy will be retained until the perceived value of accessing a given command is high enough.

The experiment which follows seeks to demonstrate that it is possible to make some accurate predictions about user behaviour, by referring to the demands of the task, with little or no knowledge about the user. It is hypothesised that the formation of accurate models of the knowledge of users groups would make such predictions even more accurate.

The hypotheses are that:-

- Users interaction behaviour can be reliably predicted in a significant number of cases.

- The distribution of that behaviour through the task does not arise as part of the normal distribution.

Hence the experiment should show that:-

1. The experimental group display the target behaviour significantly later than the control group
2. That the distribution of the behaviour around the predicted point of occurrence is not normal

The target behaviour involves the use of the 'global search and replace' facility to correct a repeated spelling error.

#### 15.4.2 Method

Ten subjects were required to edit a prepared document on the MACWRITE word processor in two trials. All subjects possessed at least one years experience with the software and they were requested to edit the document as they would normally. All subjects were videotaped. Comments, actions and keystrokes were recorded. No assistance was offered by the experimenter.

Subjects were equipped with the system disc and the text disc and were instructed to boot the system, open the text folder and edit the file named x. The document to be edited consisted of a page of text , selected by the experimenter to be of no semantic relevance to any of the subjects (it consisted of a passage on psychological testing which in itself was meaningless). Subjects were timed for all trials.

The document contained various spelling mistakes of a random variety generated by typing the document quickly and not correcting the mistakes. One word was consistently spelt incorrectly at various places in the document. This word was 'the' which was spelt as 'thwe'.

In the first trial the errors were evenly distributed throughout the passage (two in each paragraph) hence they were fewer in the target paragraph than in the experimental trial. In the second trial. The error was placed once in the first paragraph twice in the second paragraph and ten times in the third and final paragraph. Five subjects took part in the first trial and five in the second trial.

It was predicted that in the second trial (experimental condition) the subject would produce the desired behaviour whilst editing the third paragraph whilst in the first trial there should be no specific pattern to the production of behaviour. The desired behaviour involved the use of the 'global search and replace' facility to correct the repeated spelling error.

### 15.4.3 Results

#### *Resume of Hypotheses*

The hypothesis is that users interaction behaviour can be reliably predicted in relation to the task structure and that the distribution of that behaviour through the task does not arise as part of the normal distribution. Hence the experiment should show that:-

1. The experimental group display the target behaviour significantly later than the control group
2. That the distribution of the behaviour around the predicted point of occurrence is normal



Users exhibited the predicted target behaviour of utilising the 'global search and replace' facility successfully in eighty percent of the cases and unsuccessfully in one other case in the second trial. Even in the case where a subject did not know the procedure for accessing the function use of the command was attempted.

Mean time to display the target behaviour was 45.7 seconds in the experimental condition and 44 seconds in the control condition. This proved to be a significant difference of performance in the second trial as assessed by a student t-test and the null hypothesis was rejected ( $t=0.48$ ,  $df=6$ ,  $p<0.05$ ). The results suggest that the users in the experimental condition were not stimulated to generate the target behaviour due to the presentation of errors.

Data for this experiment is displayed in figure 24 below.

Access →	1st. Paragraph.	2nd Paragraph.		3rd Paragraph.			
Subject	1st.error.	1st.	2nd.	1.	2.	3.	4.
1		38s.					
2		43s.					
3		45s.					
4		-					
5		50s.					
6			39s.				
7			46s.				
8			-				
9			47s.				
10			51s.				
Mean		45.7	44				
SD		4.99	4.97				

**FIGURE 24: ACCESS POINTS AND TIMES IN EXPERIMENT 3**

All of the subjects which exhibited the behaviour did so in the second paragraph (in both trials) with five subjects accessing at the first error and four at the second. Hence, the upper limit for the

invocation of the behaviour was three errors and the lower limit was two errors.

Mean time to display the target behaviour was 45.7 seconds in the experimental condition and 44 seconds in the control condition. Distribution of the point at which the target behaviour occurred was not normal in both trials. This was determined when variance was assessed using an f-test. An f-value was produced which was not significant ( $F=1.1$   $df=3,3$  NS). This was expected in the second trial (due to the positioning of the errors) however it was not expected in the first trial. Production of the behaviour in the second paragraph during the second trial may have been due to the fact that the subject was subjected to two errors in the first paragraph and this may have provoked the production of the behaviour.

#### 15.4.4 Key points from study 3

A number of key results arose from study 3:

- Time to exhibit the predicted behaviour was significantly longer in the experimental condition (ten errors in final paragraph) than in the control condition as assessed by a student t-test
- User behaviour could be said to be significantly affected by the presentation of task stimuli however low n values may have affected this
- The increase in performance time was similar, in response to the stimuli in the experimental condition, when the variation around the mean time was non-significant as assessed by a variance-ratio (f) test

These points are discussed further in the discussion at the end of the chapter.

## 15.5 STUDY 4

### 15.5.1 Introduction

The principles of dynamic HCI postulate that an 'interactive effect' exists, the presence of which exerts an effect on interaction which although not semantically connected to, or in task context with, subsequent events will profoundly affect the performance of the user. Research suggests that users models of complex interactive systems are typically not rational ones and will be based on a pragmatic view of interaction which does not map directly onto the real-world (DiSessa 1986). Hence, users may be subject to effects upon their behaviour based on purely spurious theories

about the operation of the interface. This often fails to pose any specific problem to the user as long as the users assumptions are consistent with the task which is being performed: indeed this is a vital component in ensuring the usability of computers as it means that the user does not need to understand, or possess a literal representation of, the computer or its interface.

Conversely, however the pragmatic or heuristic component of the user model means that the users behaviour is susceptible to a number of spurious cues which the user may react to during the learning of the interface. Naive users are particularly susceptible to any spurious interactive event and will quite readily attribute any system event to their own behaviour. This effect is particularly pronounced where control is lost of the interface and work is lost due to system failures as the cost to the user is quite high (Bailey 1984): hence, the experiment described below explores this specific area. Behaviour following the event is often more cautious and less rapid even when the user is not responsible for the event and this is the basis for the experimental hypothesis.

It is a typical human trait to believe that all events have a specific and identifiable cause and that individual behaviour is often responsible for unexplained events (Gentner and Stevens 1982).

Whilst this is an inaccurate and often misleading model of the world it does mean that the user can

learn from all types of sensory phenomena, and that heuristic strategies can be devised to control the world and maintain consistency.

This testing of such a non-specific phenomena presents a 'worst-case' exploration of the effects of spurious interactive phenomena. If an effect can be displayed in this case then in cases where some link, albeit tentative, between the users behaviour and the interactive event is assumed by the user the effect will be even more pronounced. One of the most widely recognised human factors principles (Smith and Mosier 1986) is that unprompted system events should be kept to a minimum and this work represents a first step in categorising those events to show which phenomena are the most distracting to the user.

The experimental hypothesis is that there will be effect on user behaviour due to a spurious interactive event demonstrated by the fact that time to finish a task after the event will increase when compared with that of a control group within which the event does not take place.

Hence the experiment should show that:

1. Mean time to complete the users task, and associated sub-tasks, will increase significantly in the experimental group subjected to the 'interactive event' (when time to recover system stability is taken into account)

### 15.5.2 Method

Ten subjects were required to edit a prepared document on the MACWRITE word processor in two trials. All subjects possessed at least one years experience with the software and they were requested to edit the document as they would normally. All subjects were videotaped. Comments, actions and keystrokes were recorded. No assistance was offered by the experimenter.

Subjects were equipped with the system disc and the text disc and were instructed to 'boot' the system, open the text folder and edit the file named x. The document to be edited consisted of a page of text, selected by the experimenter to be of no semantic relevance to any of the subjects (it consisted of a passage on psychological testing which in itself was meaningless).

Subjects were provided with instructions on how to edit the document. These included:

1. Creating three new paragraphs
2. Checking the spelling
3. Indenting certain passages of text
4. Changing specified fonts and timesteps
5. Saving the changes

Instructions to the subjects were the same in both trials apart from the item at the start of the list presented to the second five subjects which exploited a natural bug to cause the system to crash: subjects were required to backspace twice to join two paragraphs together and this caused the system to crash. Total task time was measured along with the time taken to perform each of the sub-tasks described above. The time taken to recover from the crash in the latter 5 trials was also recorded.

### 15.5.3 Results

#### *Resume of hypothesis*

The hypothesis states that:

1. Mean time to complete the users task, and associated sub-tasks, will increase significantly in the experimental group subjected to the 'interactive event' (when time to

recover system stability is taken into account)

Total task time was recorded for each subject and the mean task time was calculated. Total task time for the experimental condition group was calculated by subtracting the time taken to recover from the crash from each subjects total to give the time taken to actually perform the task. Times are given in table 25 below:

Subject	Trial 1	Trial 2 (Crash)
1	267s.	473s.
2	313s.	433s.
3	246s.	429s.
4	363s.	390s.
5	427s.	497s.
Mean	323s.	443s.
S.D.	73.4	39.4

**FIGURE 25: TOTAL TASK TIMES (SECONDS) IN EXPERIMENT 4**

Mean task time proved to be significantly longer (443 seconds) in the experimental condition than in the control condition (323 seconds) when the significance of the effect of the interactive event was assessed according to a independent subjects t-test ( $t=3.5$ ,  $df=4$ ,  $p>0.025$ ).

Subject	Trial 1	Trial 2 (Crash)
1	42s.	98s.
2	65s.	86s.
3	49s.	93s.
4	72s.	76s.
5	89s.	99s.
Mean	63s.	90s.
S.D.	18.7	9.6

**FIGURE 26: MEAN SUBTASK TIMES (SECONDS) IN EXPERIMENT 4**

Mean sub-task time proved to be significantly longer (90 seconds) in the experimental condition than in the control condition (63 seconds) when the significance of the effect of the interactive event was assessed according to a independent subjects t-test ( $t = 2.7, df = 4, p > 0.025$ ). Data for this experiment can be found in figure 26 above.

#### 15.5.4 Key points from study 4

A number of key points arose from study 4:

- Time to finish the task was significantly longer in the experimental condition - after the system crash - as assessed by a student t-test
- Mean subtask time was also significantly longer in the experimental condition as assessed by a student t-test
- The system interrupt clearly affected the performance of the users, even when time to

recover was accounted for, however low n-values may have biased the outcome of the experiment

These points are more fully discussed in the next section.

## 15.6 DISCUSSION

This section discusses the results described above and discusses some general conclusions which can be drawn from those results in the context of the experimental designs.

The first study demonstrated some similarity in terms of types of errors across the whole subject population, however it was not possible to comment reliably on errors in particular user groups due to problems with the experimental design.

An informal comparison of command frequency indicated that subjects produced similar numbers of errors, however there was insufficient data to provide a realistic value for the distribution of errors. A t-test revealed a significant difference in the production of UNIX errors due to UNIX experience so there would seem to be some effect due to experience.

Subject numbers were almost certainly too low to reveal any group effects (n =1 from each group) and they were probably too low to facilitate the execution of valid statistical tests. A matched pairs design was used to control for spurious factors in the correlation test although poor sampling and low n. values caused some problems.

Subject numbers were too low to generate sufficient errors to rigorously test the hypothesis: errors were the only explicit behaviour which could be recorded, however since the number of errors made by each subject was limited it was not possible to gain a full picture of the pattern of error-behaviour. A larger subject population would ameliorate this problem.



Correlation tests were able to examine the group influence more readily than the t-test as it was the pattern of usage and not the frequency which was really being tested. Having said that it must be recognised that the behaviour produced may not have been normalised due to lack of data and whilst the correlation value was significant other spurious factors, outlined below, could have been involved.

Subject sampling was not sufficiently thorough in this experiment. Subjects were classified according to UNIX experience and according to their background (first degree), however other factors could have influenced the experimental results. Also the lack of a control group, again due to lack of availability of subjects, has serious implications for effect the validity of the results. It is impossible to say how a totally inexperienced group would have performed compared to the existing groups, however the correlation between members of different groups and members of the same group allowed some control over extra-group effects.

The study does, however, suggest the existence of a holistic effect at some level and supported the hypothesis of interference from previous experience. This implies that interference from other experience does exist and that it should be accounted for in the design process. The second hypothesis was supported to the extent that the correlation showed some group effect, although the experimental design limits the conclusions which can be drawn from this. This is disappointing as it is through the specification of error types for specific groups that this work finds its major application. It is possible, however that subsequent experiments, with larger, better controlled samples, would demonstrate an effect which would allow use to identify the particular types of errors inherent in given user groups.

Further work may be able to assess and describe the particular effects which exist and allow formal models to be created which would enable designers to address a target user group and account for and exploit their existing experience. This work would enable a record of perceptually similar

partial and whole machines to be formed to standardise descriptions in terms of user perception and would allow the application this type of research in future design work.

The second study presented some evidence that 'regression' behaviour can be attributed to unique user groups.

The second study is of a similar design to the first study and hence is subject to the criticisms addressed at that study, however it is worth making a number of specific comments. Statistically similar displays of errors and regression were obtained as a correlation of 'between' and 'within' group behaviours however the conclusions which can be drawn from these results are limited due to the problems with the experimental design.

Assessment of the occurrence of errors, or regression, as a ratio of the frequency of total command use showed no significant connection between the users total command repertoire and the propensity to commit errors. Hence errors or regression could not be related to any specific group behaviour, in this case, although a larger sample may have allowed significant patterns to emerge.

Again, subject numbers were too low to produce an ideal sample of behaviour although the focus of the experiment was the overall pattern of usage of the commands so the absolute number of commands may not be as significant. Sampling again was limited with subjects being chosen according to first degree and overall computer experience. Inadequacies in these sampling criteria lead to the selection of a non-parametric test (Kendall's  $\tau$ ) to minimise the effects of any non-normal distribution.

This study supports the notion of identifiable user groups as a whole and offers some access to specific user attributes in design. This provides closer and more realistic controls over the user during design. Extension of this type of study would allow categorisation of error and regression behaviour and the construction of more accurate and complete user and interaction models for:

- specific named user groups
- larger numbers of users

This activity would form the basis of any replication work.

The third study demonstrated a significant difference in the behaviour of the experimental group as against that of the control group, as assessed by a student t-test. The first hypothesis, which is that a users within the experimental group will display the predicted interaction behaviour at a later stage is clearly supported by the data. This suggests that the positioning of the errors within the text influenced the subjects tendency to display the target behaviour.

The target behaviour was, however, displayed in the second paragraph by all subjects. This was expected in the second trial (due to the positioning of the errors) however it was not expected in the first trial. Production of the behaviour in the second paragraph during the second trial may have been due the subject being presented to a given number of (two) errors in the first paragraph rather than due to the distribution of those errors. Distribution of the target behaviour around the predicted point was, however not normal, as assessed by a variance-ratio (f) test which suggests that the task structure had some influence on the user's behaviour.

Despite the failure of the subjects to respond to the experimenters precise predictions the results suggest that assumptions can be made about the likely generation of behaviours in known task contexts. The consistency of the subjects response shows the potential for predicting users mental and behavioural state with knowledge of task and function constraints. Further work is necessary to extend the scope of the work to:

- produce more data to support the hypothesis
- look at other tasks involved with text editing so that further conclusions can be drawn

- extend the method into other areas of HCI

The fourth study demonstrates a significant increase in task and sub-tasks times due to the presence of the random interactive event, as assessed by a student t-test. This result confirmed both parts of the experimental hypothesis (for tasks and subtasks). The total increase in task time may have been attributable solely to recovery from the crashed system, however this was allowed for in the timing procedure. Additionally, the increase in subtask times suggests that the subjects were generally more cautious for each subtask: this suggests a change in attitude on the part of the user.

Use of the control group means that the experimental effect can be examined, however skill differences were not completely eliminated due to a limited sampling procedure. Subjects should have been assessed for their experience in word-processing to control for experience. Alternatively, a repeated-measures design using the same subjects would have eliminated such problems, however it may have lead to practice effects. Further work to replicate these results would adopt these procedures.

This study revealed subjects sensitivity to random interactive events but it does not present any evidence as to the basis of that effect. This experiment does suggest, however that spurious interactive events should be minimised in interactive systems design and that all types of interactive stimuli should be controlled.

## 15.7 CONCLUSIONS

This section presents a summary and general discussion of the results outlined above and describes some general conclusions which can be drawn from the work as a whole. In summary, an analysis of the results presented above suggests that there is some evidence to support the principles underlying dynamic HCI.

Statistical tests did produce significant effects in many cases, however the conclusions which can be drawn in the various experiments are limited due largely to problems with the experimental designs.

These include:-

- Poor sampling
- Poor or absent control groups

Sampling techniques were of limited scope and were based mainly on simple measures of experience. This means that differences in individual behaviour could not be completely eliminated. The experiments themselves often did not call for specific subject types however there was little control over spurious factors. Experimental problems were offset, to some extent by the use of 'repeated measures' or matched 'pairs' designs but the poor sampling meant that pairs were not matched as effectively as they could. As a result conclusions which can be drawn from the results are limited.

Two of the experiments used a true control group (1 and 4) and this enabled the dependent variable to be examined clearly. The two remaining experiments suffered from a lack of a clear control. Experiment three varied the presentation of task stimuli between conditions however a true control group would have used a condition where there were no stimuli. The second experiment used no formal controls however cross-correlation between groups allowed some validation of the experimental conditions.

Overall, an effect was demonstrated in each of the studies described above, however these effects were largely non-directional and merely served to denote the presence of some experimental effect rather than some thing which could be specifically related to real-world dynamic behaviour. This remains a valid justification for carrying out the work, however as a means of supporting the theoretical framework the experiments were, in themselves, insufficient. In conjunction with the existing empirical evidence, however they can be said to add to the body of knowledge within this

subject. This concurs with the stated objective of this work: the experiments complement the existing body of knowledge in key areas to suggest the validity of the framework. Hence, the experiments can be considered to be successful.

The question remains whether this form of experimental study was appropriate for the investigation of the framework. In adopting this approach the author considered the application of observational empirical studies. In retrospect it may have been inappropriate to use formal experimental studies in the analysis of dynamic HCI as it may have been more appropriate to merely identify supporting behaviour for dynamic HCI rather than attempt to formally test it. This is discussed further in the final chapter. These empirical studies do, however, indicate a genuine need for the consideration of dynamic aspects in design: for example, the subjects were profoundly affected in the final experiment.

These experiments represent simple 'pilot' studies of principles within the thesis and as such offer substantial opportunity for further work to describe the effects demonstrated in a more specific sense. These studies merely aim to show that some effect exists due to the experimental variable selected and considerable further work would be required to categorise that effect. Proposals for future work can be found in the following chapter.

**SECTION FOUR: DISCUSSION AND CONCLUSIONS**

## 16. DISCUSSION AND CONCLUSIONS

This chapter discusses the key issues concerned with the empirical work which is presented within this thesis, and draws some conclusions as to its success and relevance. The chapter then discusses the limitations of this work and indicates where alternative methods could have been used. The chapter goes on to describe future work which will be carried out to replicate and develop upon the work described here, with particular reference to external and internal validity. The chapter also summarises the contributions made by this work. Finally, the chapter highlights some of the problems which have arisen during the study and outlines the authors aspirations for the application of the work by others in the field.

### 16.1 DISCUSSION OF THE EMPIRICAL WORK

In the light of the conclusions arising directly from the empirical work, reported at the end of experimental chapters, it is appropriate to indicate some further, more general issues and conclusions which can be highlighted concerning both the empirical work and the thesis as a whole.

The focus on the design perspective inherent in this thesis has been especially selected with a view to making the results of this work useful to the HCI community. A theoretical analysis of dynamic HCI has been avoided, to a large extent, as this would not offer substantial practical assistance in addressing specific problems within the field. Hence, the value offered by the thesis is in terms of a utility in generating practical design tools and in the application of the theory in the solution of concrete design problems. An empirical evaluation of the design tools in the context of a major system development has proved impossible, however it is believed that the evidence offered does lend some validity to the study.

Whilst samples of the user population were smaller than would have been desired, in many cases,



and as a result the amount of activity studied amounted to less than would have been desired, it is believed that a workable sample of user behaviour was obtained. A useful and manipulable kernel of behaviours was captured in the exemplar and this enabled the experimenter to model a number of typical situations, with the kernel validated by user feedback. Finite restrictions on sample size were in operation and this, of course, has important ramifications for the validity of the experimental results and the applicability of the data. It is believed, however, that a *much* larger sample would have been needed to generate statistically sound results and such samples were impractical for a project of this size. It is possible to say, however, that some effects were observed which could reasonably be attributed to dynamic HCI and the literary evidence adds weight to this argument.

Future work is necessary to extend upon the model generated here, however this sample would need to be substantial to ensure the capture of a complete model of behaviour: this would enable accurate, repeated predictions of user behaviour over a number of circumstances via the development of a robust formal model. This study has shown, however, that such models can be formed and that they are useful. Further work is needed to extend the notion to form larger models.

It has proved to be difficult to control for all the possible variance in user behaviour and whilst it is believed that a fairly representative sample, with a near normal distribution, was obtained due to the application of some background and experience controls the sample was not as wide as would have been desired. Controls took the form of repeated measures and matched pairs designs to apply internal experimental control wherever possible however independent control groups were not applied in many cases. Split trials were used, where possible, to eliminate practice and learning effects (particularly relevant to dynamic behaviour). The use of the correlational analysis revealed the significance of the effect in some of the tests and controlled for natural and unnatural variance within and between groups.

The behavioural data recorded in the empirical studies can be extended to include much more than

the interaction device knowledge selected here. This type of behaviour (interaction device knowledge) was chosen because it could easily be modelled from explicit behaviour whereas others may not. There are however many other media for representation available and more powerful behavioural elicitation techniques would, again, allow more sophisticated models to be formed.

The 'post hoc' introduction of the user groups as part of the data analysis meant that they were described in terms of realistic behaviours and ensured that there was no experimenter bias in the collection of the data. Formation of the model as a whole before any analysis was undertaken allowed a comprehensive kernel to be formed and meant that each unique behaviour need only be represented once.

The choice of word-processing activity in the empirical evaluation was successful due to a number of factors: this type of task allowed consistency of task and results between machines and allowed comparison with existing work in the field using this type of task. This is, indeed, the most common type of HCI activity and hence can be seen as a good analogy for HCI as a whole. Selection of the two exemplar machines also proved to be expeditious in that it enabled the experimenter to obtain subjects from a broad range of experience and offered sufficiently diverse behavioural and representations to test the objectivity of the model. The model itself produced a useful and realistic informal assessment of usability and its determinants.

Time-taken for the performance of the exemplar study amounted to some four weeks involved in obtaining subjects, running trials and gaining feedback. The time-taken to analyse all the results and form the generic models amounted five and a half-weeks. This would seem to make this a useful developmental tool within the timescale of a research project. Rapid iteration is necessary in most projects and the variable sample size can help to determine this and the validity of the model. The example evaluation in this study managed to produce a useful model on a rather small sample.

In summary, the empirical work has not produced totally conclusive results in terms of internal or

external validity as there is a need for some replication work with larger sample sizes. The results do, however, meet the stated goals of the work in that they show:

- the utility of the design tool
- some evidence to complement the existing work in the literature in supporting the theoretical framework for dynamic HCI

The design tool has been shown to offer a sensitive and accurate method for collecting data, an accurate and statistically valid model of the users behaviour and a method for generating performance predictions which are consistent with known usability values. In this way it been shown to be successful. Future work to extend the internal and external validity of the tool, which is described later, will allow greater conclusions to be drawn on the application of the tool in a wider design setting.

The experimental work for the validation of the theoretical framework has not provided a 'proof' that dynamic HCI exists or is relevant. This goal has, however, been rejected as being impractical within the scope of this, or even a much larger, study. It is not useful to attempt to prove the *existence* of dynamic HCI. What the experimental work does show, however, is that there is evidence to suggest the existence of dynamic influences on user behaviour for the principles which are not well-supported in the literature. This adds further evidence to the body of work which exists in the literature however the identification of such dynamic effects remains on ongoing research task. Future work envisaged within this area will attempt to generate further empirical data to support the theoretical framework from both formal and informal studies.

## 16.2 LIMITATIONS OF THE WORK

The work described within this thesis is, of course, subject to the limitations based of finite resources available, the restrictions imposed by the selection of given methodologies and the time

available to complete the project. It is important to recognise these limitations so that further work can build on the results obtained here and not be constrained by its limitations.

### 16.2.1 External Validity

External validity refers to the value or utility of a piece of work to the domain or population within which it is employed. The question which must therefore be asked of this piece of work is: 'is it useful in allowing the creation of more usable interfaces, by means of the design tools it seeks to generate?'

The design tool described within this thesis suffers from limited external validity primarily because of the lack of a complete design study in the evaluation of the tool. The true test of the evaluation work can only be found in the longitudinal application of the tool in a number of design settings with experimental controls to test for the effectiveness of the tool. The opportunity to perform this type of evaluation often depends heavily on the ready availability of a suitable software development project in which the personnel who are to use the tool are well-motivated. This has proved impractical in this case. This is true, however, of many other well accepted design tools: Card, Moran and Newell's GOMS model, for instance, was only properly evaluated in subsequent work and a number of deficiencies were highlighted (Roberts and Moran 1984). This does not, however, obviate the need for a more thorough empirical assessment of this work in a genuine design setting. The authors proposals for carrying out such work are presented in later sections.

An alternative assessment of the utility or value of a piece of work can be obtained via expert opinion. Expert opinion concerning a new piece of work is typically obtained, in the first instance, via the publication of the work and its presentation at conferences or seminars. It is here where the work can be more directly criticised and defended and direct feedback can be obtained from knowledgeable parties. To date there are no published papers arising from this work however a number of publications are planned and these are described in more detail in later sections. Expert

opinion in this case will be provided by the Phd examiners and this form of evaluation is, in essence, the basis of the Phd examination. The examiners should be aided, however by the clear presentation of the issues within the thesis to make clear that the author is aware of the criteria which determine external validity.

A further criterion upon which external validity can be assessed is that of the impact of the work upon the HCI community and their practises. Proof of this cannot be offered, of course until it has been offered to the community as a whole, however the author believes that the work will encourage those working within HCI to consider dynamic factors in their modelling and development work and will influence the design, operation and interpretation of prototyping work as part of the system development process.

Comparison of this tool with other current methodologies as a form of evaluation presents some difficulties. No current formal techniques aim to address the area targetted by this work therefore direct comparison is difficult. Indirect comparison with structured tools looking even at user issues is again of doubtful value as the key focus of this project is 'dynamic issues': this is not unusual as the work described here needs to be unique and significantly different from other current techniques to merit the qualification towards which it is directed.

It is often difficult to offer a clear measure of external validity within a Phd project. The theoretical framework of dynamic Human-Computer Interaction is difficult to assess in terms of external validity as it aims to offer a new perspective from which to approach evaluation. The novelty of the work, and consequent limitations on its empirical evaluation, mean that it is difficult to place the work within the field of HCI as a whole. Credibility of the work in the HCI community and in its application environment are essential in establishing its validity, however many tools only receive recognition after a long period of exposure.

### 16.2.2 Methodology Problems

The evaluation of the hypotheses and design tools extended within this thesis posed some quite significant problems. Selection of methodologies for the assessment of new evaluation techniques is usually determined by the methodologies which have been used to perform such evaluation in previous studies. This posed some difficulties for a number of reasons:

- The lack of a standard form of assessment for new evaluation techniques
- The novelty of the work and difficulties in comparing it with other techniques

With hindsight a rather different form of analysis from the one selected could have been chosen. In selecting the methods used for both the evaluation and the experiments the author was aiming to demonstrate some theoretical tractability for both the design tool and the principles of dynamic Human-Computer Interaction. The most appropriate test of the theory would have been the application of theory and the tools in a design context and as this proved to be impossible a statistically sound method was opted for. What may have been more useful is the exploration of a number of potential design tools in a rather less rigorous fashion to demonstrate the application of the theory and to provide the HCI community with a range of novel techniques which may have proved useful. Also the execution of a number of empirical studies to demonstrate and observe dynamic phenomena, rather than attempting to experimentally prove that such phenomena exist, may have been more appropriate as many of the experiments explored tautologous principles and did not comment on their actual form in any way.

Selection of experimental and statistical methods was made to lend the thesis some measure of credibility where a less formal perhaps more design oriented approach may have been more suitable. What would have proved useful would have been an exploration of the role of a number of design tools in the software development cycle to examine how data is gathered, how the design tool is used to fashion the prototype followed by a final measure of the quality of an interface

arising from the tool (perhaps using attitude scales). It could be said that a Phd based on design does not always need to provide rigorous proofs to be useful to the HCI community and this informal approach is discussed later this chapter.

Hence, specific limitations have arisen for each component of the analytical work:

*The theory of dynamic HCI:* The area of study proved to be so complex and diverse that it was necessary to reduce the problem-space to a number of discrete principles which were gleaned from the literature. These principles can only represent a poor approximation to what is a very rich area and hence the experimental work based upon these principles showed corresponding limitations. Furthermore, the reduction of the problem-space in such a drastic manner meant that some of the experiments appeared to test trivial phenomena. It is only within the context of the assumptions made concerning the validity of the principles to be tested that the experiments are more valid.

In addition to the problems encountered above the experimental design showed some weakness in many cases. Small sample numbers lead to a number of problems:

- control groups could not be provided
- sampling was not random in all cases

These are typical problems within this type of study however the results of the experiments must be viewed with appropriate caution. It cannot be shown that experimental effects were due to the stated variable in some cases and there is scope for some interference in the experimental effect due to similarities between subjects. The above problems largely derive from small numbers of subjects and where a particular subject attribute was required sampling problems were exacerbated. Control problems were to some extent ameliorated by the contrasting of 'within-groups correlations' with 'between-groups correlations' and variance tests for subject similarity, however this is no substitute for the allocation of an independent control group. Subjects were also classified

according to total computer experience as well as, for example, UNIX experience however this additional experience could not be shown to be irrelevant to the purposes of the experiment.

*Evaluation of the design tool:* The major methodological problem concerning the evaluation of the design tool was the lack of availability of a suitable software project to carry out an empirical assessment of the utility of the tool with designers. This meant that a novel method was needed to evaluate the tool which addressed all of the relevant attributes of the tool in a rigorous manner. Whilst the method provided useful results from a range of tests it could not assess the validity of the tool in a true design context and hence could not comment on external validity. It is also difficult to draw comparisons between the results of this work and comparable studies which have been undertaken in the past and this again points to the need for further work to extend the external validity of the work.

Focussing on one design tool also meant that it was not possible to explore the utility of design tools based on a number of the principles within the framework. This approach provides a rigorous but narrow body of support for the framework as it was thought that this was preferable to the provision of a shallower but broader body of support. This implies that it would be useful to explore a number of alternative design tools to ascertain whether they would be of equivalent utility within design.

### 16.2.3 Objective evaluation scales

The approaches described above are reflected in the methods used to carry out the empirical work. The evaluation of the theoretical dynamic interaction framework is based on experimental studies using specific hypotheses. The evaluation of the design tool and the application towards which it is targeted are based on correlation and variance studies to form a statistical model of the users groups behaviour. This model is then used to comment on the usability of the two interfaces.



The dynamic interaction framework represents a methodology in the sense that it offers a theoretical pattern within which range of design and evaluation tools could be developed. It is not a structured methodology in the current sense of the term, in the sense that it offers a prescriptive guidelines for the application of a set of existing tools, rather it helps identify the dynamic issues within Human-Computer Interaction for which evaluation tools can be developed in the future.

Finally, the framework offers the scope for the development of metrics, measurements scales and design or evaluation tools dependent upon the nature of the problem to be addressed but it is not in itself an evaluation or design tool. Chapter 13 offers a number of types of dynamic properties which applications may possess and a set of principles that those applications should display. This outlines both the relevant types of application and the hypothetical principles to be tested in the development of future design tools.

### 16.3 THE NEXT STEP

In the light of the above limitations a range of future activities are planned in order to develop the work and in particular to enhance its internal and external validity : these activities involve further empirical assessments of the design tools, the development of novel design tools and additional observational and analytical work to allow the development of the theory to account for new HCI phenomena.

First, it is essential to offer some more concrete evidence of external validity. The credibility of this work within the HCI community can only be ensured through a genuine empirical evaluation. The design tool will be evaluated by asking a number of software designers to use it in their work. Feedback on the utility of the tool will be obtained by asking the designers to report on how the tool has improved their work in comparison to an equivalent previous project. The author is aware of a number of ongoing projects which present a requirement for such a modelling tool. By encouraging designers to utilise the tool described within this work it will be possible to assess the

overall benefits of the tool over and above their current methods.

Ideally, the tool should be evaluated by asking one group of designers to use it whilst another control group of similarly skilled designers performed the same task without the tool. The differences in the quality of the software produced would give a very accurate assessment of the validity of the tool. It is difficult to envisage such opportunity for carrying out a controlled, comparative, empirical evaluation, however by asking designers to report their experience of the tool it should be possible to obtain some useful objective and subjective feedback on its effectiveness.

Further exploration of the role of the dynamic HCI framework in generating prototypes is essential. The role of the theory in design is seen to be the provision of an interim representation between the static textual notation and the dynamic prototype. Hence, this is an important area for future work. This work would involve the use of the theory in the development of design tools to assist in the construction of prototypes: the author is currently engaged in the production of a number of prototypes and hence is in position to undertake this first-hand. The tools currently being developed take two forms: text-based action models of interaction outlining the range of physical actions and interactions of the user to outline the repertoire of the user. This includes features of the users environment which can impinge upon the users interaction style. Second, guidelines are being developed for the use of constructive simulation techniques and the use of very low-cost simulations of interaction. Feedback on the usefulness of these techniques would offer another source of evidence of the external validity of the work. This activity is again an ongoing task and it is hoped that a wide range of tools can be made available, via the application of the theoretical framework, within the near future.

Further study is also necessary to ensure the internal validity of the work. The empirical work described within this thesis provides some evidence for the validity of the approach, however there is a need to examine the experimental effects under more robust conditions. Hence, further work

will be carried out to attempt to replicate the results here with: larger subject numbers, more careful population sampling and more rigorous experimental designs. The use of larger subject numbers will allow the capture of a larger sample of data and will allow any experimental effect to be more clearly demonstrated. More thorough population sampling will allow specific user groups to be examined and will ensure a closer control over skill differences due to individual differences. Finally, a reworking of the design of some of the experiments to include true control groups to ensure that the experimental condition is valid; to control further for subject differences using matched pairs designs; and to control for practice effects by using split trials and repeated measures designs will be undertaken.

Ongoing observation and analyses of dynamic phenomena in HCI will be carried out with a view to validating the dynamic principles and attributes and identifying additional principles. The experimental work described within this thesis provides some evidence to support the theory however it is believed that an observational approach can also provide support for the notion of dynamic HCI. The observational approach provides examples of real-world HCI and design problems which need to be addressed by the theory (Young 1986) and it is believed that the theory can be extended in this way. As long as the framework can account for novel dynamic behaviours and phenomena and allow them to be employed within design tools then it remains useful. The theory will also be revised to account for new or important phenomena which arise within HCI design. For instance, a practical application of the calibration scales can be seen in alleviating synchronisation problems between video and audio media in 'multi-media' systems by developing a scaling tool which allows the designer to capture the users objective temporal requirements for the presentation of media. The design tool will also be extended to allow the modelling of requirements for Computer Supported Co-operative Working (CSCW) systems. A generic model of the task requirements for the CSCW task will be formed using the techniques described within this thesis. The group analysis of the generic knowledge kernel will then be undertaken based on the requirements of different groups of users on different sites. The results of this work will be presented in future papers.

Finally, the theory and tools will be revised based on comments on the completed thesis and any subsequent papers which may arise from the work. It is hoped that the theory can be refined through criticisms of the written output of the project.

#### **16.4 THE PRACTICAL APPLICATION OF THIS WORK**

This section provides a resume of the key points made earlier in the thesis to summarise the practical contributions and applications of this tool.

The principal objectives of this work are:

1. To provide a practical framework for the generation of design tools for the integration of dynamic features of HCI into the design of computer interfaces
2. To demonstrate such a design tool and hence offer support to the notion of dynamic HCI

Thus, the framework for dynamic HCI essentially contributes in two ways to HCI:

1. It allows the generation of design tools which bridge the gap between static description and prototype - action models of interaction which help designers build prototypes
2. It provides a simple framework for the analysis and understanding of complex HCI behaviours to allow them to be integrated into the design process

In accounting for these two aspects of HCI it is believed that this work contributes significantly to HCI design practice.

## 16.5 ADDITIONAL FEATURES OFFERED BY THIS THESIS

A number of features included within this thesis offer a source of novel material for the advancement of the field of HCI, interactive systems design and usability measurement. These are in terms of theoretical aspects and practical design and specification tools.

Firstly, the three-dimensional model of interactive behaviour is postulated which represents interaction in terms of a number of layers, a number of dimensions within those layers and as a task behaviour on a temporal task-path within a given dimension and layer. Behaviour within this model can be represented in terms of a number of media, such as dialogue or information and hence this model is very flexible and can capture some of the dynamic features in many application environments. User behaviour along the task-path can be described as a series of discrete interaction states in which the behavioural repertoire of the user remains relatively constant. Hence, short-term behavioural strings can be inserted reliably into this model and can be said to be valid for that state. Three states identified in the exemplar evaluation are naive, novice and expert.

Secondly, new behavioural data extraction techniques are proposed and implemented in an object-based, generic format and these are highlighted as the fundamental basis for a sound HCI study. These new techniques offer the opportunity to identify the discrete and well-integrated strings of behaviour which are inherent in HCI. Such modular or object-based units are not chronologically defined and hence describe realistic interactive events from the perspective of the user. Hence, these units can be recombined to form a realistic approximate model of interaction which focusses on the significant levels of interaction for the user and describes the changes in interactive behaviour which are of most use in a dynamic analysis. Both abstract and working machines can be described using these techniques. A tool is offered for the recombination of user group descriptions to form more general models. Termed the 'Group Overlap Tool' it offers the potential to generate general, reliable and realistic models of behavioural commonalities between

groups under close experimental controls.

Thirdly, temporal calibration scales can be described as a means of representing absolute and relative temporal features during interaction and this represents an objective framework for the representation of dynamic issues in interaction. This is seen as a possible method for implementing adaptive systems. A standard task-path description framework is offered as a general basis for outlining the constraints on behaviour according to position in interaction. This forms part of the framework for the construction of long-term temporal descriptions from the short-term predictive bursts on the calibration scale.

A number of grains-of-analysis are proposed for the analysis of interaction at a variety of significant levels. These levels are:-

- The task level
- The group level

These levels enable models of behaviour to be formed which facilitate: the statistical analysis of group repertoires of interaction behaviour and the design of interfaces to reflect these repertoires, and the analysis of discrete strings of behaviour, again for group usage, but more importantly to determine where they occur in terms along the continuum of user experience and therefore to allow models of users of differing experience to be formed.

Interaction device knowledge has been identified as a useful medium for creating user descriptions which correlates well with the users knowledge of the functionality of the system and hence offers a practical task and functionally oriented representation which can be used by designers (after Kieras and Polson 1985). This enables generic knowledge about interaction to be standardised and described. This also offers a source of reliable objective and behaviourally explicit data about interaction.

The adoption of formally described user groups has been suggested as the basis for the inclusion of real-world behaviour at a useful level of analysis and as a set of valid controls for user behaviour and experience in experimental design. This concept has been implemented in terms of generic user behaviour in the exemplar evaluation and hence can be utilised in similar formats elsewhere.

The existence of dynamic information processing models is extended to reflect the dynamic nature of perception and their nature and relevance to interactive systems design is discussed. Perception is taken to be 'active' and the users employment of knowledge in determining interaction behaviour is discussed. The further use of cognitive maps is supported for the representation of cognitive states and activity and for the descriptions of a cognitive interaction model which can be used in design. Novel uses of map structures are suggested to optimise user memory. Procedural memory is identified as an implicit source of information and information patterns are seen as a means for expressing context.

A number of general discrete knowledge states are identified which have been demonstrated to be applicable across a range of machines and these have been quantified in terms of experience and task knowledge. A formal definition of dynamic HCI is proposed in terms of eleven salient properties and these are experimentally tested. An extension to the current method of dynamic evaluation has been proposed in terms of 'context' and 'experience'. A standard system specification cycle has been described and implemented and has produced concrete result. Finally, a number of experimental studies have been undertaken which have yielded interesting and significant results which would be of use to the HCI community.

The thesis provides a critical assessment of current metrics and suggests means by which these can be improved, including their combination to give a more comprehensive assessment and their quantification to allow the introduction of real-world data. A broader base of criteria are suggested and this is advocated as the only means of effecting complete evaluation. Comparison along

individual criteria would allow a more precise comparison.

A novel multi-layer model is offered (Watkinson 1986) which presents an operational and procedural view of interaction and seeks to describe machine concepts in more natural terms. This model is intended to stress a functional and task based description and hence could be more useful in interactive systems design. Dynamic user models are suggested as a means of incorporating real-world user behaviour into descriptions and as a means of quantifying layer models. Heuristic specification techniques are suggested as a practical means of describing behaviour in situations where identifiable behavioural constraints exist. A formal heuristic classification and categorisation system is offered to assist in implementing such models.

The use of taxonomic description structures is proposed for the representation of user knowledge and information structures due to the compatibility they display with human knowledge and data structures. The role of perspective in such models has been identified as a means of expressing the inherent stylistic concerns which exist in the behaviour of given user groups.

A number of formal usability definitions have been offered which offer novel and design oriented perspectives on the definition of usability. This is congruent with the advocated approach to usability assessment which forms a major tenet of this thesis and enables dynamic situations to be modelled. A formal definition of 'Cognitive Style' is offered with suggestions as to how this feature can be accounted for in interactive systems design. This is an attempt to exploit the natural variation inherent in user behaviour which enables them to interpret task descriptions in an optimal fashion.

The notion of 'Functional Ergonomics' is added to that of 'Cognitive Ergonomics' to highlight the key role that accessibility and task issues play in the determination of usability. A functional approach is highlighted as a whole as interaction is seen as purposeful (task-driven behaviour) and functional descriptions are seen as being task and implementation transparent which enables a



variety of situations to be modelled. Methods have been offered for the quantification of visibility and accessibility and the application of these concepts in the design process has been discussed.

Some of the above features have been formally justified and all of them are supported in the literature, however they all constitute original and novel work which could be utilised to facilitate progress in the HCI field.

## **16.6 PROBLEMS WITH TIMESCALES**

One of the difficulties met in the management of the development of a usability project, as it is with all research projects, is the choice of an optimal time limit within which all research goals should be met. Limits are often practical - grant-length - or technology constrained -when the new piece of equipment turns up - or determined by the success of other research - if they get there first, we give up. Notwithstanding such factors an essential process which must be undertaken at some time is the constraint of the possible field of study. Possible strategies or targets must be rapidly and savagely reduced to a finite set of specific and definable goals which does not compromise on the overall concept of the research. The top-level goals of the work should be supported by the methods developed, not determined by them. Hence, choice of methods is often made on a pragmatic basis which is determined by the circumstances of the work.

### **16.6.1 Problems with the timescale of this work**

The field of dynamic HCI evaluation is equally susceptible to the above difficulties, if not more so. The concept of dynamic evaluation or even one of its premises could form a project by itself. It is unlikely that substantive proofs could therefore be offered - as they so rarely are in Human-Computer Interaction and Artificial Intelligence - within this timescale for the notion of dynamic evaluation and its viability as a theoretical framework. Hence, proofs of dynamic interaction tend to

show the presence of a dynamic effect but do not quantify or qualify it. This is a 'design' project and thus it suffices to show that the consideration of dynamic factors has some value in the design of computer systems rather than providing an objective value of the performance of a given interface.

Some of the material offered in support of the concept of dynamic HCI is literary and this has left more time to research the viability of the theoretical framework within HCI which, it is believed, offers much more practical assistance to the field. It is clear that much more experimental work would be needed to offer substantive proof that dynamic HCI exists but it is believed that a greater contribution would be offered by the 'non-provable' theoretical framework. This Phd, in the opinion of the author, need not necessarily be about formally 'provable' things, however it should contribute some useful 'knowledge' to the field. The concept of dynamic HCI is amenable to 'proof' in the context of a much smaller project, however the results would not have proved to be as useful to the HCI community .

## **16.7 THE PRACTICAL USES OF THIS TECHNIQUE**

The ultimate question which must be asked of this work is what will it be used for and why? This can be re-phrased as what does it do and why is it important?

It is acknowledged that the design tool developed here may never be used again in the form in which it is presented in this thesis, however it is hoped that the theoretical framework suggested by this work exerts some influence in the design of other studies and offers a concrete conceptual schema for the consideration of HCI. It is hoped that by embracing the notion of dynamic HCI researchers will be encouraged to take a broader perspective on evaluation and perform more comprehensive and realistic analyses. This work is important as it focusses coherently on a whole new aspect of Human-Computer Interaction and it looks at the use of computers in a novel way. It is only by adopting such radically different stances to the analysis of Human-Computer Interaction

that the types of quantum leap in usability achieved by the highly successful 'direct manipulation' systems can be achieved. The development of such interfaces did not wait until they were formally proved to be viable (Smith 1983) they merely developed the system and waited for the world to catch up. The work described above, however, aims to codify the key requirements for the application of a new theoretical framework in a way which is both coherent and usable. The author hopes that systems designers can recognise the importance of dynamic factors and begin to address them in their day-to-day systems development work.

**SECTION 5: APPENDICES**

**APPENDICES**

**1. REFERENCES**

- ALEXANDER H. (1986): The formal specification of Human-Computer Interaction. UNIVERSITY OF STRATHCLYDE Technical Report.
- ARGYLE M. (1977): Doctor-patient communication. IN - MEDICAL ISSUES IN SOCIAL PSYCHOLOGY Pelican.
- ANASTASI A. (1982): Psychological Testing. Collier Macmillan
- BADDELEY A.D. (1976): The psychology of memory. Basic books.
- BAILEY R.W. (1982): Human Error in computer systems. Addison-Wesley.
- BARNARD P. (1988): Approximate modelling of cognitive activity with an expert system: A theory-based strategy for developing an interactive design tool. THE COMPUTER JOURNAL, VOL. 31, NO. 5.
- BERRY, D.C. and BROADBENT, D.E. (1987): Expert systems and the Man-Machine Interface. EXPERT SYSTEMS, Vol.4, No.1.
- BEWLEY W.I., ROBERTS T.L., SCHROIT D. AND VERPLANKW.L.(1983): Human Factors testing in the design of the XEROX 8010 "Star" office workstation. In Janda A. (ed.) CHI '83 Proceedings ACM, New York.

BOLT R.A. (1985): The Human interface. MIT Press.

BOWER G.H. and SPRINGSTON F. (1970): Pauses as recoding points in letter sequences. JOURNAL OF EXPERIMENTAL PSYCHOLOGY, 226.

BROOKS R. (1977): Towards a theory of the cognitive processes in computer programming. THE INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES 9.

BRUNER J.S. (1955): Learning the mother tongue. HUMAN NATURE, 262.

BRUNER J.S. (1974): Beyond the information given. George, Allan and Unwin.

BUXTON W. (1986): There's more to Interaction than meets the eye - Some issues in manual input. IN - USER CENTERED SYSTEM DESIGN eds. Norman D.A. and Draper S.W., Lawrence Erlbaum.

CARD S.K., ENGLISH W.K. and BURR B.J. (1978): Evaluation of mouse, rate controlled isometric joystick and task keys for selection on a CRT. ERGONOMICS 21.

CARD S.K., MORAN T.P. and NEWELL A. (1982): The psychology of human- computer interaction. Lawrence Erlbaum.

CARROLL J.M. (1982): Learning, using and designing filenames and command paradigms. BEHAVIOR AND INFORMATION TECHNOLOGY 1.

CARTER J.A. (1986): A taxonomy of user oriented functions. THE INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES 24.

CHAPANIS A. (1974): Interactive human communication; Some lessons learned from laboratory experiments. IN: Human-computer interaction; Human factors aspects of computers and people (ed. Shackel).

CHAPANIS A. (1981): Evaluating Ease of Use. IN - PROCEEDINGS OF IBM SOFTWARE AND INFORMATION USABILITY SYMPOSIUM.

CHRISTIE E. (1985) (ed): Information technology and the electronic office. Springer Verlag.

CHRISTIE B. (1986): Human Factors in Information technology. Cambridge.

CLARKE A.A. (1986): A three level human computer interface model. THE INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES 20.

CERF V. (1973): "PARRY" encounters the "DOCTOR". IN - DATAMATION Vol 3 No.4.

COOPER M. and HOCKLEY A. (1985): The evaluation of adaptive interfaces. IN - IEEE COLLOQUIUM ON ADAPTIVE INTERFACES.

CRAIK F.I.M. and WATKINS M.J. (1973): The role of rehearsal in Short Term memory. JOURNAL OF VERBAL LEARNING AND VERBAL BEHAVIOUR, 244.



CYPHER A. (1986): The structure of users activities. IN - USER CENTERED SYSTEM DESIGN eds. Norman D.A. and Draper S.W. Lawrence Erlbaum.

DICKERSON K. (1988): Embedded user models. Proceedings of BCS HCI Specialist Group Meeting on User Models.

DISESSA A. A.(1986): Models of computation. IN - USER CENTERED SYSTEM DESIGN eds. Norman D.A. and Draper S.W., Lawrence Erlbaum.

DOOB E. (1953): Time patterning. Macmillan.

DUBOULAY G.M., O'SHEA T. and MONK J. (1982): Presenting computer concepts to novices. IN: THE INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES 14.

EASON K.D. (1977): A study of manager computer interaction (Phd thesis, LUT ).

EDMONDS E.A.(1986): Beyond Computable numbers. Proceedings of the Inaugural Lecture LUT.

EDMONDS E.A. (1981): Adaptive Man-Computer Interfaces. IN - Computing skills and the user interface, editors Coombs M.J. and Alty J.L., Horwood Ellis.

EDMONDS E.A. (1988): Some comments on evaluation. PROCEEDINGS OF L.U.T.C.H.I. WORKSHOP.

EVANS C.R. (1974): Improving communication between people and computers.

IN - HUMAN COMPUTER INTERACTION; HUMAN FACTORS ASPECTS OF COMPUTERS AND PEOPLE ed. Shackel B..

FAENRICH J. and ZIEGLER A.D. (1986): An evaluation of Direct manipulation workstations.

IN - PROCEEDINGS OF SIGGRAPH Springer Verlag.

FITTS, P. (1986): Human performance. Greenwood Press.

FOLEY J.D. and VANDAM A. (1982): The fundamentals of Interactive Computer Graphics

Addison Wesley.

FOWLER C., MACAULAY L., CASTELL A. and HUTT A. (1989): A evaluation of the usability

of a Human Factors based Requirements Capture methodology. PROCEEDINGS OF HCI '89 -

People and Computers 5, Sutcliffe A.and Macaulay L. (eds).

GAINES B. and SHAW M. (1984): The art of computer conversation. Ellis Horwood.

GARDNER A., MAYFIELD T. and MAGUIRE M. (1984): Human factors guidelines for

computer based systems. INTERACT 84 (Conference) Ed. Shackel B.

GENTNER D. and STEVENS A.L. (1983): Mental models. Hillsdale.

GILB T. (1976): Software metrics. Winthrop.

- GREEN T.R.G., PAYNE S.J. and VAN DER VEER G.C. (1983): The psychology of computer use. Academic Press.
- GREENBERG S. and WITTEN I.H. (1982): Adaptive personal interfaces.  
IN: BEHAVIOUR AND INFORMATION TECHNOLOGY 1.
- GRUDIN J. and BARNARD P.J. (1984): The role of prior task experience in command name abbreviation. INTERACT '84(Conference) Ed. Shackel B.
- HART A. (1986): Knowledge acquisition for expert systems. Kogan Page.
- HAYES -ROTH F., WATERMAN D.A. and LENAT D.B. (1986) : Building Expert systems Addison Wesley.
- HEBB D.O. (1972): Textbook of psychology. Saunders.
- HENDERSON D.A.(1986): The Trillium User Interface design environment.  
In - PROCEEDINGS OF CHI '86.
- HITCH G.J. , SUTCLIFFE A.G. , BOWERS J.M. and ECCLES L.M. (1986): Empirical evaluation of map Interfaces: A preliminary study. PEOPLE AND COMPUTERS: PROCEEDINGS OF THE SECOND CONFERENCE HCI SPECIALIST GROUP eds. Johnson P and Cook s. Cambridge University press.
- HUBEL D.H. and WIESEL T.N. (1964): Receptive fields and functional architecture of monkey striate cortex. JOURNAL OF PHYSIOLOGY, 42.

HUTCHINS E.L. , HOLLAN J.D. and NORMAN D.A. (1986): Direct Manipulation Interfaces. IN - USER CENTERED SYSTEM DESIGN eds. Norman D.A. and Draper S.W., Lawrence Erlbaum.

INNOCENT P., PLUMMER D., TEATHER D., MORTON B.A., WILLS K.M. and DUBOULAY G.M. (1984): The design and implementation of flexible interfaces : A case study. INTERACT '84 (Conference) Ed. Shackel B.

JACOB R.J.K. (1986): Survey and examples of specification techniques for user computer interfaces. Draft Report, NRL, Washington.

JOHNSON P.(1984): Task skills and knowledge: task analysis for knowledge descriptions. IN - INTERACT '84 ; Springer Verlag.

JOHNSON P. (1986): TKS - Representing user knowledge. ICL Technical Journal, No.8.

KENNEDY T.C.S. (1982): Some behavioural factors affecting the training of naive users in an interactive computer system. INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES 7.

KIDD A. (1984): Knowledge elicitation for expert systems. Academic Press.

KIDD A. (1986): Man-machine dialogue design. British Telecom Research Report.

KIERAS D. and POLSON P.G. (1985): An approach to the formal analysis of User complexity. INTERNATIONAL JOURNAL OF MAN -MACHINE STUDIES 22.

LANSDALE M. (1988): Personal information systems: Some design issues. PROCEEDINGS OF L.U.T.C.H.I. WORKSHOP.

LENOROVITZ S.R., PHILLIPS R.S., ARDREY R.S. AND KLOSTER G.V. (1984):

A taxonomic approach to characterising Human-computer Interfaces.

In - HUMAN-COMPUTER INTERACTION; Advances in Human Factors/Ergonomics Conference. Ed. G Salvendy.

LEVISON W.H. (1979): A model for mental workload in tasks requiring continuous information processing. In - MORAY N. (1979): Mental Workload: It's theory and Measurement. Plenum.

LINDQUIST S. (1984): Assessing the usability of the Human Computer interface. IEEE SOFTWARE 2(11).

LINDSAY H. and NORMAN D.(1982): Human Information Processing. Harper and Row.

LOO, R. (1978): Individual differences and the perception of traffic signs.

HUMAN FACTORS Vol.20, 65-74.

McCAULEY M.E. (1984): Human factors in voice technology. IN- HUMAN FACTORS

REVIEW ed. Frederick A..

MALONE T.W.(1982): Heuristics for designing enjoyable user interfaces.

IN: HUMAN FACTORS IN COMPUTER SYSTEMS '82 (Conference) Gaithersburg.

MALONE T.W. (1985): Designing organisational Interfaces.

PROCEEDINGS OF C.H.I. '85. New York, ACM.

MAYER R.E.(1981): The psychological study of how novices understand computer programming.

COMPUTING SURVEYS 13 (1).

MILLER L.A. and THOMAS J.C. (1977): Behavioural issues in the use of interactive systems.

INTERNATIONAL JOURNAL OF MAN- MACHINE STUDIES (9).

MILLER G. (1956): The magical number Seven Plus or Minus Two: Some limits on our capacity for processing information. PSYCHOLOGICAL REVIEW 63.

MILLER R.B.(1974): Human ease of use criteria and their tradeoffs. IBM TECHNICAL REPORT 00.2185 POUGHKEEPSIE.

MIYATA N. (1984): Constructive Interaction. COGNITIVE SCIENCE 10.

MIYATA Y. and NORMAN D.A. (1986): Psychological issues in the support of multiple activities. IN - USER CENTERED SYSTEM DESIGN eds. Norman D.A. and Draper S.W., Lawrence Erlbaum.

MORAN T.P. (1981) : The command language grammar. INTERNATIONAL JOURNAL OF MAN- MACHINE STUDIES 15.

- MORAN T.P. (1984): Getting into a system: External Internal Task mapping analysis. IN - HUMAN-COMPUTER INTERACTION, Advances in Human Factors and Ergonomics Conference. Ed. G Salvendy.
- MORAY N. (1979): Mental Workload: It's theory and Measurement. Plenum.
- NEISSER .U (1967): Cognitive Psychology. Appleton Century Crofts.
- NEILSEN A.J. (1983): A virtual protocol model of Human-Computer Interaction. BEHAVIOUR AND INFORMATION TECHNOLOGY 2.
- NELSON .T (1980): Interactive systems and the design of virtuality. IN - CREATIVE COMPUTING 6.
- NEWELL, A. and SIMON, R. (1972): Human problem solving.
- NEWELL, A: (1965): Production systems: Models of controls structures. IN- VISUAL INFORMATION PROCESSING Academic Press.
- NEWMAN J. (1984): Command interfaces and the hierarchy of needs. INTERACT '84 (Conference) Ed. Shackel B.
- NORMAN D.A. (1976): Memory and attention: An introduction to Human Information Processing, Wiley.
- NORMAN M. (1988): Comments on user models, PROCEEDINGS OF L.U.T.C.I.I.

## WORKSHOP.

O'MALLEY C. (1986): Helping users help themselves. IN - USER CENTERED SYSTEM DESIGN eds. Norman D.A. and Draper S.W., Lawrence Erlbaum.

ORNSTEIN R. (1977): The perception of time. Pelican.

OWEN D.B.(1984): Users in the real world. INTERACT '84 (Conference) Ed. Shackel B.

PARNAS A. (1981): Finite state grammars. THE INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES 18.

POULSON D. (1987): Towards simple measures of the perceived quality of software interfaces. HUSAT Draft Memo, HUSAT, Loughborough University of Technology, Loughborough.

POSTMAN L. (1971): The present status in interference theory. IN - VERBAL LEARNING AND VERBAL BEHAVIOUR ed.Cofer C.N, 231.

REISNER P. (1981): Formal grammar and human factors design of an interactive system. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING 7 (229).

REISNER P.(1985): Use of psychological experimentation as an aid to the development of a query language. IEEE SOFTWARE 1.

REISNER P. (1984): Formal grammar as a tool for analysing ease of use: Some formal concepts. IN - HUMAN FACTORS IN COMPUTER SYSTEMS ed Thomas J.C.



- REISNER P. (1984): Further developments towards using formal grammar as a design tool. IN - HUMAN FACTORS IN COMPUTER SYSTEMS ed Janda A.
- RHEINHART W. and MARKEN R. (1985): Control systems analysis of computer pointing devices. IN - PROCEEDINGS OF THE HUMAN FACTORS SOCIETY 29.
- RICH E.A. (1983): Users are individuals: Individualising user models. INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES 18.
- ROBERTS T.L. and MORAN T.P. (1984): Evaluation of Text-Editors: Methodology and empirical results. COMMUNICATIONS OF THE ACM 26.
- ROBERTSON I.T. (1978): Individual differences in information processing strategy and style. IN - PROCEEDINGS OF IEE CONFERENCE ON MAN MACHINE SYSTEMS.
- ROSKE-HOFSTRAND, R.J. and PAAP, K.R. (1986): Cognitive networks as a guide to menu organisation: An application in the automated cockpit. ERGONOMICS Vol.29 No.11, 1301-1311.
- RUMELHART D.E. and NORMAN D. (1982): Simulating a skilled typist: A study of skilled cognitive motor performance. COGNITIVE SCIENCE 6.
- RUMELHART D.E. and NORMAN D. (1982): Human Information Processing, Ellis Horwood.

RUTKOWSKI C. (1982): An introduction to the human applications standard Computer Interface. BYTE 7.

SCHAFFER G. (1988): Functional analysis of office requirements. Wiley.

SCHANK R. (1982): Dynamic Memory. Cambridge University Press.

SCHMANDT C. (1985): Voice communication with computers. IN - ADVANCES IN HUMAN COMPUTER INTERACTION ed. Hartson H.R., Ablex.

SELF J. (1984): Student models in CAI. INTERNATIONAL JOURNAL OF MAN -MACHINE STUDIES, 6.

SELFIDGE R and NEISSER U. (1960): Pattern recognition by machine. SCIENTIFIC AMERICAN 203.

SHACKEL B. (1982): The concept of usability. IN: VISUAL DISPLAY TERMINALS Ed. BENNETT.

SHINAR D. and STERN H.I. (1985): The relative effectiveness of alternate selection strategies in menu driven computer programs. IN - PROCEEDINGS OF THE HUMAN FACTORS SOCIETY 29.

SHNEIDERMAN B. (1980): Software Psychology. Winthrop.

SHNEIDERMAN B. (1986): Designing the User Interface. Addison Wesley.

- SHNEIDERMAN, B. (1979): Multi Party Grammars. **INTERNATIONAL JOURNAL OF MAN - MACHINE STUDIES 9.**
- SKINNER B.F. (1932): The behaviour of organisms. Appleton Century Crofts.
- SMITH R. (1982): Designing the XEROX STAR. **BYTE 19.**
- SMOLENSKY P. (1984): formalising task descriptions for command specification and documentation. **IN - INTERACT '84, Springer Verlag.**
- SOKOLOV E.N. (1967): Learning and memory: Habituation as negative learning. **IN - NEURAL MECHANISMS OF LEARNING AND MEMORY eds Rosenzweig M.R and Bennet E.L. Cambridge.**
- STEVENS G.C. (1983): User-friendly computer systems: A critical examination of the concept. **BEHAVIOUR AND INFORMATION TECHNOLOGY 2.**
- SUTCLIFFE A.G. (1985): Use of conceptual maps as Human Computer Interfaces. **IN - PEOPLE AND COMPUTERS: DESIGNING THE INTERFACE eds. Johnson P and Cook S., Cambridge University press..**
- TALBOT M. (1985): Speech technology: Is it working? **People and computers: Proceedings of the conference of the BCS Human-Computer Interaction Specialist Group.**
- THIMBLEBY H. (1985): Evaluating User Interfaces, University of York.

- THOM R. (1982): Catastrophe Theory. IN - Catastrophe Theory: Denis Postle, Fontana.
- VAN DER VEER G.C., TAUBER M.J., WAERN Y. and VAN MUYLWIJK B. (1985): On the interaction between system and user characteristics. BEHAVIOUR and INFORMATION TECHNOLOGY Vol. 4, No.4.
- VON BEKESE J.A. (1973): An investigation into the neural cochlear impulses of guinea pigs JOURNAL OF NEUROSCIENCE 34.
- WATKINSON N.S. (1986): End of year research report, L.U.T.C.H.I. INTERNAL REPORT LUT.
- WEISENBAUM J. (1976): Computer power and human reasoning. Pelican.
- WHITEFIELD A. and LONG J. (1989): Cognitive ergonomics and the user interface. Wiley.
- WICKLEGREN W.A. (1956): The long and short of memory. PSYCHOLOGICAL BULLETIN, 245.
- YOUNG R.M. (1985): User models as design tools for software engineers. ALVEY WORKSHOP ON MMI/SE.
- YOUNG R.M. (1981): The machine inside the machine : users models of pocket calculators. THE INTERNATIONAL JOURNAL OF MAN - MACHINE STUDIES 15.

YOUNG R.M. (1986): Turbocharging the tortoise of cumulative science. Proceedings of CHI '86.

**2. BIBLIOGRAPHY**

**ALLEN R.B. (1982): Cognitive factors in Human Interaction with Computers. BEHAVIOUR AND INFORMATION TECHNOLOGY 1.**

**ARBLASTER A. (1980): Human factors in the use of computer languages. INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES 1..**

**BALLANTINE R.M.(1979): Psychology, Ergonomics and the HCI. IN- USER FRIENDLY SYSTEMS Infotech state of the art.**

**BARNARD P.J. (1984): User representations of ordered sequences of command operations. IN - INTERACT '84 Springer Verlag.**

**BARNARD P.J.(1979): Consistency and compatability In Human-Computer dialogue. INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES 15.**

**BLESER T. and FOLEY J. (1982): Towards specifying the Human factors of User-Computer Interfaces. IN - HUMAN FACTORS IN COMPUTER SYSTEMS ed. Thomas J.C.**

**BENNETT C.A. (1982): Toward empirical, practicable, comprehensive task economy. HUMAN FACTORS 13.**

**BENYON S. and INNOCENT P. (1985): Experiments in adaptive Interfaces. IN- IEEE COLLOQUIUM ON ADAPTIVE MAN-COMPUTER INTERFACES.**

BOIES S.J. (1981): User behaviour on an interactive computer system. IBM SYSTEMS JOURNAL 13.

BOLT R.A. (1985): The Human Interface M.I.T. press.

CARROLL J.M. (1982): Learning ,using and designing filenames and command paradigms. BEHAVIOUR AND INFORMATION TECHNOLOGY 1.

COOPER P.A. (1985): Adaptive intelligent dialogues project -Phase one summary. IN- IEEE COLLOQUIUM ON ADAPTIVE MAN-COMPUTER INTERFACES.

EASON K.D. (1982): Towards the experimental study of usability. BEHAVIOUR AND INFORMATION TECHNOLOGY 3.

FITTER M. (1977) : Towards more natural interactive systems. INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES 11.

GREEN T.R.G. (1979): Programming as a cognitive activity. IN- HUMAN INTERACTION WITH COMPUTERS ed. Green T.R.G. Academic Press.

HECKEL P. (1983): Walt Disney and User oriented software. BYTE 8.

INNOCENT P.R. (1980): Towards self-adaptive systems. INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES 16.

KENNEDY T.C.S. (1977): The design of interactive procedures for man machine interaction. INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES 6.

LINDQUIST T. (1985): Assessing the usability of Human-Computer Interaction. IEEE SOFTWARE 12.

LAM S.T. and GREENSTEIN J.S. (1984 ) : Input medium and task allocation strategy. IN - INTERACT '84 ed. Shackel.

McCAULEY L.A. and NORMAN M.A. (1984 ): Designing Interfaces for different types of users. IN - INTERACT '84 ed. Shackel.

MASKERY H.S. (1984): Adaptive interfaces for naive users. IN - INTERACT '84 ed. Shackel.

MAYER R.E. (1979): The psychological study of how novices understand computer programming. COMPUTING SURVEYS 13.

MONK A. (1984): How and when to collect behavioural data. IN -FUNDAMENTALS OF HCI ed.Monk A. Academic Press.

MORAN T.P. (1979): An applied psychology of the User. COMPUTING SURVEYS 13.

NICKERSON R.S. (1969): Man -Computer Interaction. ERGONOMICS 12.

NORMAN D.A. (1984 ): Four stages of User activity. IN - INTERACT '84 ed. Shackel.



- PAYNE S.J. (1984 ):Task action grammars. IN - INTERACT '84 ed. Shackel.
- QUINN L. and RUSSELL M. (1986): Intelligent Interfaces:User models and planners. IN - PROCEEDINGS OF THE CONFERENCE ON CHI ed.Duboulay B.
- ROSENBERG J.K. and MORAN T.P. (1984 ): Generic commands. IN - INTERACT '84 ed. Shackel.
- SASSO R (1984 ).:Personalising the User Interface. IN - INTERACT '84 ed. Shackel.
- SCHNEIER C.A. and MEHAL M.E. (1984): Evaluating the usability of application interfaces. IN- HCI : ADVANCE IN HUMAN FACTORS AND ERGONOMICS (CONFERENCE) ed. Salvendy G.
- SHACKEL B. (1981): The concept of Usability. IN- VISUAL DISPLAY TERMINALS ed. Shackel Addison Wesley.
- SHNEIDERMAN B. (1986): Seven plus or minus two: Central issues in HCI. PROCEEDINGS OF THE CONFERENCE ON CHI ed.Duboulay B.
- SHNEIDERMAN B. (1976): Measuring Computer program quality and comprehension. INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES 9..
- TAUBER M.T.(1987): Top down design of Human-Computer Interfaces (Draught paper).
- TOMESKI E.A.and LAZARUS R. (1979): People oriented computer systems. Harcourt Brace

Jovanovich.

TREU S.:Assessment of Interactive systems in terms of mental work. INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES 7 (1978).

WATKINSON N.S. (1986): A five-layer model of Human-Computer Interaction. LUTCHI Phd. Research Report.

WEINBERG G.M. (1981) : The psychology of computer programming. Academic Press.

YOUNG R.M. and GAMMACK J.G. (1984): Psychological techniques for eliciting expert knowledge. IN -BCS CONFERENCE ON EXPERT SYSTEMS.

### 3. Knowledge Kernels from exemplar evaluation

Users knowledge of interaction behaviour was described as sets of action/object pairs which described interaction in terms of generic interface concepts and the actions which could be performed upon them. Hence commands could be abstracted their syntactic form and compared between machines in a meaningful fashion.

#### a. Total Kernel

The first kernel represents the total sample of the users knowledge of interaction behaviour from all the trials of both machines. These pairs are classified in functional groups for clarity:

POWER-UP MACHINE

RESTART MACHINE

INTERRUPT MACHINE

SHUT-DOWN MACHINE

LOGIN-TO MACHINE

CONFIGURE MACHINE

CONFIGURE KEYBOARD

SELECT DISC/DRIVE

ACCESS DISC/DRIVE

ERASE DISC/DRIVE

INITIALISE DISC/DRIVE

COPY DISC/DRIVE

LIST DISC/DRIVE

CHECK DISC/DRIVE

MOVE DISC/DRIVE

CHANGE CLOCK

CHANGE DATE

CHANGE TERMINAL SETINGS

CHANGE KEY SETTINGS

CHANGE PASSWORD

CREATE FILE

DELETE FILE

MOVE FILE

OPEN FILE

CLOSE FILE

SAVE FILE

SAVE-ASFILE

NAME FILE

RENAME FILE

CHANGE FILE

LIST FILE

PRINT FILE

SELECT FILE

COPY FILE

COMPARE FILE

FIND FILE

SORT FILE

JOIN FILE

RECOVERFILE

CREATE DIRECTORY

DELETE DIRECTORY

LIST DIRECTORY

CHANGE DIRECTORY

OPEN DIRECTORY

MOVE DIRECTORY

SELECT DIRECTORY

NAME DIRECTORY

RENAME DIRECTORY

PRINT DIRECTORY

CHANGE DIRECTORY

COPY DIRECTORY

IDENTIFY DIRECTORY

OPEN WINDOW

CLOSE WINDOW

RESIZE WINDOW

MOVE WINDOW

SELECT WINDOW

PRINT WINDOW

SCROLL WINDOW

MOVE-UP FILE TREE

MOVE-DOWN FILE TREE

MOVE-ACROSS FILE TREE

MOVE-TO-TOP-OF FILE TREE

CREATE FILE TREE

RUN APPLICATION

QUIT APPLICATION

MOVE-TO APPLICATION

SELECT APPLICATION

READ EMAIL

SAVE EMAIL

SEND EMAIL

CREATE EMAIL

EDIT EMAIL

INITIALISE PRINTER

SELECT PRINTER

CHECK PRINTER

CREATE TEXT

DELETE TEXT

CUT TEXT

PASTE TEXT

SELECT TEXT

FIND TEXT

CHANGETEXT

SCROLL TEXT

MOVE TEXT

FORMAT TEXT

NAME TEXT

SIZE TEXT

PRINT TEXT

STYLE TEXT

PLACE TEXT

COUNT TEXT

CHECK TEXT

RECOVERTEXT

FORMAT . PAGE

NUMBER PAGE

END PAGE

FIND PAGE

CREATE HEADER/FOOTER

DELETE HEADER/FOOTER

EDIT HEADER/FOOTER

FIND HELP

QUIT HELP

PLACE CURSOR

MOVE CURSOR

SET CURSOR

PRINT SCREEN

SET ALARM

1. MACINTOSH (Macwrite)

POWER-UP MACHINE

RESTART MACHINE

INTERRUPT MACHINE

SHUT-DOWN MACHINE

CONFIGURE MACHINE

CONFIGURE KEYBOARD

SELECT DISC/DRIVE

ACCESS DISC/DRIVE

ERASE DISC/DRIVE

INITIALISE DISC/DRIVE

COPY DISC/DRIVE

LIST DISC/DRIVE



CHECK DISC/DRIVE

MOVE DISC/DRIVE

CHANGE CLOCK

CHANGE DATE

CHANGE TERMINAL SETINGS

CHANGE KEY SETTINGS

CREATE FILE

DELETE FILE

MOVE FILE

OPEN FILE

CLOSE FILE

SAVE FILE

SAVE-ASFILE

NAME FILE

RENAME FILE

CHANGE FILE

LIST FILE

PRINT FILE

SELECT FILE

COPY FILE

FIND FILE

RECOVERFILE

CREATE DIRECTORY  
 DELETE DIRECTORY  
 LIST DIRECTORY  
 CHANGE DIRECTORY  
 OPEN DIRECTORY  
 MOVE DIRECTORY  
 SELECT DIRECTORY  
 NAME DIRECTORY  
 RENAME DIRECTORY  
 COPY DIRECTORY  
 IDENTIFY DIRECTORY

OPEN WINDOW  
 CLOSE WINDOW  
 RESIZE WINDOW  
 MOVE WINDOW  
 SELECT WINDOW  
 PRINT WINDOW  
 SCROLL WINDOW

MOVE-UP FILE TREE  
 MOVE-DOWN FILE TREE  
 MOVE-ACROSS FILE TREE  
 MOVE-TO-TOP OF FILE TREE  
 CREATE FILE TREE

RUN APPLICATION  
QUIT APPLICATION  
MOVE-TO APPLICATION  
SELECT APPLICATION

INITIALISE PRINTER  
SELECT PRINTER  
CHECK PRINTER

CREATE TEXT  
DELETE TEXT  
CUT TEXT  
PASTE TEXT  
SELECT TEXT  
FIND TEXT  
CHANGETEXT  
SCROLL TEXT  
MOVE TEXT  
FORMAT TEXT  
NAME TEXT  
SIZE TEXT  
PRINT TEXT  
PLACE TEXT  
CHECK TEXT  
RECOVERTEXT

FORMAT PAGE

NUMBER PAGE

END PAGE

FIND PAGE

CREATE HEADER/FOOTER

DELETE HEADER/FOOTER

EDIT HEADER/FOOTER

FIND HELP

QUIT HELP

PLACE CURSOR

MOVE CURSOR

SET CURSOR

PRINT SCREEN

SET ALARM

## 2. UNIX (Vi)

POWER-UP MACHINE

RESTART MACHINE

INTERRUPT MACHINE  
SHUT-DOWN MACHINE  
LOGIN-TO MACHINE  
CONFIGURE MACHINE

SELECT DISC/DRIVE  
ACCESS DISC/DRIVE  
COPY DISC/DRIVE  
LIST DISC/DRIVE  
CHECK DISC/DRIVE  
MOVE DISC/DRIVE

CHANGE CLOCK  
CHANGE DATE  
CHANGE TERMINAL SETINGS  
CHANGE KEY SETTINGS  
CHANGE PASSWORD

CREATE FILE  
DELETE FILE  
MOVE FILE  
OPEN FILE  
CLOSE FILE  
SAVE FILE  
SAVE-ASFILE

NAME FILE

RENAME FILE

CHANGE FILE

LIST FILE

PRINT FILE

SELECT FILE

COPY FILE

COMPARE FILE

FIND FILE

SORT FILE

JOIN FILE

CREATE DIRECTORY

DELETE DIRECTORY

LIST DIRECTORY

CHANGE DIRECTORY

OPEN DIRECTORY

MOVE DIRECTORY

SELECT DIRECTORY

NAME DIRECTORY

RENAME DIRECTORY

PRINT DIRECTORY

COPY DIRECTORY

IDENTIFY DIRECTORY

MOVE-UP FILE TREE  
MOVE-DOWN FILE TREE  
MOVE-ACROSS FILE TREE  
MOVE-TO-TOP OF FILE TREE  
CREATE FILE TREE

RUN APPLICATION  
QUIT APPLICATION  
MOVE-TO APPLICATION  
SELECT APPLICATION

READ EMAIL  
SAVE EMAIL  
SEND EMAIL  
CREATE EMAIL  
EDIT EMAIL

INITIALISE PRINTER  
SELECT PRINTER

CREATE TEXT  
DELETE TEXT  
FIND TEXT  
CHANGETEXT  
SCROLL TEXT

MOVE TEXT

FORMAT TEXT

NAME TEXT

SIZE TEXT

PRINT TEXT

STYLE TEXT

PLACE TEXT

COUNT TEXT

CHECK TEXT

RECOVERTEXT

FORMAT PAGE

NUMBER PAGE

END PAGE

FIND PAGE

FIND HELP

QUIT HELP

PLACE CURSOR

MOVE CURSOR

PRINT SCREEN

SET ALARM



b. A list of generic objects

TEXT (as in a word-processor)

PAGE ( " )

HEADERS FOOTERS ( " )

HELP (On-line documentation)

CURSOR (Text and command language)

SCREEN (Contents of the display)

ALARM ( System clock or timed processes)

MACHINE (The system)

KEYBOARD (The virtual character set)

DISC/DRIVE (The device, or medium, and its contents)

CLOCK (the visible clock or the implicit system timeframe)

DATE ( " Calendar " )

TERMINAL SETTINGS (for example, VT100)

KEY SETTINGS (the allocation of any special function keys)

PASSWORD (any security or access code)

FILE (a document or collection of text )

DIRECTORY (a collection of files, in the Macintosh context a folder)

WINDOW ( a subsection of the screen with title bar, scroll bar, boxes and arrows)

FILE TREE (the implicit or explicit hierarchical collection of files)

APPLICATION (any software device distinct from of the system software)

EMAIL (any mailing or messaging system)

PRINTER (a device for producing hardcopy of documents created on the system)

c. A list of generic actions

SET

PRINT

MOVE - TO

- ACROSS

- UP

- DOWN

- TO-TOP-OF

PLACE

FIND

RUN

QUIT

EDIT

DELETE

CREATE

FORMAT

NUMBER

END (terminate)

RECOVER

CHECK

COUNT

STYLE (apply style to)

CONFIGURE

RESTART

INTERRUPT

LOGIN-TO

SHUTDOWN

OPEN

CLOSE

IDENTIFY

LIST

SORT

JOIN (concatenate)

COMPARE

COPY

SAVE

SAVE-AS (store under a new name)

SIZE

RESIZE

NAME

SCROLL

SELECT (choose)

CHANGE

CUT

PASTE

INITIALISE

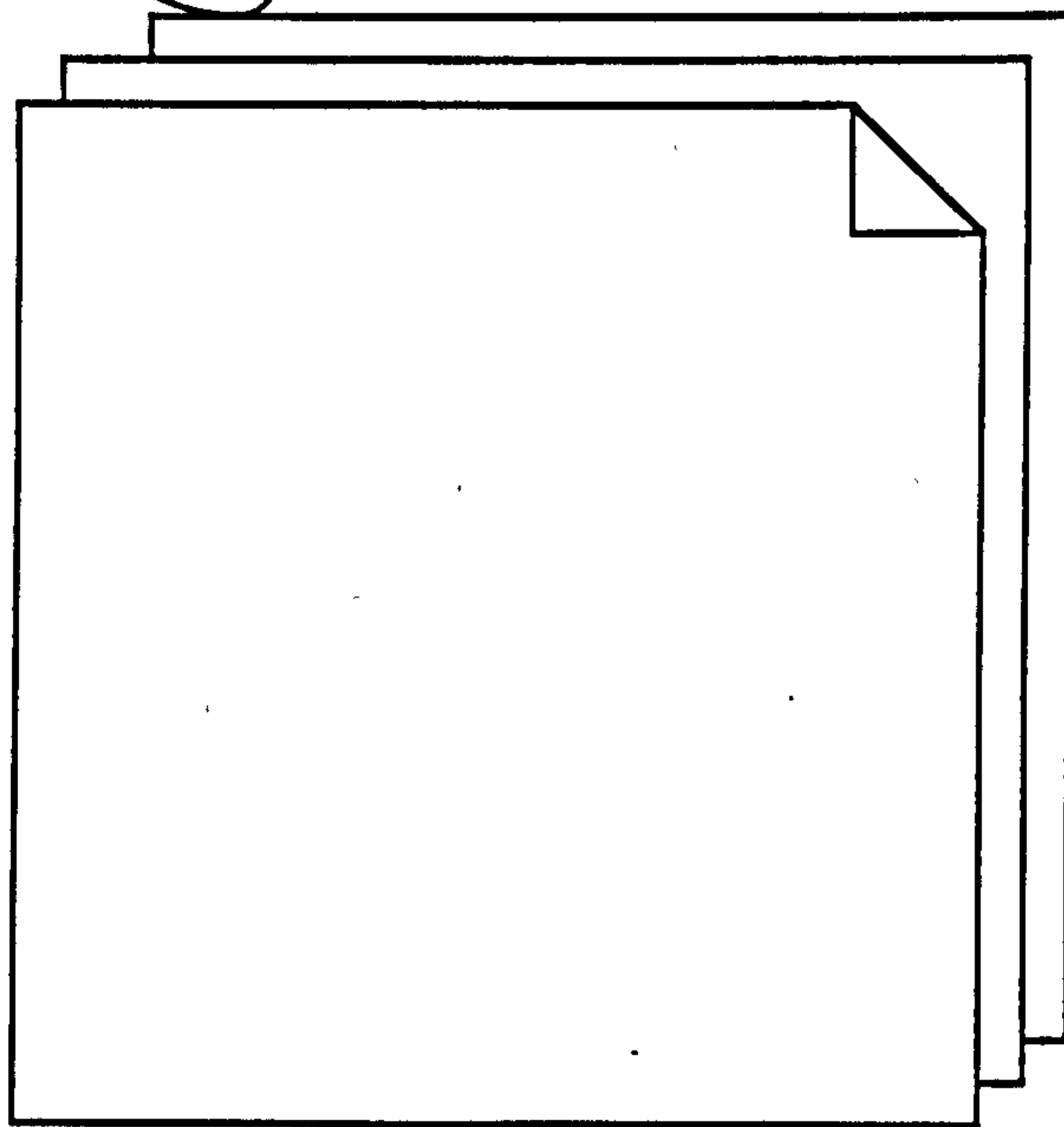
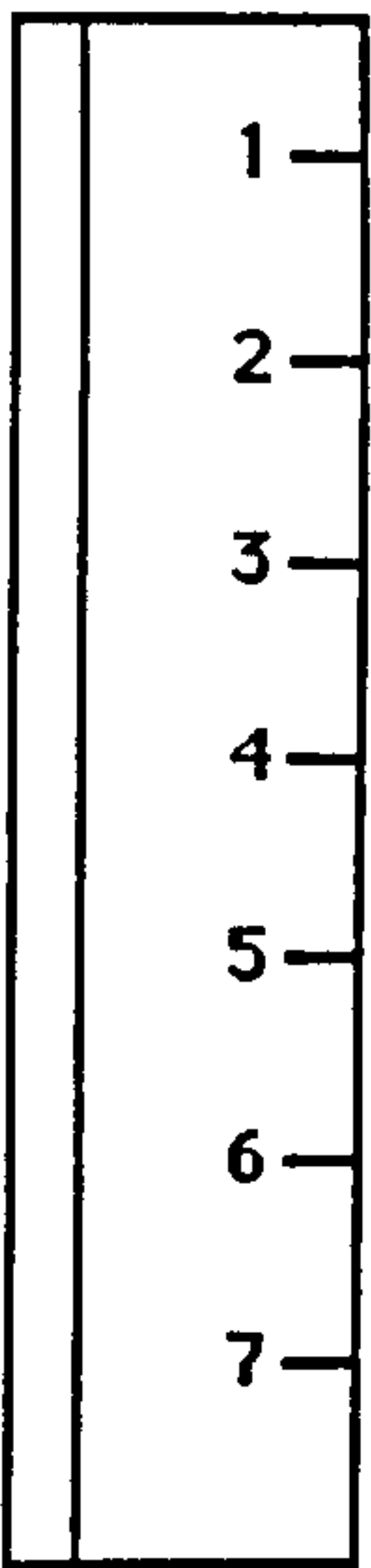
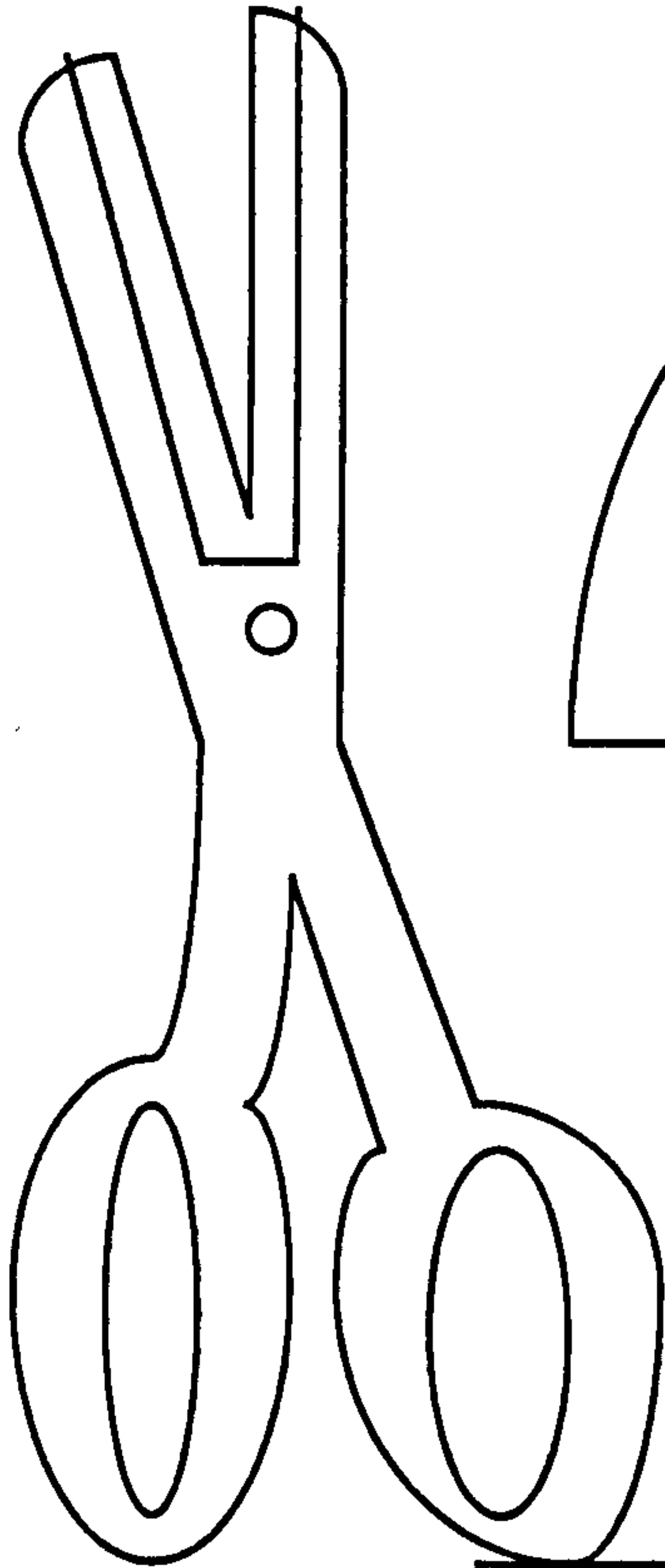
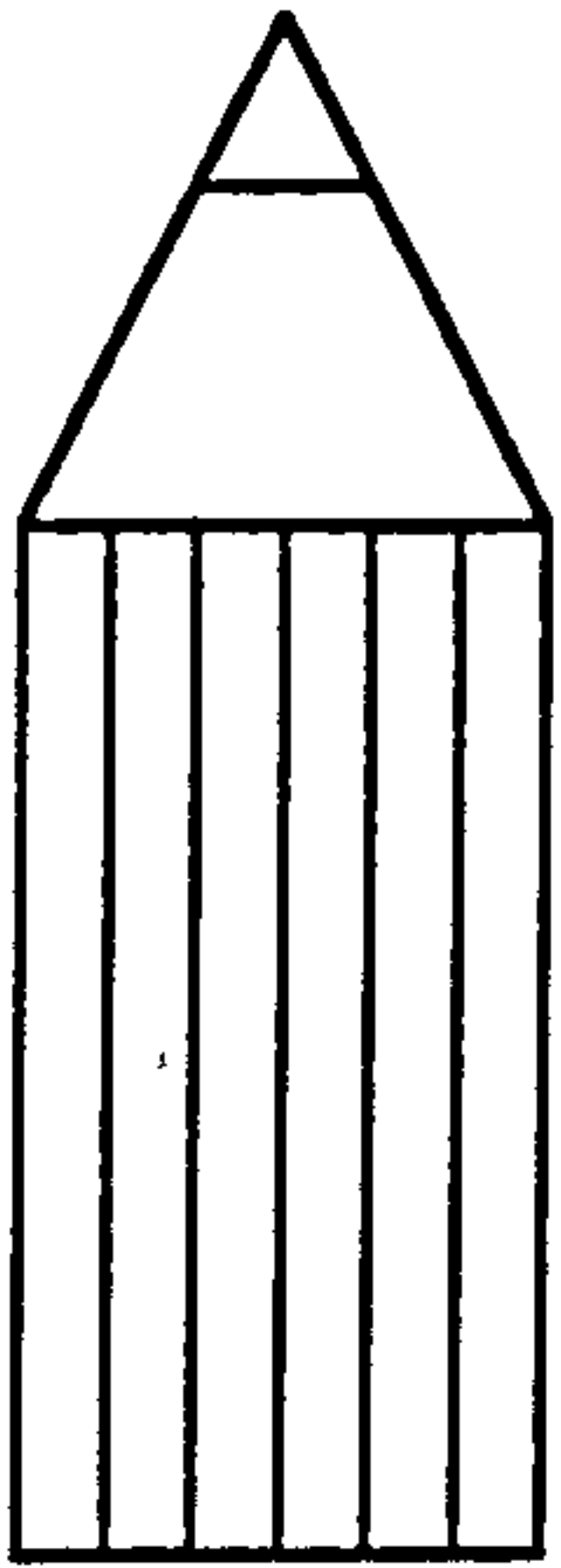
READ (message/file)

#### 4. Proposed system description

The proposed system which was evaluated during the exemplar consisted of a number of generic virtual objects and a number of generic operations which could be performed upon them. Included within the system were such objects as a pencil, an eraser, pieces of paper, headers and margins a glue pot and a pair of scissors. All of these were contained within a frame which also represented a virtual object. The final virtual object was a cursor and a paper simulation of a pointing device (a mouse).

The subjects were able to use the cursor and the mouse to invoke a number of operations on the virtual objects. These operations included cut, write, stick, push and squash. Hence, the scissors could cut paper, the headers could push paper and the margins could squash paper. Subjects were given paper simulations of all of the above objects and presented with text handling tasks to perform using the objects and the list operations. Feedback was offered by the experimenter in some cases. Subjects performed their task and reported on the type of results they expected. The integrity displayed by these objects proved to be very useful in this sense.

Objects used are shown on the next page:-



Virtual objects used for the Simulated Word-Processor

## 5. Structured Interviews

### a. In the identification of commands.

Question 1. What command did you use here?

Question 2. What was the purpose of the use of this command?

Question 3. Did you attempt to use command x. here?

Question 4. Were you aware of command y? (If an alternative exists).

Question 5. If so, why did you not use it?

### b. In the identification of Users actual methods.

Question 1. How would you perform this task? (Showing subjects a written task description).

Question 2. Would this be a good description of how you would perform this task? (Showing subject the predicted sequence according to the general kernel.)

Question 3. What changes would you make to make this description more like your method?

### c. For the identification of User Groups.

Question 1. How would you perform this task? (Showing subjects a written task description).

Question 2. Would this be a good description of how you would perform this task? (Showing subject the predicted sequence according to the Users groups method.)

Question 3. What changes would you make to make this description more like your method? 5.

**6. Glossary of abbreviations**

HCI - Human Computer Interaction

HCIF - Human Computer Interface

HCC - Human Computer Communication

HIP - Human Information Processing

GBIP/GIP -Generic Behavioural Interaction Primitives

PM - Procedural Memory

