

Open Research Online

The Open University's repository of research publications and other research outputs

Database systems: different interfaces for different users

Thesis

How to cite:

Jennings, Frances (1991). Database systems: different interfaces for different users. MPhil thesis. The Open University.

For guidance on citations see [FAQs](#).

© 1990 The Author

Version: Version of Record

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's [data policy](#) on reuse of materials please consult the policies page.

oro.open.ac.uk

DATABASE SYSTEMS: DIFFERENT INTERFACES FOR DIFFERENT USERS

Frances Jennings, MA

Submission for MPhil

December 1990

Discipline: Human-Computer Interaction

Date of submission: 17th December 1990

Date of award: 5th April 1991

ProQuest Number: 27758393

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent on the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 27758393

Published by ProQuest LLC (2019). Copyright of the Dissertation is held by the Author.

All Rights Reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

Abstract

Users of database systems, especially public information systems, can vary widely in their aptitudes and experience with using computer systems. This study investigates whether users with different aptitudes and experience are suited to database system interfaces with different dialogue styles.

Experiments were conducted to examine the performance of users on interfaces with different dialogue styles. The results suggested that dialogue styles which aid navigation through the database, and which constrain the dialogue, are suitable for users with both a low spatial ability and a low experience of using command language style interfaces. However dialogue styles which offer little assistance with navigation, and which allow an open and flexible dialogue, are suitable for users with a high spatial ability whatever their experience, and for users with a low spatial ability but high experience of using command language style interfaces.

The suggestion that different users of a database system are suited to interfaces with different dialogue styles, raises the issue of how to present different users with different interfaces. The study investigates the possibility of producing an adaptive database system, which automatically provides each user with the interface which suits them.

A demonstration adaptive version of one of the databases used in the experimental work was developed. The system was designed to present an interface which aided navigation, and constrained the dialogue, to users with low spatial ability and low actual and potential experience of using

command style interfaces. The system was designed to present an interface allowing an open and flexible dialogue to users with high spatial ability, or low spatial ability but high actual or potential experience of using command style interfaces. The adaptive system was constructed with the aid of a prototype 'adaptive system shell', designed to provide a generic architecture for the mechanisms necessary for an adaptive system.

Acknowledgement

I should like to thank David Benyon for supervising this thesis, and the National Physical Laboratory who funded the work.

Contents

Acknowledgement	ii
List of Figures	iv
List of Tables	vi
1. Introduction	1
2. Experiment One	
2.1 Introduction	11
2.2 Method	16
2.3 Results and Discussion	23
2.4 Conclusions	36
3. Experiment Two	
3.1 Introduction	39
3.2 Method	40
3.3 Results and Discussion	46
3.4 Conclusions	60
4. Adaptive System Shell	
4.1 Introduction	64
4.2 Implementation	72
4.3 Conclusions	81
5. General Discussion	83
References	89
Appendices	93

List of Figures

1. Graphs of mean task completion times for the low and high scoring groups on each test for each interface. 25
2. Mean test session times with the command and menu interfaces for subjects in the low and high spatial ability groups. 47
3. Mean test session times with the command and menu interfaces for low and high spatial ability subjects with low and high previous command experience. 49
4. Individual subjects' test session times for the command interface against their spatial ability scores, giving their command experience. 51
5. Mean ease and enjoyment ratings for low and high spatial ability subjects with low and high previous command experience. 53
6. Mean number of errors per person per test session with each interface for low and high spatial ability users with low and high previous command experience. 57

7. User model.	145
8. Adaptation rules.	157
9. Dialogue record.	161
10. Inference rules.	165
11. Management system.	168

(Figures 7-11 are in Appendix 12)

List of Tables

1. Correlations between test scores and task completion times for the five interfaces.	24
2. Mean task completion times (in seconds) for the low and high scoring groups on each test for each interface.	26
3. Correlations between subjects' scores on the psychological tests.	29
4. Correlations between test scores and errors and uses of help for the command interface.	34
5. Mean test session times (in seconds) with the command and menu interfaces for low and high spatial ability subjects with low and high previous command experience.	49
6. Mean ease and enjoyment ratings for the command and menu interfaces for low and high spatial ability subjects with low and high previous command experience.	52
7. Mean number of errors per person per test session with each interface for low and high spatial ability subjects with low and high previous command experience.	57

1.

INTRODUCTION

Scope

This study investigates how a number of users' individual characteristics relate to their performance on a number of interface dialogue styles for a database system. The focus of the study is on a practical investigation of the interaction of individual differences with dialogue styles, rather than a theoretical analysis of individual differences between users. The aim of this work is to examine the feasibility of implementing findings such as these in a real 'adaptive system shell'. The adaptive system shell provides a generic architecture designed to allow the findings to be coded into it, to produce an adaptive system which automatically presents appropriate interfaces to the appropriate users according to the findings.

**

Database systems, especially systems such as public information systems, often have a large end-user population. With a large user population, there can be a considerable diversity among the users in terms of individual characteristics such as aptitudes and experience. Recent studies have suggested that large performance differences on computer based tasks can result from individual differences such as these, performance differences in the order of 20:1 not being uncommon for a group of thirty users (Egan, 1988). To ensure that all the users in a diverse user group are served well by a computer system, it may be necessary to produce different designs for the system to suit users with different individual characteristics. This study considers one area of system design: interface dialogue style.

Advances in computer technology, such as high resolution screens and direct manipulation devices, have allowed the development of a variety of interface

dialogue styles for computer systems. Dialogue style refers to the manner of communication between the user and the computer, for example whether the user communicates with the computer by typing in commands, by selecting items from menus, or by manipulating icons using a mouse. Different dialogue styles place different demands on the users of a system. For example, command dialogue styles require a user to remember the range of options for input, and the syntax of the input. Menu dialogue styles, on the other hand, present a user with all the options, so that the user does not have to remember the range of options or syntax. Iconic dialogue styles offer options for input, but require a user to interpret icons (Schneiderman, 1986).

It is likely that the different demands posed by different dialogue styles will suit some users better than others, according to their individual characteristics. Database systems which are currently available do have different dialogue styles, for example many systems have command style interfaces and many have menu style interfaces. This suggests that different dialogue styles do suit different users of database systems, but it is not clear which dialogue styles suit which users. Everest (1986) makes assumptions about which dialogue styles suit users with which characteristics, but offers no evidence in support of these assumptions. For example, he suggests that menu style interfaces suit users who are novices, because the user does not need to know the options for input and their syntax. Whereas command style interfaces suit users who are frequent users, because the users are familiar with the commands. However Davis (1989), comparing the use of a command and a menu style database interface, found that first time users generally performed better with the command interface than with the menu interface.

This study examines whether different dialogue styles do in fact suit different users, and tries to identify which dialogue styles suit users with which individual characteristics. Two experiments were carried out. The first experiment was an exploratory experiment to examine the performance of users on different dialogue styles, and to try to establish which characteristics of the dialogue styles and which characteristics of the users could account for any performance differences. The experiment was designed to generate hypotheses about which dialogue styles with which characteristics suit users with which characteristics.

The study concentrated on looking at certain types of user characteristics. It is important to provide users with an interface dialogue style which suits those of their characteristics which are stable, and which the users cannot alter in order to suit themselves to a dialogue style. Van Muylwijk, van der Veer and Wearn (1983: in Wearn, 1989) placed user characteristics along a dimension of stability from personality characteristics, which are the most resistant to change, through cognitive style and learning style to personal knowledge, which is the least stable. This study concentrated on looking at cognitive abilities and cognitive styles. Cognitive abilities represent quantitative differences between users, for example short term memory capacity where one person can remember more than another. Cognitive styles, on the other hand, represent qualitative differences between users (Robertson, 1985). For example with the style of logical-intuitive thought, people process information in a different way at different ends of the dimension, logical thinkers basing decisions on objective facts, and intuitive thinkers on subjective feelings.

Robertson makes the distinction between cognitive strategies and cognitive styles. Cognitive strategies he suggests are situation-specific differences in information processing and can differ across different situations for a single person, whereas cognitive styles represent more stable underlying differences in information processing which are maintained across situations. It is therefore cognitive styles that are of interest here rather than cognitive strategies, as these are fundamental differences in the cognitive processing of users which apply to many situations.

A data-driven approach was adopted for the first experiment in this study, to generate ideas about what the important characteristics of different dialogue styles are which determine people's success or failure of use, and how these relate to the users' individual characteristics. A data-driven approach was adopted because there is no real taxonomy of different dialogue styles and what are the important differences between them. However, some general distinctions have been drawn. For example, Fowler, Macaulay and Siripoksup (1987) use three distinctions to describe different dialogue styles: form, structure and content. The form of the dialogue concerns the degree of control a user has over the dialogue. The user can have a high degree of control, in which case the dialogue is user-guided; or the control can be placed with the system, in which case the dialogue is system-guided. The structure of the dialogue concerns the flexibility of response which is possible. The user can be restricted to specified inputs in given sequences, in which case the dialogue style is inflexible, or can have much more choice of input, in which case the structure is more flexible. Lastly, the content of the dialogue concerns how natural or formal the dialogue is. A natural dialogue uses meaningful words or sentences, whereas a formal dialogue uses codes and abbreviations.

Fowler et al. suggest that particular dialogue styles consist of a combination of these three factors. For example, question and answer dialogue styles are system-guided, inflexible and can be either natural or formal. Menu dialogue styles are again system-guided, but offer a user a choice of responses making them flexible. They can again be either natural or formal in content. Command language dialogue styles are user-guided, usually flexible, and with formal content. These distinctions provide a useful starting point for describing the differences between different dialogue styles. However there appear to be other distinctions between dialogue styles which are not covered by these. For example, not all dialogue content is in a natural or formal verbal form; some dialogue styles, such as iconic dialogue styles, have a pictorial content.

Rather than approaching the first experiment from a top-down point of view, hypothesizing what the important differences between interface dialogue styles are, and how users are likely to perform on these according to their individual differences, this first experiment is designed to help generate these hypotheses. The second experiment tests the hypotheses generated from the first experiment about what the important distinctions between different dialogue styles are, and the important characteristics of users which relate to their performance on the dialogue styles.

The experiments examine which dialogue styles suit which users after the initial learning stage of a system has taken place, looking at the dialogue styles which suit users in the long run. This study therefore represents a different approach from studies which are concerned with individual differences in gaining experience. Van der Veer, Tauber, Wearn and van Muylwijk (1985) looked at individual differences in gaining experience,

looking at how individual users can be aided in developing a correct mental model of a computer system. They outline how individual differences could relate to users' success at forming a mental model. Once a user has become experienced with a system, the user has developed an adequate mental model of that system. However, it is still possible at this stage that users' individual differences will affect how well users are able to use and maintain this mental model.

The suggestion that different users of a database system require interfaces with different dialogue styles according to their individual characteristics, has implications for the design of database systems. It raises the issue of how to design a system which can provide different users with different interfaces. One solution would be to make the different interfaces for a system available for the users to choose which one to use. However this would involve users in additional workload, as users would have to be aware of the range of possible options, and would have to try out the options and try to decide which was the best for them.

An alternative solution would be for the computer system to determine which of the different interfaces for the system was suitable for a user, and to present the user with just this interface. For the system to do this, it would have to gather information about users' individual characteristics, and relate this information to the most suitable interface for a user. A computer system which is capable of automatically altering some aspect of itself in this way to suit its users is known as an 'adaptive system'.

These two solutions of choice by the user and choice by the system are represented on a table of types of adaptivity, produced by Edmonds (1987).

Choice by the user corresponds to the first column: on request, and choice by the system corresponds to the third column: automatic.

	On Request	Prompted	Automatic
Common User Errors			X
User Characteristics		X	X
User Performance	X	X	
User Goals	X	X	
Information Environment	X		

A third solution: prompted, in the second column, represents a combination of choice by the user and choice by the system. A cross is given on the table where Edmonds considers adaptation of the given type to be possible. It can be seen from this table that automatic adaptation is considered to be possible by Edmonds for user characteristics as in this study.

A system which automatically provides users with the interface which is suitable for them, appears to involve the users in less work than a system in which they have to work out which interface is best. However a system such as this incurs large computational and developmental overheads for the mechanisms necessary to gather information about users and relate this information to the suitable interfaces for users. Computational overheads are no longer crucial with the current development of more powerful computer systems at lower costs, but development overheads are.

One way of reducing development overheads would be by producing a generic architecture for the mechanisms necessary for an adaptive system, into which the details of a specific system could be entered to make the system adaptive in some way: an 'adaptive system shell'. Benyon, Murray and

Jennings (1990) have developed a prototype for just such a shell. A successful shell would make viable the production of adaptive systems.

The final part of this study considers the use of a version of the prototype shell developed by Benyon et al., to examine whether the shell can work successfully for producing an adaptive database system, which automatically provides different users with interfaces with different dialogue styles to suit their individual characteristics.

This experiment compared the performance of users on several different dialogue styles for a database system with several of their individual characteristics, to see if there were any performance differences on the dialogue styles which related to the users' individual differences. The different dialogue styles were examined to try to establish the characteristics of the dialogue styles which could account for differences in performance. This allowed hypotheses to be developed about which dialogue styles with which characteristics are suitable for users with which characteristics.

Five dialogue styles were considered: command language, question and answer, mouse and menu, mouse and button and iconic. These dialogue styles were chosen because they represent a range of typical dialogue styles available with current computer technology. Many mainframe applications, such as E-mail on the VAX, employ command language dialogue styles, a user typing in syntactically correct commands at a system prompt. Many commercial data entry systems employ a question and answer dialogue style, the user typing in answers to specific questions or answers to prompts for specific information. Most Apple Macintosh applications, for example Microsoft Word, employ a mouse and menu style interface, the user selecting options from menus of the possible alternatives using a mouse. Many Hypercard applications employ a mouse and button dialogue style, the user again selecting options, but this time using a mouse to select named buttons representing the alternatives. Finally applications such as MacPaint or

MacDraw on the Apple Macintosh employ an iconic dialogue style, the user selecting pictures representing objects and operations using a mouse.

Six characteristics of users were considered: spatial ability, verbal ability, field dependency, short term memory (STM) capacity, logical versus intuitive thought, and previous experience with using different dialogue styles. These characteristics were chosen because, from previous research relating user characteristics to performance on different interfaces, they looked likely to be of relevance to users' performance on different dialogue styles.

Spatial ability is a measure of an individual's ability to perform tasks such as picturing objects and the relationships between them, and being able to picture the objects and their relationships in different orientations. Vicente, Hayes and Williges (1987) found that subjects who had a low spatial ability took twice as long to find a specific piece of information in a hierarchical file system than subjects with a high spatial ability. Vicente and Williges (1988) altered the file system by providing a partial map of the hierarchy and an analogue indicator of current file position. They found that this improved the performance of both the low and high spatial ability subjects. The results of these experiments suggested that spatial ability relates to the ability to navigate unaided through a hierarchical structure. As some interface dialogue styles can involve navigation through a hierarchical structure, spatial ability could relate to users' performance on different dialogue styles.

Vicente et al. (1987) also found that verbal ability was a predictor of performance on the hierarchical file searching task, although it was less

influential than the spatial ability measure. Verbal ability is a measure of linguistic capabilities such as comprehension of written material, breadth of vocabulary and reading speed. As some interface dialogue styles can require the reading of a lot of information from the screen and the entering of a lot of written input, verbal ability could also relate to users' performance on different dialogue styles.

Field dependency is a measure of an individual's ability to separate an item from an organized field (for example, in the Embedded Figures Test). Field dependent individuals find it hard to pick out salient features and ignore background distractions. Fowler and Murray (1988) suggested that field dependent individuals would operate best using computer systems with inflexible dialogue structures which restrict the range of user responses to set inputs in specified sequences. As some dialogue styles can allow the user a great deal of flexibility and others can be more restrictive, field dependency could relate to performance.

Benyon, Milan and Murray (1987) examined how users' STM capacity related to their ability to retain material presented in a fast or slow dialogue on a computer. STM capacity is a measure of the amount of information a user can hold in a rehearsal loop. Benyon et al. found that subjects with a high STM capacity performed better than subjects with a low STM capacity, for the slow dialogue presentation. The fast dialogue presentation, however, proved too fast for both the high and low STM subjects, and both performed poorly. The experiment showed that STM capacity can relate to users' success at retaining material. Some dialogue styles require a lot of information to be retained, whereas others do not.

Logical versus intuitive thought refers to whether individuals' tend to base decisions on objective facts or on subjective feelings. Garceau, Oral and Rahn (1988) compared the performance of subjects on a decision making task given presentation of data in two formats: tabular and graphic (ie. tables of numbers and bar graphs). They found that, on the whole, subjects with a logical cognitive style performed better with the tabular presentation of data, and subjects with an intuitive cognitive style performed better with the graphic presentation of data. As some dialogue styles have a more graphic or visual way of presenting information than others, users' tendencies towards logical or intuitive thought could affect how well they perform on different dialogue styles.

Users' previous experience with using different dialogue styles was considered as a user characteristic, although no research has been carried out which clearly shows whether this could be an important characteristic or not. Vicente et al. (1987) looked at whether users' number of hours of computer experience related to their performance on the hierarchical file searching task, but did not look at a detailed level at users' specific experience of systems similar to the test system. Similarly, Davis (1989) looked at whether the number of computer courses users had taken, and the number of computer languages and application packages users were familiar with, related to their performance on command and menu style interfaces to a database system. Again he did not look at a detailed level at users' specific experience of using command and menu style interfaces.

As stated earlier, it is important to provide users with an interface which suits those of their characteristics which are stable, and which they cannot alter in order to suit themselves to an interface. According to Van

Muylwijk, van der Veer and Wearn's dimension of stability, spatial ability, verbal ability, field dependency, STM capacity and logical versus intuitive thought do fall into this category. Previous experience does not. However if users only have infrequent access to computer systems, experience can then remain fairly stable and consequently could prove important.

Subjects

Twenty four subjects, sixteen male and eight female, participated in this experiment. The subjects were graduates, aged between twenty five and forty five, who used computers as part of their everyday work for various tasks from word processing to programming. The subjects were familiar with both mouse and keyboard input. This subject group was selected as the experiment was not concerned with looking at which dialogue styles suit which users when they first use a system, but with the dialogue styles which suit them in the long run. The subjects therefore needed to be capable of learning a system quickly to a reasonable degree of proficiency, and had to be able to type and use a mouse. The subjects were paid volunteers.

Test System

A mail order catalogue database was developed as the test system for this experiment with five interfaces, one running on each of the five dialogue styles: a command interface, a question interface, a menu interface, a button interface and an iconic interface. (See Appendix 1). All the interfaces supported only one task, that of accessing items available from the catalogue. To do this, with each interface the users had to specify the type of item they were interested in, for example vacuum cleaners, and attributes for the item type, for example that the vacuum cleaners must cost less than a hundred pounds, be grey in colour and have a power of a thousand

watts. The user was then presented with a list of items available from the catalogue which fitted the specification.

Each interface was designed to be as clear and as easy to use as possible, and to be a typical example of the use of the particular dialogue style it employed. The interfaces were evaluated by an independent HCI expert (Rogers, 1990), and recommendations for improvement were implemented.

Interfaces such as the iconic interface did involve the partial use of other dialogue styles, but this is typical of iconic interfaces, such as MacDraw and MacPaint which rely on the use of menus for some operations which do not lend themselves to an iconic representation. The test system was designed to be simple to learn, so that users could become proficient at using the system quickly.

The test system was implemented in KEE on a SUN workstation. The system was set up so that only the left hand mouse button needed to be used, the other two were disabled. This meant that the test system was comparable to mouse systems already used by the subjects (for example, Apple Macintosh).

User Characteristic Tests

Tests were chosen to measure the user characteristics which corresponded to the tests used in the research which had suggested the relevance of the characteristics. All the tests chosen were suitable for graduate level subjects.

Vicente et al. (1987) found that the spatial visualization section of the spatial ability tests which they used was the most accurate predictor of

performance on the hierarchical file searching task. A test was therefore chosen to measure spatial ability for this experiment which was deemed to involve spatial visualization; the test involved determining which of several cubes could be produced from a given patterned cube net (Spatial Reasoning ST7, Saville and Holdsworth Ltd.).

Vicente et al.'s measure of verbal ability involved tests of reading rate, vocabulary and comprehension. For this experiment, a verbal ability test was chosen which involved subjects reading passages of text and then answering questions about the passages in a set time (Verbal Ability, NFER-NELSON ASE Division). Subjects with a high reading rate, large vocabulary and good comprehension skills were likely to do well on this test.

The test which was chosen to measure field dependency was an abstract ability test which was designed to measure whether a person could think flexibly, could recognize order in the midst of apparent chaos and whether they could focus on certain aspects of a task and ignore irrelevant detail. The test involved deciding which of two sets of patterns of shapes a given pattern of shapes belonged to (Abstract Ability, NFER-NELSON ASE Division). A low score on this test was taken to indicate field dependence.

A simple word memory test was chosen to measure STM capacity. The test involved giving subjects a list of ten four letter nonsense words to look at for thirty seconds, and asking them to write down as many as they could remember in any order (Benyon et al., 1987).

The thinking-feeling dimension of the Myers Briggs Type Indicator was chosen to test for logical versus intuitive thought.

Users' previous experience with using the different computer dialogue styles was measured using a questionnaire, which asked users to rate their previous experience with each dialogue style on a three point scale. (See Appendix 2).

Performance Measures

Four performance measures were chosen to assess users' success at using the different test system interfaces. The measures involved a combination of quantitative and qualitative measures. The first measure was the time subjects took to complete a set sequence of tasks using the different interfaces. The time subjects took to complete the tasks was likely to indicate the level of difficulty they experienced, as the more difficult they found an interface the longer they were likely to take determining appropriate input, and the more errors they were likely to make. The time difference measure could be used to indicate the comparative difficulties subjects found with each interface, but could not be used to examine the relative difficulties of the different interfaces for each subject, as each interface took a different amount of time to use to carry out tasks due to its actual design. For example, the menu interface required more mouse clicks than the iconic interface to carry out the same task.

Two qualitative measures were chosen, which involved asking each user to rate on a five point scale how easy they found each interface to use, and how enjoyable they found each interface to use.

The final measure involved examining the number and types of errors subjects made on each interface. The number of errors was likely to indicate how

difficult subjects found each interface to use, and the type of errors was likely to indicate where the difficulties lay.

Experimental Design

Each subject used all five interfaces. The order of presentation of the interfaces was randomized for each subject. With each interface, the subject carried out a series of tasks using the interface as a practice session, followed by a series of tasks as a test session. Subjects were given different but equivalent tasks to carry out for the practice and test sessions for the five interfaces, to reduce practice effects while maintaining comparability. The tasks were balanced in terms of the type of the tasks, the number of steps required to complete the tasks, the amount of typing involved and the catalogue categories the tasks involved. (See Appendix 3 for example test session tasks).

Procedure

Before each subject began the experiment, the purpose and the procedure of the experiment were explained to the subject. (See Appendix 4 for instruction sheet). The practice session for the first interface was then started. During the practice session the experimenter was available to provide any necessary help to the subject and to answer any questions the subject had about how to use the interface to carry out the practice session tasks. The practice session was not pre-structured as different subjects required different amounts of help with the different interfaces. When the subject was confident that they had learned how to use the interface and was

happy to move on to the test session, the test session was started. In the test session the subject was required to carry out the test session tasks on their own without help from the experimenter. The experimenter sat away from the subject behind a screen. This procedure of the practice session followed by the test session was then repeated for the remaining four interfaces. The experimenter timed how long the subject took to carry out the test session tasks with each interface, and videoed the test session interactions. The complete session lasted approximately one hour.

After each subject had completed the computer session, they were given the five point rating scales on which to record their perceived ease and enjoyment of each interface. They were also given the six user characteristic tests to complete. The subjects were allowed to carry out the tests in the order they wished. The tests took approximately two hours.

Statistical Analysis

For each measure of subjects' performance, two analyses were carried out. Firstly, Pearson product moment correlation coefficients were calculated between subjects' scores on the user characteristic tests and their performances on the interfaces, to see if there were any significant correlations which could suggest a link between a user characteristic and performance on an interface. Secondly, the subject group was divided into two according to subjects' scores on the user characteristics, the twelve highest scorers constituting a 'high' group and the twelve lowest scorers constituting a 'low' group for each characteristic. Where a significant correlation did suggest a link between a user characteristic and performance on an interface, the mean performances of subjects in the high and low

groups for that characteristic on the interface were compared using two sample t-tests, to see whether users in the high and low groups did perform significantly differently on the interface. This two-stage analysis followed that of Vicente et al. (1987).

The second analysis was only carried out if the correlational analysis was significant, to reduce the chance of Type 1 errors. The likelihood of Type 1 errors would have been very high if t-tests had been carried out between the mean performances of low and high scoring groups on each user characteristic for each interface for each performance measure in this post hoc manner.

All the subjects completed all the test session tasks correctly using the five interfaces.

Test Session Times

Results:

Table 1 gives the Pearson product moment correlation coefficients calculated between the subjects' scores on the user characteristic tests and the times they took to complete the test session tasks for each interface. Scores on the spatial test correlated significantly with performance on both the command and the question interfaces, and approached a significant correlation with performance on the button interface; in each case the higher score relating to faster performance. Scores on the verbal test correlated significantly with performance on the question interface, a high verbal score relating to faster performance. Scores on the field dependency test approached a significant correlation with performance on the command interface, people who were more field independent performing faster. Scores on the thinking/feeling scale approached a significant correlation with performance on the iconic interface, people towards the thinking end of the scale performing faster. Scores on the short term memory test, and the previous experience ratings, did not correlate significantly with performance on any of the interfaces.

Table 1. Correlations between test scores and task completion times for the five interfaces.

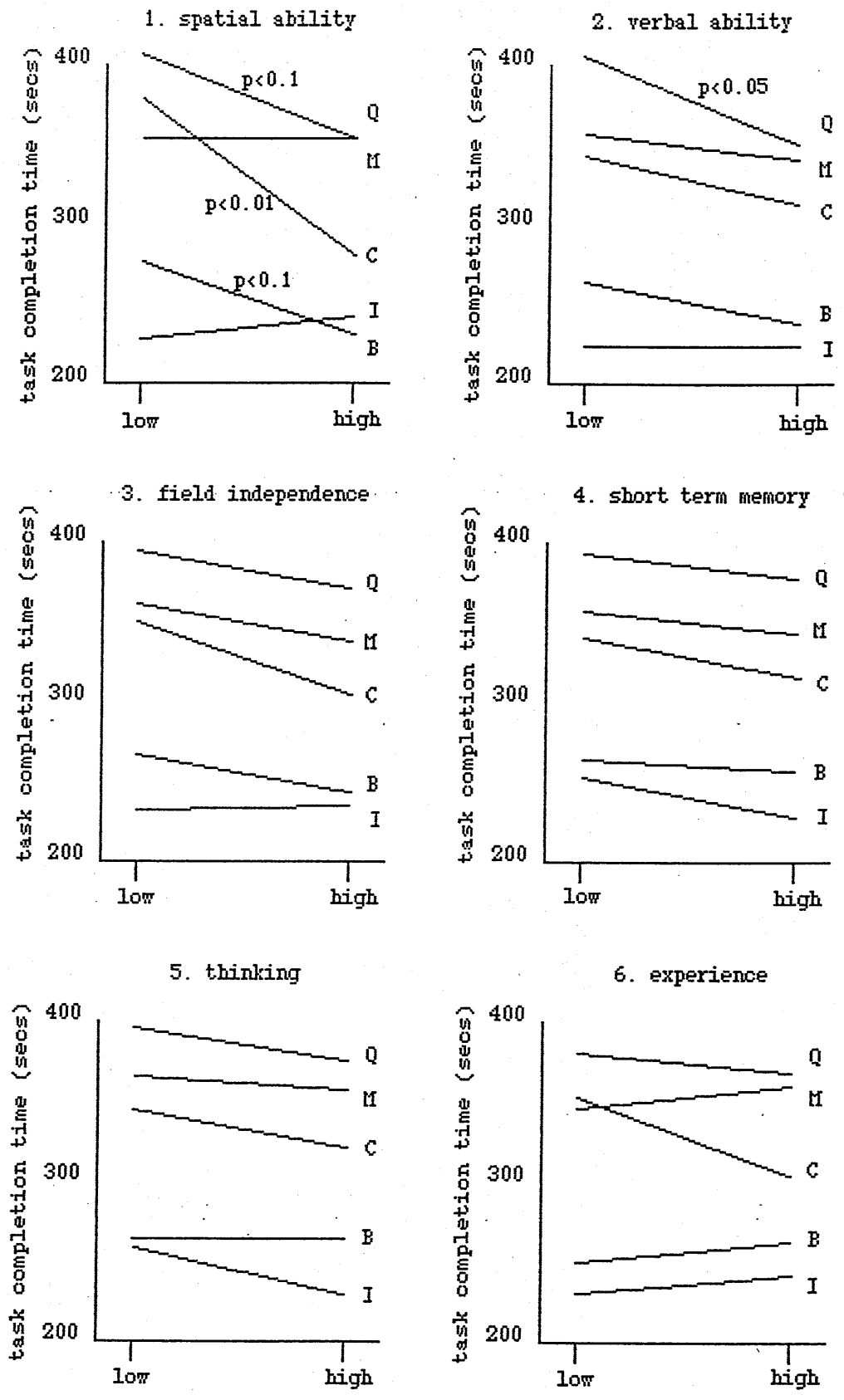
	interface				
	command	question	menu	button	iconic
spatial ability	-0.58 ^{***}	-0.48 ^{**}	-0.07	-0.37 [*]	-0.02
verbal ability	0.01	-0.43 ^{**}	-0.13	-0.07	0.06
field independence	-0.39 [*]	-0.08	-0.07	-0.19	0.20
short term memory	0.03	-0.15	-0.34	-0.08	-0.29
thinking/feeling	-0.08	-0.08	-0.13	0.08	-0.36 [*]
experience	0.33	-0.03	-0.12	-0.05	-0.21

* $p < 0.1$

** $p < 0.05$

*** $p < 0.01$

Table 2 gives the mean times taken for the highest twelve scorers and the lowest twelve scorers on each user characteristic test to complete the test tasks with each interface. Figure 1 shows these means graphically. Mean times have been included for the experience characteristic, although the three point rating scales for experience proved rather crude, many subjects rating themselves on the centre point. Therefore the high and low groups for the experience characteristic each contain a large random section of users rating themselves in the centre. Two sample t-tests (two-tailed) showed that the high and low groups on the spatial ability test differed significantly in their performance on the command interface ($t=3.34$, $df=22$, $p < 0.01$), and approached a significant difference in their performance on the question ($t=1.88$, $df=22$, $p < 0.1$) and button ($t=1.73$, $df=22$, $p < 0.1$) interfaces. The high and low groups on the verbal ability test differed significantly in their performance on the question interface ($t=2.11$, $df=22$, $p < 0.05$). However the high and low groups on the field independence test did not differ



C = command Q = question M = menu B = button I = iconic

Figure 1. Graphs of mean task completion times for the low and high scoring groups on each test for each interface.

significantly in their performance on the command interface ($t=1.20$, $df=22$, $p>0.1$), despite the correlation between the field independence score and command interface task time approaching significance. The high and low groups on the thinking/feeling scale (high=thinking, low=feeling) also did not differ significantly in their performance on the iconic interface ($t=1.60$, $df=22$, $p>0.1$).

Table 2. Mean task completion times (in seconds) for the low and high scoring groups on each test for each interface.

		interface				
		command	question	menu	button	iconic
spatial ability	low	377	406	355	275	231
	high	278	352	352	232	238
verbal ability	low	340	409	368	268	235
	high	316	350	340	240	235
field independence	low	349	388	369	266	231
	high	307	371	339	241	239
short term memory	low	344	383	360	258	243
	high	312	376	347	250	226
thinking	low	334	393	359	254	250
	high	322	366	348	254	222
experience	low	356	384	341	242	225
	high	300	375	365	265	245

Discussion:

The results of the test session times analysis suggested that spatial ability and verbal ability were the important user characteristics in determining which dialogue styles suited which users. The results suggested

that any of the five dialogue styles would be suitable for users with a high spatial ability, but that dialogue styles such as the command, question and button styles would not be suitable for users with a low spatial ability. Also, any of the dialogue styles would be suitable for users with a high verbal ability, but dialogue styles such as the question style would not be suitable for users with a low verbal ability.

By comparing the command, question and button interfaces to see what their dialogue styles have in common, it is possible to suggest what it is about the command, question and button dialogue styles which was causing problems for the low spatial ability users. The command, question and button interfaces all require navigation through a hierarchy to achieve the tasks. However, the iconic and menu interfaces, on which the low spatial ability subjects performed equally well with the high spatial ability subjects, require this to a much lesser extent. The command interface has three levels: the system level from which the catalogue can be accessed and to which the user returns when they quit from the catalogue, the catalogue level where the user can specify the catalogue items they are interested in and from where they can access the catalogue help system, and the help system level. Although different system prompts indicate the different levels, the user has to keep in mind the level they are in, and how the levels link together. For example a user cannot go straight from the system level to the help level without going through the catalogue level. The button and question interfaces are hierarchical in the sense that they require the user to go through a series of hierarchical category choices in order to reach the item type they are interested in. The user has to understand the structure of the hierarchy and be able to navigate around within it. The iconic interface however, after an initial choice of two

categories, presents a single visual scene from which the user can pick out the type of item they are interested in. The menu interface involves a walk-through menu in order to select the desired item type; the hierarchy of category choices is displayed as the user walks through the menu so that the structure does not have to be remembered by the user. These results suggest that spatial ability could relate to a user's ability to cope with an interface requiring navigation within a hierarchical structure. Being able to cope with a hierarchy involves knowing where things are in the structure and how to navigate through the structure efficiently to reach them.

The difference in performance between the low and high spatial ability groups was much greater for the command interface than for the question and button interfaces. This suggests that there is something additional about the command interface which was causing users difficulty who have a low spatial ability. The command interface involves a much less constrained dialogue than the question, button, menu and iconic interfaces in which the interaction is quite structured. With the question, button, menu and iconic interfaces the user is presented with screens clearly indicating whether a category, item type or attribute choice should be made next, or an operation used to open, close or return to the start of the catalogue. However with the command interface a prompt is displayed at all points, with little indication of what type of input is expected next from the user. Spatial ability could therefore also relate to a user's ability to cope with interfaces allowing a very open and flexible dialogue. Being able to cope with a flexible dialogue involves being clear when and where particular input can be used to best advantage and what its outcome will be.

However this latter result may be due to the fact that the users' spatial ability and field independence scores correlated significantly. (See Table 3 for Pearson product moment correlation coefficients calculated between subjects' scores on the psychological tests). Eighteen out of the twenty four subjects were in the same group for spatial ability as they were for field independence ie. in the high or low groups for both characteristics.

Table 3. Correlations between subjects' scores on the psychological tests.

	verbal ability	field independence	short term memory	thinking/feeling
spatial ability	0.22	0.53 *	0.18	0.02
verbal ability		-0.05	-0.08	0.35
field independence			0.27	0.07
short term memory				-0.16

* $p < 0.05$

It may in fact be field dependency which relates to the ability to cope with a flexible dialogue, as suggested by Fowler and Murray (1988). Figure 1.3, lends some support to this idea. Subjects with a high field independence performed faster on the command interface than subjects with a low field independence, although the result did not reach significance, while performing similarly on the other interfaces. The alternative argument is that these effects for the field dependency scores were due to the correlation between spatial ability and field independence. However, if this was the case, a correlation of some degree would be expected between the field dependency scores and performance on the button and question interfaces as well. The correlation between field dependency and performance

on the question interface was so low ($r=-0.08$), that it seems likely that field dependency was indeed causing a separate performance effect for the subjects on the interfaces from spatial ability. The question interface, although hierarchical, has a very constrained dialogue.

The results therefore suggest that users with a low spatial ability, who are mostly field dependent as well, are not suited to any interface dialogue styles which require navigation through a hierarchy or allow an open and flexible dialogue structure.

The question interface can be compared with the command, menu, button and iconic interfaces in a similar way to suggest what it is about the question interface dialogue style which caused problems for low verbal ability subjects. The question interface involves the reading of a lot of information from the screen and the entering of a large amount of accurate verbal input. This is not involved in the button, menu and iconic interfaces which are mouse operated, or in the command interface where a short statement is used to specify the type of item a user is interested in. This suggests that verbal ability could relate to a user's ability to cope with interfaces requiring the reading of information from the screen and the entering of accurate verbal input.

The results therefore suggest that low verbal ability users are not suited to any interface dialogue styles which involve a large amount of verbal input and output.

These results could be used to suggest that a non-hierarchical, inflexible dialogue style without a large amount of verbal input or output could be used as a single interface dialogue style to suit all users, such as the

iconic dialogue style in this experiment. All subjects performed equally well on the iconic interface, and it was quicker for completing the tasks than the other interfaces. However database data often lends itself to a hierarchical representation, or to some type of representation which involves several relations and therefore requires the user to remember where things are in the database. It would be hard to produce a database interface which avoided any navigational component. The iconic interface in this experiment only avoided this navigational component as it was attached to an unrealistically small database of information, which allowed many choices from the database to be presented on single screens. Users who are low on spatial ability would need to be given some kind of aid for carrying out any necessary navigation involved in using an interface, whereas this would not be necessary for users who are high on this characteristic. Although the high spatial ability users performed well with the highly constrained interfaces, if a large amount of work was being done with the system, it would prove very laborious for users to go through set routines who have the ability to take short cuts. This suggests that the high spatial ability users would be better off with an open and flexible dialogue style even though they are capable of using the constrained dialogue style. It does seem sensible to reduce verbal input and output for interfaces for all users though. Verbal input is time consuming, and can be reduced by the use of abbreviations or function keys for keyboard input, or by the use of mouse input if available.

It can be concluded from this analysis that two forms of interface dialogue style would be necessary for a database system in order to suit the users: a dialogue style which aids any necessary navigation and constrains the dialogue for users with a low spatial ability who are mostly field dependent

as well, and a dialogue style which allows free navigation and an open and flexible dialogue with the system for users with a high spatial ability who are mostly field independent. Both interfaces should minimize the amount of verbal input and output.

Ease and Enjoyment Ratings

Results:

The Pearson product moment correlation coefficients calculated between the scores subjects achieved on each of the user characteristic tests and the subjects' ratings of ease and enjoyment for each of the interfaces, showed only one significant correlation: between scores on the field independence test and the self rated ease of the question interface ($r=-0.53$, $df=22$, $p<0.05$). This suggested that subjects with a high field independence found the question interface easier than those with a low field independence. However a two sample t-test (two-tailed) between the mean ease ratings for the question interface for the low and high field dependency groups did not show a significant difference ($t=1.39$, $df=22$, $p>0.1$).

Discussion:

The ease and enjoyment ratings failed to show any links between users' characteristics and their feelings about the interfaces. However, even if some links did exist, these could have been obscured by users' different interpretations of the points on the rating scales. For example, some subjects tended to rate all the interfaces towards the lower end of the ease scale, and some tended to rate all the interfaces towards the higher end.

However this did not necessarily mean that the first group found the interfaces harder to use than the second group. It could mean that they were more critical of their own performance and less confident, and therefore did not like to say that they had found any of the interfaces very easy to use, but used more modest ratings. When comparing users' ratings of ease or enjoyment for a single interface, differences in users' interpretations of the scales produce a large amount of noise. This noise could be so great as to disrupt the statistical detection of any between subjects effects due to the experimental variable.

Errors

Results:

The number of errors made by subjects during their interactions with the interfaces were counted from the video recordings of the subjects' interactions. Two types of error were examined. The first, syntactic, was when for example a user missed out brackets in the item specification statement with the command interface, or wrote the wrong number of arguments or wrote the arguments in the wrong order. Syntactic errors for the mouse operated interfaces included errors such as a user clicking in the wrong place. The second type of error, command use, was when for example with the command interface a user typed in an inappropriate command for the level of the system they were in eg. typing the item specification statement when still in the help system. For the question, button, menu and iconic interfaces command use errors included errors such as the user using an inappropriate operation, for example using 'start' to return to the very beginning if they entered a category, item type or attribute choice wrongly,

instead of 'previous' just to repeat the last entry. In addition, the number of uses of the help system for the command interface was looked at.

Table 4. Correlations between test scores and errors and uses of help for the command interface.

	syntactic error	command use error	use of help
spatial ability	-0.23	-0.21	-0.13
verbal ability	0.12	-0.27	0.13
field independence	-0.21	-0.65**	-0.17
short term memory	-0.03	-0.03	0.09
thinking	-0.09	0.18	0.26
experience	-0.05	0.36*	0.32

* $p < 0.1$

** $p < 0.01$

Analyses were only carried out on the errors subjects made on the command interface. Less than one quarter of subjects made any of the errors on the other interfaces, so this did not provide enough data to analyse. Table 4 shows Pearson product moment correlation coefficients calculated between the users' numbers of errors and uses of help for the command interface and their scores on the user characteristic tests. Subjects' scores on the field independence test correlated significantly with the number of command use errors, the higher a subject's field independence the less errors were made. Subjects' ratings of previous experience with command style interfaces also approached a significant correlation with the number of command use errors. Two sample t-tests (two-tailed) showed that the mean number of command use errors for the low and high field independence subjects differed

significantly ($t=2.53$, $df=22$, $p<0.05$), but not for the low and high command experience subjects ($t=1.60$, $df=22$, $p>0.1$).

Discussion:

The fact that the command use errors on the command interface related to a low field independence fits in quite well with the idea that subjects with a low field independence have trouble with flexible dialogue structures. The command use errors are the type which would be expected in this case, the subject forgetting when and where commands could be used.

This experiment suggested that at least two different interface dialogue styles are needed for a database system if it is to suit a range of users: a dialogue style which aids any necessary navigation and which constrains the dialogue for users with a low spatial ability, who are mostly field dependent as well; and a dialogue style which does not need to aid navigation and which allows an open and flexible dialogue for users with a high spatial ability who are mostly field independent. Both dialogue styles should minimize the amount of verbal input and output necessary. The results fit in with those of Vicente et al. (1987), who suggested a link between spatial ability and the ability to navigate a hierarchical structure; and with the work of Fowler and Murray (1988) who suggested a link between field dependency and the ability to cope with an open and flexible dialogue.

The fact that subjects' scores on a user characteristic relate to their performance on an interface, does not however show that the characteristic actually causes the performance differences. The user characteristic could correlate with another characteristic which actually causes the effect (Egan and Gomez, 1985). As mentioned earlier, some of the performance differences which relate to users' spatial ability may be due to users' field dependency, which correlated with their spatial ability. The performance differences could not however have been caused simply by a general intelligence factor, as users' scores on the spatial ability test in this experiment did not correlate for example with their scores on the verbal ability test, suggesting that the tests are measuring some differences other than purely differences in general intelligence.

If the type of abilities which are tapped by the spatial ability test used in the experiments are compared with the demands of an interface requiring navigation, it does look as if spatial ability could actually be a causal factor in performance rather than just a correlative factor. The spatial ability test required users to be able mentally to fold up the cube net, and visualize the orientations and adjacencies of the patterned sides of the resultant cube. To cope with an interface requiring navigation, users have to be able to form some mental view of the system as a whole, so that they can picture its structure and how the different parts of the system relate together. The two activities do therefore seem to bear some relation. Van der Veer (1990) has suggested also that there is a link between spatial ability and users' success at forming mental representations of computer systems.

However, even if spatial ability is not actually the causal factor, if the truly effective characteristics correlate consistently with users' spatial ability and the interface style is designed to suit the users' spatial ability, then the interface will automatically suit the truly effective characteristics as well.

The results of this experiment suggested two important characteristics of dialogue styles, which determine whether a dialogue style suits a user or not. The first is whether the dialogue style requires the user to navigate a structure where the full structure is not displayed; and the second is whether the dialogue style provides the user with the options for input or not. This second characteristic of dialogue styles corresponds to Fowler, Macaulay and Siripoksup's (1987) 'structure' characteristic.

From the first experiment it was concluded that two interface dialogue styles are necessary to suit the users of a database system: a dialogue style which aids any necessary navigation and which constrains the dialogue for users with a low spatial ability, who are mostly field dependent as well; and a dialogue style which does not need to aid navigation and which allows an open and flexible dialogue for users with a high spatial ability, who are mostly field independent.

The first experiment was designed to generate ideas as to how interface dialogue styles relate to user characteristics. Many of the effects found in the experiment, apart from that for spatial ability and performance on the command interface, were not statistically very strong. The results did not therefore indicate any definite relationships between interface dialogue style and user characteristics, but suggested the sort of relationships which could exist. This second experiment tests the conclusions of the first experiment.

Two interfaces were constructed to a database system, one with a dialogue style which aided navigation and constrained the dialogue, and one with a dialogue style which did not aid navigation and which allowed an open and flexible dialogue. The interfaces were tested to see whether users with a low spatial ability did perform better on the aided navigation and constrained dialogue interface than on the un-aided navigation and flexible dialogue interface as hypothesized; and whether users with a high spatial ability did perform better on the un-aided navigation and flexible dialogue interface than on the aided navigation and constrained dialogue interface.

Subjects

Thirty subjects, eighteen male and twelve female, participated in this experiment. The subjects were again graduates, aged between twenty five and forty five, who used computers as part of their everyday work for various tasks from word processing to programming. All the subjects could type. The subjects were paid volunteers.

Test System

A database was developed for this experiment which contained information about students and staff working at a university; with one interface with aided navigation and a constrained dialogue style, and one with un-aided navigation and an open dialogue style. (See Appendix 6).

The aided navigation and constrained style interface was based on the use of menus to access information. The interface was hierarchically organized to some extent, in that different choices from one menu opened up different sub-menus. However the interface forced users to start from the top level menu and move sequentially through the sub-menus each time they wished to access information. This meant that navigation was highly restricted and the user did not have to keep track of where they were and how to move to the other menu levels. In this interface, users' navigation was therefore aided by giving the system control over the paths a user could take through the hierarchy. At each point in the dialogue, the user was given a clear

indication of the possible input at that point, thereby constraining the dialogue.

The un-aided navigation and open style interface was based on the use of SQL (Structured Query Language) to access information. The interface contained three levels: the top system level, the database level and a help level. The user had to remember which level they were in, how to reach each level from the other levels and the operations which were possible at each level. The interface allowed users to move between the system, database and help levels at any point. The user had to construct syntactically correct query statements, and the interface allowed users to retrieve and edit previous statements. The interface therefore allowed free navigation and an open and flexible dialogue.

Both interfaces were designed to be as clear and as easy to use as possible. For example for the menu interface, menu headings were designed to be as unambiguous as possible and crowded menu screens were avoided, and for the command interface the line editing system used meaningful keys (eg. ^F for moving the cursor forwards, ^B for backwards). Both interfaces supported exactly the same range of queries.

The command interface provided potentially a very fast way of accessing information, as one statement could be entered to access the information, and statements could be retrieved, edited and reused. However if users had difficulty with the interface, many errors were possible and the interface was potentially very slow. The fact that the menu interface dialogue was very structured suggested that it would prove easy to use for all users, but that the number of steps which were involved each time to access information

would make this interface slower to use than the command interface for subjects who were proficient at using the command interface. This meant that the command interface should be faster to use than the menu interface for high spatial ability users, and therefore more suited to them; and that the menu interface should be faster to use than the command interface for low spatial ability users, and therefore more suited to them.

Interestingly, the two interface styles correspond to the two most commonly available styles of database system interface, suggesting that there is a recognized requirement for both types of interface in order to suit the user population.

The database system with its two styles of interface was implemented in KEE on a SUN workstation. Both interfaces operated using keyboard input only (ie. no mouse input). The database interfaces were displayed on the left hand side of the SUN workstation screen. On the right hand side of the screen the queries which the subject was required to answer using the database system were displayed. For the command interface, the names and the column headings of the relations in the database were displayed. With most SQL systems users have access to this information to help them formulate query statements, although not usually on-line.

User Characteristic Tests

The test used to measure users' spatial ability was the same test which was used in the first experiment, which involved users determining which of several cubes could be produced from given patterned cube nets.

Users' previous experience with using interface dialogue styles akin to the two test system dialogue styles was considered in this experiment, even though no clear effects of previous experience had been found in the first experiment. This was done because the three point rating scale used in the first experiment had proved rather crude, and could have been too crude to ascertain any effects of previous experience. A five point rating scale was used in this experiment. (See Appendix 7).

Performance Measures

The same four performance measures were used as in the first experiment: the time subjects took to complete set sequences of tasks using the interfaces, ease and enjoyment ratings for the interfaces on five point scales and the incidence of errors. However, in addition, in this experiment subjects were asked for any comments they had about the interfaces.

Experimental Design

Each subject used both interfaces. The order of presentation of the interfaces was counterbalanced, half the subjects receiving the command interface first and half the menu interface first. As for the first experiment, with each interface the subject answered a series of queries using the interface as a practice session, followed by a series of queries as a test session. Subjects received different but equivalent queries to answer using the database for the practice and test sessions for the two interfaces, to reduce practice effects while maintaining comparability. The queries were balanced in terms of the amount and the complexity of the

information which had to be extracted from the database. (See Appendix 8 for example queries).

Procedure

The procedure for this experiment followed that of the first experiment. Before each subject began the experiment, the purpose and the procedure of the experiment were explained to the subject. (See Appendix 9 for instruction sheet). The practice session for the first interface was then started. During the practice session the experimenter was available to provide any necessary help to the subject and to answer any questions the subject had about how to use the interface to answer the practice session queries. The practice session was not pre-structured as different subjects required different amounts of help with the different interfaces. When the subject was confident that they had learned how to use the interface, and was happy to move on to the test session, the test session was started. In the test session the subject was required to answer the test session queries on their own without help from the experimenter. The experimenter sat away from the subject behind a screen. This procedure of the practice session followed by the test session was then repeated for the second interface. The experimenter timed how long the subject took to answer the test session queries with each interface, and videoed the test session interactions. The complete session lasted approximately one hour.

After the final test session, the subjects were given the five point rating scales on which to rate how easy and enjoyable they found each of the interfaces, and were asked for any comments they had about the two

interfaces. Subjects were then given the previous experience and spatial ability tests. The spatial ability test took twenty minutes.

Statistical Analysis

The statistical analysis for this experiment was based on the second analysis which was used on the data in the first experiment. This involved comparing the mean performances of groups of subjects on the interfaces who were classified as low or high on user characteristics. For this experiment the comparisons were carried out using analyses of variance. As this experiment only looked at two user characteristics, spatial ability and previous experience, and two interfaces, analyses for all groups could be run without too high a risk of Type 1 errors. However tests for simple effects were only run when a significant interaction was obtained.

All the subjects answered all the queries correctly using the two interfaces to the database, apart from two subjects who each made one error.

Test Session Times

The subject group was divided into two according to their spatial ability scores: a high spatial group for the subjects who scored above the mean score in the spatial ability test, and a low spatial ability group for the subjects who scored below the mean in the spatial ability test.

The mean times for the subjects in each of these groups to complete the test session using each of the interfaces were calculated. The low spatial group took a mean time of 401 seconds to complete the test session using the command interface, and 410 seconds using the menu interface. The high spatial ability group took a mean time of 303 seconds to complete the test session using the command interface, and 398 seconds using the menu interface. (See Figure 2).

A 2(spatial ability group) \times 2(type of interface) analysis of variance showed that there was a significant interaction between spatial ability group and type of interface ($F(1,28)=8.00$, $p<0.01$). Simple effect tests for the low and high spatial ability groups showed that the low spatial ability subjects took a similar amount of time to complete the test session using the command interface and the menu interface ($F<1$), whereas the high spatial ability

subjects completed the test session significantly faster with the command interface than with the menu interface ($F(1,28)=21.2, p<0.001$).

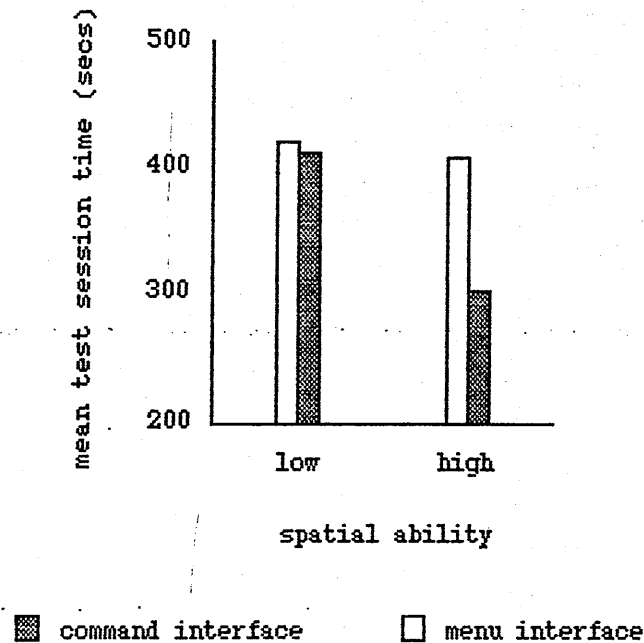


Figure 2. Mean test session times with the command and menu interfaces for subjects in the low and high spatial ability groups.

The above results did support the hypothesis that the command interface would be better than the menu interface for the high spatial ability subjects, as they did answer the queries quicker with the command interface. But the results did not support the hypothesis that the menu interface would be better than the command interface for the low spatial ability subjects, the interfaces proved equally good. In fact the results suggested that only the command interface would be necessary to suit all the users better or as well as the menu interface.

However, examination of the raw test times, showed that not all subjects had similar test session times for the command and menu interfaces, or always faster test session times using the command interface than using the menu interface. Six subjects in fact took longer to complete the test session using the command interface than the menu interface, so the menu interface did appear to be more suitable than the command interface for these subjects. All these subjects belonged to the low spatial ability group, and had rated their previous experience with using command style interfaces as low (ie. four or five on the five point rating scale, one equalling a lot of experience and five no experience). Previous experience with command style interfaces therefore seemed important as well as spatial ability in deciding the more appropriate of the two interfaces for the users.

Six subjects in the high spatial group rated their previous experience with using command style interfaces as high (ie. one or two on the five point scale), and six low (ie. four or five). Similarly six of the subjects in the low spatial group rated their previous experience with command style interfaces as high, and six as low. The remaining six subjects rated their previous experience as intermediate (ie. three). Mean test session times were calculated for each interface for the six high spatial ability and high experience subjects, for the six high spatial and low experience subjects, for the six low spatial and high experience subjects and for the six low spatial and low experience subjects. (See Table 5 and Figure 3).

Separate 2(command experience)x2(type of interface) analyses of variance were carried out for the low spatial ability subjects and the high spatial ability subjects. The low spatial ability subjects analysis showed that there was a significant interaction between previous experience and

Table 5. Mean test session times (in seconds) with the command and menu interfaces for low and high spatial ability subjects with low and high previous command experience.

		interface	
spatial	command experience	command	menu
low	low	525	423
	high	305	410
high	low	311	408
	high	260	362

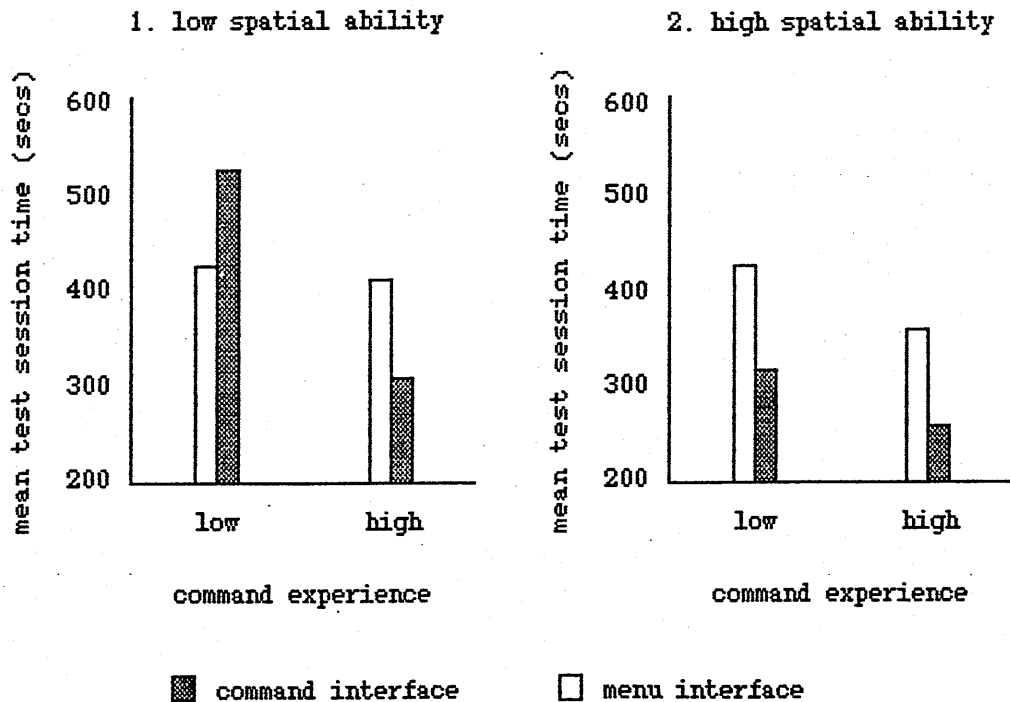
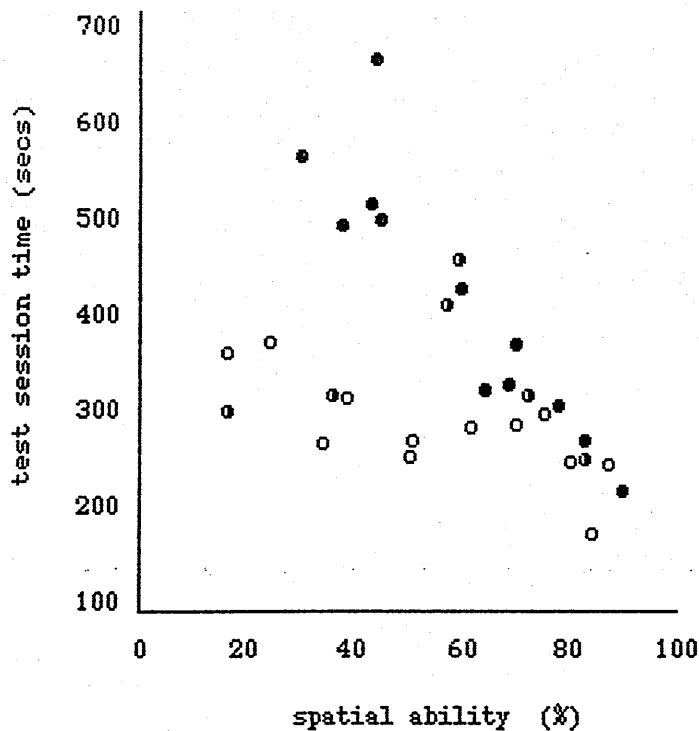


Figure 3. Mean test session times with the command and menu interfaces for low and high spatial ability subjects with low and high command experience.

interface type ($F(1,10)=46, p<0.001$). Simple effect tests for the low and high experience groups showed that the low experience subjects performed significantly faster on the menu interface than the command interface ($F(1,10)=22.1, p<0.001$), while the high experience subjects performed significantly faster on the command interface than on the menu interface ($F(1,10)=23.9, p<0.001$). The analysis of variance for the high spatial ability subjects showed that there was no significant interaction between previous experience and interface type ($F<1$), but that there was a main effect of interface type ($F(1,10)=87, p<0.001$), and the main effect of previous experience approached significance ($F(1,10)=2.72, p=0.13$). All the high spatial ability subjects performed faster on the command interface than the menu interface, and the subjects with a high previous experience of command style interfaces performed slightly faster on both interfaces than those with a low previous experience.

These results suggested that the command interface was more suitable than the menu interface for high spatial ability subjects and for low spatial ability subjects with high experience of command style interfaces, whereas the menu interface was more suitable than the command interface for low spatial ability subjects with low command experience. However any low spatial ability low experience users who use a command system regularly will increase their command experience, and consequently become low spatial ability high experience users for whom the command interface is better than the menu interface. So in fact intended frequency of use is important as well as spatial ability and previous experience in determining the appropriate interface: only low spatial ability users with low previous experience who are occasional users and will therefore not increase their experience are better suited to the menu interface. Low spatial low



previous experience: o high (1, 2) □ medium (3) • low (4, 5)

Figure 4. Individual subjects' test session times for the command interface against their spatial ability scores, giving their command experience.

experience users who are regular users and will therefore increase their previous experience are in fact better suited to the command interface.

The results showed that many limitations on subjects' performance on the command interface, which related to their spatial ability, could be overcome with experience. However, a scatterplot of individual subjects' spatial ability scores against their test session times for the command interface which gives their previous command experience, shows clearly that not all the limitations of low spatial ability are overcome by high experience. (See Figure 4). For just the high experience subjects, there is a significant Pearson product moment correlation coefficient between subjects' spatial

ability scores and their test session times ($r=-0.69$, $df=10$, $p<0.02$), showing that even for subjects with a high level of experience their performance is worse the lower their spatial ability.

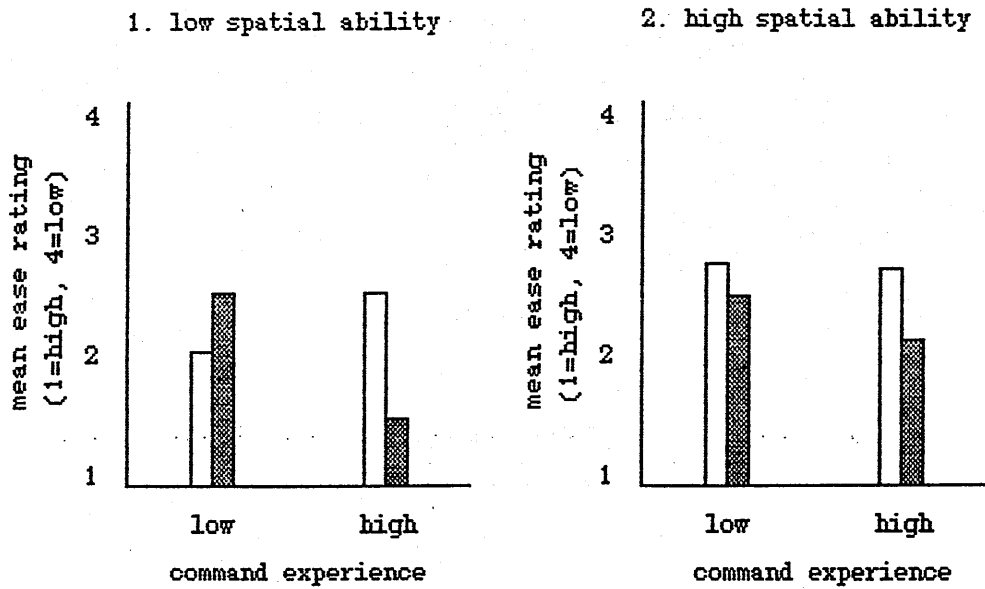
Ease and Enjoyment Ratings

Mean ease and enjoyment ratings were calculated for each interface for the six high spatial ability and high command experience subjects, for the six high spatial and low experience subjects, for the six low spatial and high experience subjects and for the six low spatial and low command experience subjects. (See Table 6 and Figure 5). Both rating scales were five point scales, with one indicating easy or enjoyable, and five indicating difficult or not enjoyable.

Table 6. Mean ease and enjoyment ratings for the command and menu interfaces for low and high spatial ability users with low and high previous command experience.

		ease rating		enjoyment rating	
spatial	command experience	command	menu	command	menu
low	low	2.5	2.0	2.2	2.8
	high	1.5	2.5	1.8	3.2
high	low	2.5	2.7	2.7	3.2
	high	2.2	2.7	2.0	3.2

Ease Ratings:



Enjoyment Ratings:

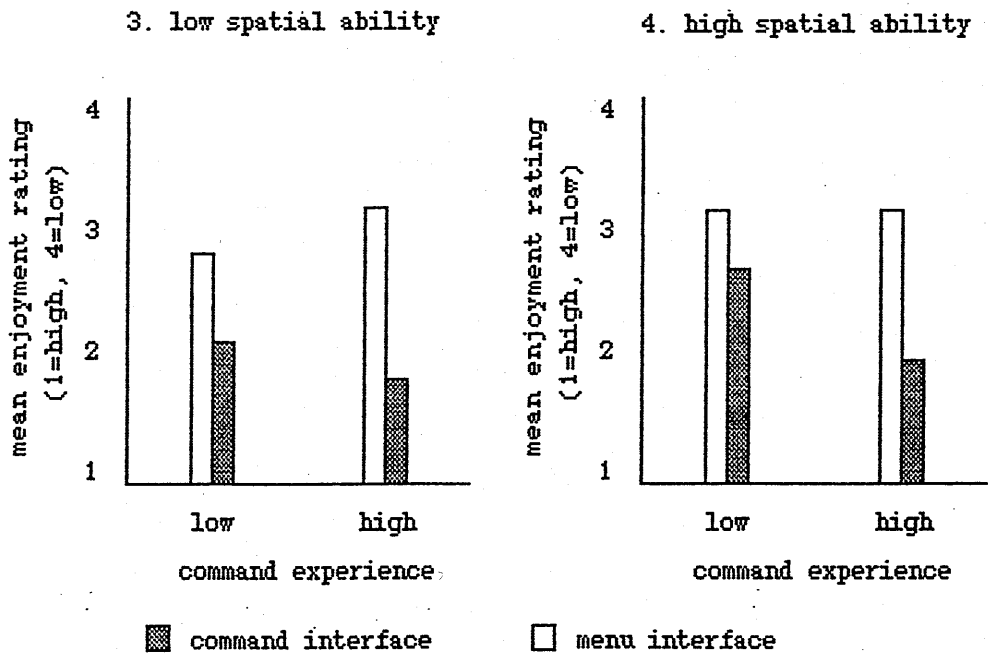


Figure 5. Mean ease and enjoyment ratings for the command and menu interfaces for low and high spatial ability subjects with low and high previous command experience.

To examine the ease ratings, separate 2(command experience)x2(type of interface) analyses of variance were carried out for the low spatial ability subjects and the high spatial ability subjects. The low spatial subjects analysis showed that there was a significant interaction between previous experience and interface style ($F(1,10)=5.87, p<0.05$). Simple effect tests for the low and high experience groups showed that the ease ratings for the two interfaces did not differ significantly for the low experience subjects ($F(1,10)=1.29, p>0.1$); but that they did differ significantly for the high experience subjects ($F(1,10)=5.17, p<0.05$). The analysis of variance for the high spatial ability subjects, showed that there was no significant interaction between previous experience and interface type, and no significant main effects of previous experience or interface type (all $F_s<1$).

The directions of the means of the ease ratings (Figure 5.1 and 5.2) followed the same pattern as the means for the test session times (Figure 3.1 and 3.2) for each interface, in terms of the within-subjects results. Although only one of the differences actually reached significance in this case, the fact that the directions were the same suggested that the results of this analysis supported the conclusions of the test session times analysis: ie. that the menu interface is better than the command interface for subjects with low spatial ability and low previous experience with using command interfaces who are unlikely to increase their experience through frequent computer use, while the command interface is better than the menu interface for the rest of the subjects.

To examine the enjoyment ratings, separate 2(command experience)x2(type of interface) analyses of variance were carried out for the low spatial

subjects and the high spatial subjects. The low subjects analysis showed that there was no significant interaction between previous experience and interface type ($F(1,10)=1.54$, $p=0.24$), there was no significant main effect of previous experience ($F<1$), but that there was a significant main effect of interface type ($F(1,10)=13.9$, $p<0.005$). The low spatial ability subjects rated the command interface as more enjoyable than the menu interface whatever their previous command experience. The analysis of variance for the high spatial ability subjects showed that there was again no significant interaction of previous experience and interface type ($F(1,10)=1.60$, $p=0.24$), there was no significant main effect of previous experience ($F<1$), but there was a significant main effect of interface type ($F(1,10)=10$, $p=0.01$). The high spatial ability subjects, like the low spatial ability subjects, rated the command interface as more enjoyable than the menu interface whatever their previous command experience.

The results of the enjoyment ratings analysis suggested that although the command interface was more difficult for the low spatial low experience subjects, the subjects actually found the command interface more enjoyable to use. This raises the issue of whether it is ease of use or enjoyment of use which should be considered to be most important when determining the most suitable dialogue style for a user, because, as shown here, they do not necessarily correspond. If enjoyment of use only is considered, the command interface appears to be more suitable than the menu interface for all the subjects. However when accuracy and efficiency are important, which is the case for most uses of database systems, ease of use is likely to be a more relevant measure.

Errors

The video records of subjects' interactions with the two interfaces were examined, and the number of errors made by subjects was counted. The mean number of errors made per subject per test session was calculated for each interface for the six high spatial and high command experience subjects, for the six high spatial and low experience subjects, for the six low spatial and high experience subjects, and for the six low spatial and low experience subjects. (See Table 7 and Figure 6). The errors counted included syntactic errors; for example formulating the basic query statement wrongly in the command interface, or missing out greater than, less than or equals signs when specifying required registration dates or exam grades for students on the menu interface. Inappropriate command errors were also counted, for example when subjects tried to type in a database query statement at the top level system prompt or the help level prompt. The other types of errors made by the subjects were typing errors, these were very few and were not counted as they were deemed to have more to do with users' typing skills than their operation of the interfaces.

Separate 2(command experience) \times 2(type of interface) analyses of variance were carried out for the low spatial ability subjects and the high spatial ability subjects. The low spatial ability subjects analysis showed that there was a significant interaction between previous experience and interface style ($F(1,10)=7.48, p<0.05$). Simple effect tests for the low and high experience groups showed that the low experience subjects produced significantly more errors for the command interface than the menu interface ($F(1,10)=19.9, p<0.001$), whereas the high experience subjects produced a similar number of errors on each interface ($F<1$). The high spatial ability

Table 7. Mean number of errors per test session with each interface for low and high spatial ability users with low and high previous command experience.

		interface	
spatial	command experience	command	menu
low	low	2.7	0.2
	high	0.5	0.2
high	low	0.8	0.2
	high	0.2	0

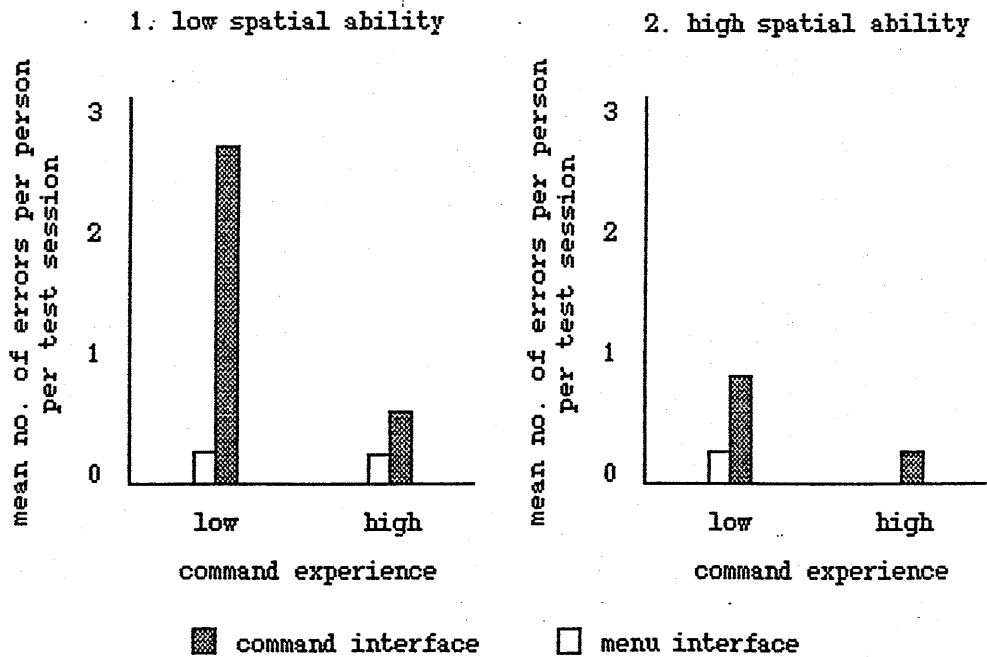


Figure 6. Mean number of errors per person per test session with each interface for low and high spatial ability users with low and high command previous experience.

subjects analysis showed that the interaction between previous experience and interaction style approached significance ($F(1,10)=3.46$, $p=0.092$).

All subject groups, apart from the high spatial high experience group, produced a few errors on the menu interface. The number of errors produced on the command interface was similar to that on the menu interface except for the low spatial low experience group who produced many more errors on the command interface than on the menu interface. The error analysis therefore supported the idea that the menu interface is more suitable than the command interface for the low spatial low experience subjects.

Comments

Subjects who rated the menu interface as being more easy or enjoyable than the command interface, made comments such as the following about the command interface:

'instructions have to be committed to memory'
'[formulating a query statement involved] translating from natural language into programming language'
'had to learn the structure of the query'

and comments such as the following about the menu interface:

'needs very little thought'
'no need to remember syntax'
'didn't let you make mistakes as much'
'prompted at each stage'

Subjects who rated the command interface as being more easy or enjoyable than the menu interface, made comments such as the following about using the command interface:

'compact ... short cuts available ...'
'able to request information in a single step'
'could recall and edit commands'

and comments such as the following about the menu interface:

'slow and repetitive, frustrating'
'laborious'
'tedious, starting [accessing information] from beginning each time'

These comments suggested that it was the freedom of navigation and the openness and flexibility of the dialogue allowed with the two interfaces which was determining whether subjects found one more easy or enjoyable to use than the other. References to short cuts and the ability to recall and edit commands for the command interface, and to the menu interface prompting at each stage and not allowing mistakes as much, all relate to the constraint of the dialogue. Comments about having to start from the beginning each time to access a new piece of information with the menu interface relate to the freedom of navigation.

The results of this experiment did show that some users were more suited to the menu interface and others to the command interface. However the relationship between which interface was right for which user did not relate simply to users' spatial ability as hypothesized in the first experiment. The results suggested that previous experience with using command dialogue styles was also an important factor and, because of this, frequency of use of computer systems, as frequency of use determines whether a user's previous experience changes or not. The results suggested that the menu interface was suitable for low spatial, low command experience users, who are occasional users and will not therefore increase their previous experience; whereas the command interface was suitable for all high spatial ability users whatever their previous experience, for low spatial high experience users, and for low spatial low experience regular users who will increase their previous experience quickly to become low spatial high experience users.

Experience with using command style interfaces may play a role as well as spatial ability in determining the suitability of command style interfaces for users, because high command experience may mean that users have already developed a mental framework for representing the structure of command systems. If users have a low spatial ability, they will not be good naturally at holding mentally a model of the system they are using, but if they have an already developed framework through experience they may be able to do so. High spatial ability users, on the other hand, can naturally easily hold a mental model of the system they are using, and so it is not

crucial whether they have a prior framework for doing this or not built up from previous experience.

It is not clear, however, whether these results are generalizable to other subject groups. It may be that a non-graduate subject group could contain some subjects with lower spatial abilities than the subjects in this graduate group. Subjects with a very low spatial ability may not be able to overcome their performance difficulties on the command interface by increasing their experience, in which case all very low spatial ability users may be better off with the menu interface than the command interface. In fact the positive correlation between performance on the command interface and spatial ability score for the high experience subjects which was found in the second experiment, supports the idea that very low spatial ability subjects will have performance difficulties on the command interface even when their experience is high. This may result in their performance on the command interface actually being worse than that on the menu interface, performance on the menu interface appearing relatively free from effects of spatial ability.

It is also not clear whether the results are generalizable to more complex uses of SQL for the command interface. The SQL statements which users were required to formulate in the command interface were relatively simple. None of the statements involved the subjects having to link two relations. If the use of SQL had been more complex in the command interface, it may be the case again that the low spatial ability subjects would not produce a good performance on this interface whatever their previous experience. One subject with a very low spatial ability but high previous command experience, who produced a faster time for the command interface than for

the menu interface, commented that if the use of SQL had been more complicated he probably would have preferred the menu interface to the command interface.

Although this experimental work has suggested which interface dialogue styles are suitable for which users of a database system, the results may only apply to this group of users and this database system. Further work still needs to be done to look at other subject groups and other database systems. It is important to look at the use of more complex database systems, as in a realistic situation task difficulties may in fact swamp users' difficulties due to the interface dialogue style (Booth, Fowler and Macaulay, 1987). In these experiments the database system tasks were purposefully kept simple so as not to interfere with any effects of dialogue style.

4.

ADAPTIVE SYSTEM SHELL

This section of the study investigates the use of a version of the prototype adaptive system shell developed by Benyon et al. (1990), to see whether it can be used successfully to produce an adaptive database system, which automatically provides the different users of the system with interfaces with dialogue styles which suit their individual characteristics. The shell is used to try to produce an adaptive version of the university database system used in the second experiment, which provides the command and menu interfaces automatically to the appropriate users.

The shell consists of several different components: a user model, adaptation rules, a dialogue record, inference rules, a domain model and a management system. The role played by each of these components is outlined below, and the system-specific information which has to be entered into each of the components to produce an adaptive version of the university database system is outlined. The next part of the report describes how the components of the shell were implemented, and shows the entering of the information into them for the university database system.

User Model

The user model stores information about users which is relevant to their use of a computer system. One user model can be constructed for each user to contain their individual characteristics, however if a group of users are very homogeneous in terms of their characteristics, a single user model could be constructed for all the users in this group which they share. Rich

(1979) suggests that user models can fall along a dimension from individual (one model per user) to canonical (one model for all users). A canonical user model can be used to provide a default user model for a group of users about whom little is known initially. Separate user models can then be constructed for the users as information is gathered about them.

The user model contains three broad sections: a 'personal profile' to contain general information about users such as their occupation, interests and previous experience with using computer systems; a 'cognitive model' to contain information such as users' personality, cognitive abilities and cognitive styles; and a 'student model' to contain information about users' knowledge of the specific computer system they are using.

To make the university database system adaptive, the user model would have to store information about users' previous experience with using command style interfaces and their frequency of use of computer systems in the personal profile, and information about their spatial ability in the cognitive model. The users' previous experience with using command style interfaces would be represented on a scale of one to five (ie. expert, high, medium, low, none) and their spatial ability as low or high. These categories follow those used in the experimental study which differentiated adequately between the users. Users' computer usage can be classified as either occasional or frequent. This is a rather crude classification, but it is sufficient to demonstrate the idea of taking a user's frequency of computer use into account.

For the university database system, one user model will be constructed for each user as the users are not homogeneous in terms of command experience

and spatial ability. However all the users will be associated with a default user model at first, to provide the users with an initial user model before information about their command experience, computer use and spatial ability can be obtained.

Adaptation Rules

The adaptation rules specify how the characteristics in the user model relate to which interface is best for a user. These rules can take the form of if... then... statements, with multiple premises and multiple conclusions.

For the university database system, the rules need to specify how users' positions on the command experience characteristic (ie. expert, high, medium, low or none), the spatial ability characteristic (ie. low or high) and the computer use characteristic (ie. occasional or frequent) relate to which out of the command and menu interfaces is the more suitable for them. If... then... rules such as the following specify this information:

if spatial ability = high

then interface = command

if spatial ability = low and command experience = none

and computer use = frequent

then interface = command

if spatial ability = low and command experience = low

and computer use = occasional

then interface = menu

Dialogue Record

The dialogue record has to contain information about users' interactions with the system, which is relevant to determining their levels on the user model characteristics. Not all the characteristics in the user model have to be determined, however, from users' interactions with the system. For certain user model characteristics, such as those in the personal profile, a user can simply be questioned about their position on the characteristics at the beginning of the interaction. In the experimental work, users were asked to rate their levels of previous experience themselves, they were not tested in any way. The cognitive model characteristics, however, cannot be obtained by questioning. In the experimental work, users were given a twenty minute written test to determine their spatial ability. Giving a user a test for each cognitive characteristic relevant to the use of a computer system would be a lengthy process, and would only be suitable in cases where the user was subsequently going to invest a lot of time in using the system. For the cognitive model characteristics, inferring users' positions unobtrusively from their interaction with the system involves the user in a lot less work, and therefore appears a much better solution.

Previous research has suggested that this type of inference is possible. Vicente and Williges (1988) showed that users' levels of spatial ability related to their frequency of use of particular commands when interacting with a hierarchical file system. Therefore to allow spatial ability to be inferred in this case, the dialogue record would have to contain a log of the number and types of commands used to carry out a number of tasks. Only certain information is suitable for use in the dialogue record. For example, using the time users take to carry out tasks to infer characteristics is

unlikely to be reliable, as under uncontrolled conditions users may simply stop half way through a task to do or think about something else. The system would not be able easily to distinguish between this and the user being stuck at this point.

For the university database system, the dialogue record has to contain information about users' interactions with the system which is relevant to determining their level of spatial ability. In the experimental study, five out of the six users with a low spatial ability who rated their previous command experience as being low or none (four or five), made two or more errors (excluding spelling errors) during the session of twelve tasks, the sixth making one error. Twenty three out of the twenty four remaining users, who did not have a low spatial ability and a low or no command experience, made one or no errors during the session, the twenty fourth making two errors. Therefore, if a record is kept of the number of non-spelling errors which are made by the users, and the number of tasks they have completed for the command interface, the number of errors per number of tasks can be used to place correctly twenty eight out of the thirty users who took part in the experiment in terms of whether they have a low spatial ability and a low or no command experience or not. The number of errors per number of tasks does not determine spatial ability alone, it only determines whether or not users have both a low spatial ability and a low or no command experience. However for the university database system, this distinction is sufficient as all users who do not have a low spatial ability and a low or no command experience are suited to the command interface, whether they are high spatial users, or low spatial users with a high command experience. For this system it is important only to identify users as having a low spatial ability if they have a low or no command experience as well.

To allow the dialogue record to be constructed while a user is interacting with a system, functions have to be added to the system to update the dialogue record. For example, for the university database system, functions would have to be added to increment the error count each time the user makes an error, and to increment the task number count each time the user successfully completes a task.

Inference Rules

The inference rules specify how the information in the dialogue record relates to users' levels on the user model characteristics. These rules, like the adaptation rules, take the form of if... then... statements with multiple premises and multiple conclusions.

For the university database system, the rules need to specify the fact that a user's spatial ability and command experience can be taken as both being low if the user makes more than one error when completing twelve tasks using the command interface. This can be specified using the following if... then... rule:

```
if interface = command and tasks = 12 and errors > 1
then spatial ability = low and command experience = low
```

Domain Model

The domain model contains a formal description of the computer system for which the shell is being used. The student model refers to users' level of knowledge of the attributes of the system as outlined in the domain model.

The adaptation rules also relate the user characteristics to which of the attributes of the system suit which user.

For the university database system, the domain model can be very simple. The university database system simply consists of two complete alternative interfaces: the command and menu interfaces.

Management System

The management system coordinates the other components of the shell to present the appropriate interface to each user. The management system allows a user to inspect their user model, so that they can enter information into the personal profile, and see any user model values which have been inferred by the system. A user therefore has access to any information which is stored about them, and is free to change this information. The management system activates the adaptation rules to calculate which interface is best for a user given the characteristics in their user model, and presents this interface to the user. Each time information is added to the dialogue record during the users' interaction with the system, the inference rules are activated to see if anything can be inferred about the user from their dialogue record. When a value in the user model is added or altered as a result of the inference rules, the adaptation rules are again activated to see if a different interface is now calculated as being more suitable for the user. If a different interface is suggested, the management system informs the user. At this point the user can agree to accept the change, or can gain access to their user model and alter any system generated values they do not agree with, and the best interface for them is then recalculated.

The user is consulted about changes to the interface, to try to avoid annoying or confusing the user. If the system simply changed the interface while a user was interacting with the system, this could make it hard for the user to develop a coherent model of the system (Greenberg and Witten, 1985), and could induce negative feelings of loss of control (Morcio, 1989). Users could attempt to disguise characteristics about themselves from the system (Wahlster and Kobsa, 1986) to try to stop or to influence changes taking place to the interface. Giving the user some say in the changes made to the interface should allay these problems.

The adaptive system shell was implemented in KEE on a SUN workstation. KEE provides a support environment for the development of object oriented systems. KEE provides the framework for specifying objects (called units in KEE), and for specifying the attributes of these objects (called slots). The slots can contain two kinds of information: descriptive information to specify the facts about an object, and procedural information to specify the behaviour of an object. KEE also offers several special facilities for manipulating objects. These facilities make KEE ideally suited as a development tool for the adaptive system shell, and will be discussed in detail where they have relevance to the components of the shell.

This section explains how each component of the shell was implemented in KEE, and shows information being entered into the components for the university database system to produce an adaptive version of this system. The student model and the domain model have not been implemented here, as the personal profile and cognitive model sections of the user model are sufficient to store all the information necessary about users for the purposes of the university database system; and the domain model is so simple in this case.

User Model

The user model has to store characteristics about each of the users. To do this, an object is constructed to represent each of the characteristics entered in the user model. For example, if spatial ability is entered as a

characteristic in the user model, an object is created to represent spatial ability. These objects representing characteristics each have a slot which contains the value of the characteristic for a user. For example, the object spatial ability would contain a slot which would store either the value 'high' or the value 'low' for a user. KEE offers a facility which allows the same object to exist in different 'worlds', where the attributes of the object can differ. This facility is used for the user model, as the value of a user model characteristic is different for different users. Only one object is set up to represent each user model characteristic, but a different world is created for each user (or homogeneous group of users) so that the user model characteristic can have different attributes for the different users. For example, the object representing spatial ability could exist in two worlds, one for one user where its value is low, and one for another user where its value is high. KEE worlds provide a good method of storing the user model information, as all the users share the same basic user model characteristics but have different values on these characteristics. The use of KEE worlds provides a more economical way of storing the user model information than setting up a separate object to represent each user.

Figure 7 (in Appendix 12) shows what the user model component of the shell looks like. Each of the screens is explained below. Figure 7.1 shows the shell menu, from which the user model component of the shell can be selected by clicking on the 'user models' button. The initial user model screen is shown in figure 7.2. The generic user model is the user model which gives the general format of the user models for all the users of a system. This has to be set up according to the user characteristics of relevance for a system. Figure 7.3 shows the generic user model being selected to look at,

by clicking on the 'view/edit user model' button and specifying the generic user model. The generic user model is initially blank as shown in figure 7.4, and the 'add characteristic' buttons are used to enter user model characteristics into the personal profile and cognitive model. Figures 7.5 to 7.7 show characteristics being added which apply to the university database system. The name of each user model characteristic and the range of values each user model characteristic can take are specified, and these are displayed on the generic user model screen.

To set up the user models for each of the users the 'add user model' button is used. Figures 7.8 and 7.9 show a default user model being set up, and figure 7.10 shows individual user models being set up which are associated with this default user model. The user models are displayed on the screen, linked to the default to which they are associated. Figure 7.11 shows the default user model being selected to look at. This user model (figure 7.12) has the same format as the generic user model, but here values can be chosen for the user model characteristics by clicking on the names of the required values and then on the 'enter values' button. The names of the chosen values are highlighted. In this case the value of spatial ability is set to high in the default. As all the users are associated with this default, the value of spatial ability in their individual user models is automatically set to high. This can be seen in figures 7.13 and 7.14, where the user model for an individual user is selected to look at. Additional information relevant only to that particular user can be added at this point, for example values for the personal profile characteristics (figure 7.15). Again the chosen values are highlighted. The default spatial ability has been set to high in this case so that all the users are initially presented with the command interface to the university database system. The users' error rate for this

interface is looked at, so that the users who in fact have a low spatial ability and a low or no command experience can be recognized and be changed to the menu interface.

The user model component allows user models to be deleted (figure 7.16), for example if a user stops using a system. The name of the deleted user is removed from the screen (figure 7.17). The range of possible values for a user model characteristic can be edited in the generic user model (figures 7.18 and 7.19), for example if the range of values is found not to be wide enough. This changes all the user models, for example figure 7.20 shows the user model for an individual user with the new choice of 'rare' added as a value for the computer use characteristic. The chosen value of 'frequent' for this user for this characteristic is maintained as this value still appears as an option in the newly specified range of possible values for the computer use characteristic. User model characteristics can also be deleted (figures 7.21 and 7.22), for example if they are found not to be relevant to the best interface for a user. Again this changes all the user models; figure 7.23 shows how this characteristic has disappeared from the user model for an individual user.

Adaptation Rules

KEE provides a framework to set up rules to reason with objects and their attributes. The rule framework supports the production and execution of rules of the following form:

if the <attribute> of <object 1> is <attribute value>
then the <attribute> of <object 2> is <attribute value>

The rule can have multiple premises and multiple conclusions, which are automatically taken as being connected by 'and' operators. When a rule is executed, if the premises prove true according to the objects and attributes they refer to, then the conclusion succeeds, and the attributes of the objects named in the conclusions are automatically assigned the attribute values specified. KEE provides a weighting system for the rules, so that when a number of rules are activated, if any of the rules result in conflicting conclusions, the rule can be specified which should succeed.

Each adaptation rule is written as a KEE rule, for example:

```
if the value of spatial is high
then the interface of user is command
```

The premises refer to the values of the user model characteristics, and the conclusions insert a value into an object representing the user with an attribute representing the best interface for that user. The values of the user model characteristics are stored in different worlds for each user. KEE provides the facility to allow rules to operate in specific worlds; therefore to calculate the best interface for a user, the rules are activated in that user's world, so that the premises refer to the user model characteristic values for that user.

Figure 8 (in Appendix 12) shows what the adaptation rules section of the shell looks like. The initial adaptation rules screen is shown in figure 8.1. To add a rule, the 'add rule' button is clicked on. The name of the rule, the weighting of the rule and the rule itself are then entered. The name of the rule is displayed on the screen. Figures 8.2 and 8.3 show the entering of a rule for the university database system specifying that the command interface is best for users with a high spatial ability, and figures

8.4 and 8.5 show the entering of a rule specifying that the menu interface is best for users with a low spatial ability and no command experience who are occasional computer users. The rules are given equal weighting for the university database system, as none of the conclusions can conflict.

A rule can be viewed or edited by clicking on the 'view/edit rule' button. The rule is displayed and can be altered if required (figure 8.6). A rule can also be deleted using the 'delete rule' button (figure 8.7). The name of the deleted rule is removed from the screen (figure 8.8).

Dialogue Record

For the dialogue record, objects and attributes are constructed to represent each characteristic of the dialogue which is to be recorded. For example an object 'task' is constructed with an attribute 'number' if the number of tasks completed is to be recorded. The dialogue records are different for different users, so these objects again exist in different worlds for different users, so that they can hold different values. For example, one user may have only completed two tasks, whereas another user may have completed ten. The values of the attributes are incremented by code written into the university database system, for example the number attribute of the object task is incremented by one each time a task is completed by the user.

Figure 9 (in Appendix 12) shows what the dialogue record component of the shell looks like. Figure 9.1 shows the initial dialogue record screen. To add a record, the 'add record' button is clicked on. The characteristic of the dialogue to be recorded is then entered, and is displayed in the screen (figures 9.2 and 9.3). Figure 9.4 shows the screen when all the dialogue

record characteristics have been entered which are necessary for the university database system. The dialogue record for a particular user can be viewed by clicking on the 'view record' button, and specifying the user whose dialogue record is required (figure 9.5). A dialogue record characteristic can also be deleted by clicking on the 'delete record' button and specifying the name of the characteristic to be deleted (figure 9.6). The name of the deleted characteristic is removed from the screen (figure 9.7).

Inference Rules

The inference rules component of the shell, like the adaptation rules component, makes use of the KEE rule framework. Each inference rule is written as a KEE rule. For example:

```
if the name of interface is command
    the number of tasks is 12
    the number of errors is >1
then the value of spatial is low
    the value of command experience is low
```

The premises of the rules refer to the values of the dialogue record characteristics, and the conclusions insert values into the user model characteristics. The values of the dialogue record characteristics, like the values of the user model characteristics, are stored in different worlds for different users. These rules, like the adaptation rules, are therefore activated in the specific world of the user who is being considered.

Figure 10 (in Appendix 12) shows what the inference rules section of the shell looks like. Figure 10.1 shows the initial inference rules screen. To add a rule, the 'add rule' button is clicked on. The name of the rule, the weighting of the rule and the rule itself are entered (figures 10.2 and 10.3). The name of the rule is displayed on the screen. A rule can be viewed or edited by clicking on the 'view/edit rule' button. The rule is displayed and can be altered as required (figure 10.4). A rule can also be deleted using the 'delete rule' button (figure 10.5). The name of the deleted rule is removed from the screen (figure 10.6).

Management System

The management system links the above four components. The management system makes use of what are called 'active values' within KEE. Active values can be used to 'watch' the attributes of certain objects; if the value of the attribute is changed the active value causes a secondary action to take place. Active values are used to link the components of the shell. When a user is interacting with the database system, and a change is made to the user's dialogue record (for example, the number of tasks completed is incremented by one), an active value linked to the dialogue record characteristic is activated and this causes a secondary action of activating the adaptation rules. If the adaptation rules result in the value of a user model characteristic being altered, the inference rules are activated to calculate if a different interface is required for the user. If the value of the appropriate interface for a user is changed, this activates an object within the management system which manages the negotiation with the user about a change to the interface.

Figure 11 (in Appendix 12) shows the management system in operation. When the user comes to the system, they enter their user name to enter the system (figure 11.1). The user is then asked if they wish to view or edit their user model (figure 11.2). The user model is displayed with any values which have already been entered (figure 11.3). In this case the user's spatial ability is set to high as this was set in the default user model. The user can then enter their values for the personal profile characteristics (figure 11.4). When the user clicks on the 'continue' button, they are then presented with the interface to the database system which has been calculated as being appropriate for them (figure 11.5), and the user can use this system (figure 11.6). If the user makes more than one error in the completion of the twelve tasks, so that the inference rules change values in the user's user model, the user is informed that this change has occurred and is given the option to view their user model and change the system generated values if they wish (figure 11.7). Figure 11.8 shows how the values have been changed for the user to low spatial ability and low command experience. If the user does not alter these values, and clicks to continue the user is presented with the new menu interface (figures 11.9 and 11.10). However, if the user changes their command experience back to medium before clicking to continue (figure 11.11), they are returned to the command interface at the point where they were interrupted (figure 11.12).

This section of the study investigated whether some type of adaptive system shell could be used to produce an adaptive database system, which provides users of the system with interfaces with different dialogue styles to suit the users' individual characteristics. The use of a prototype shell to produce an adaptive version of the university database system, suggested that this could be possible.

The adaptive version of the university database system was however only a demonstration system, it was not actually tried out on real users. Many refinements would be needed to the characteristics and rules entered into the shell for the university database system if it was to be properly operational.

For example, for this demonstration, the computer use characteristic was only given two levels: occasional or frequent. The computer use characteristic in fact needs to be a lot more detailed than this. The characteristic needs more levels, and account needs to be taken of the nature of users' computer use. Some users who use a computer system frequently, may nevertheless gain very little experience if they only use a very limited number of the system facilities each time they use the system. As the computer use characteristic is being used here as a gauge of whether users' will increase their computer experience rapidly or not, the nature of the computer use is an important factor to consider as well.

The inference rule which was written for the university database system, to infer from an interaction whether a user was low on both spatial ability and

command experience, was also over-simplified and needs refinement. The rule was based on the results of the experimental work, where users were each given twelve tasks to carry out using the interfaces. These tasks were matched in terms of complexity, whereas under uncontrolled conditions where a user is choosing which tasks to carry out using the system, twelve tasks carried out by one user may not match the complexity of twelve tasks carried out by another user. As the complexity of tasks, as well as the number of tasks carried out, is likely to relate to the number of errors made, where users are carrying out tasks of different complexities the inference rule needs to take the complexity of the tasks carried out by users as well as their number into account.

Also the actual use of error rate as the basis for the inference rule may need to be re-examined. For the university database system, looking at the number of errors on the command interface allowed twenty eight out of the thirty users who took part in the experiment to be classified correctly in terms of spatial ability and command experience. However, two users were incorrectly classified; and as there was not a large difference in the numbers of errors made by the users with a low spatial ability and a low command experience and those without, there is no guarantee that the proportion of users classified correctly would be as large for another group of users. Also, under uncontrolled, non-experimental conditions, fluctuating characteristics such as attention may affect error rate.

The system would also have to be tested in a real situation to see if the way in which the system handles changes to the interface is satisfactory for the users, for instance whether it avoids negative feelings of loss of control and allows the user to form a coherent mental picture of the system.

5.

GENERAL DISCUSSION

This study investigated whether different users of database systems do suit interfaces with different dialogue styles, and tried to identify which dialogue styles with which characteristics suit users with which characteristics. The results of the experimental work suggested that an interface dialogue style which aids navigation and which constrains the dialogue is appropriate for users with both a low spatial ability and low experience of command style interfaces; and a dialogue style which does not aid navigation and which allows an open and flexible dialogue is appropriate for users with a high spatial ability, and for users with a low spatial ability but high experience of command style interfaces. However, as pointed out, further work needs to be done to see if these findings do hold for other groups of users and for database systems other than those used in the experiments.

This study has spoken of the dialogue style for a database system interface as if it can be considered in isolation from the database which the interface forms part of. This study looked at the access of information from databases only, however data entry is also an important issue. The type of data entry which a database system needs to support may in fact limit the interface dialogue styles which are possible for the system. If the entry of data into a database involves the entry of new relations, not just new records, SQL would be a suitable dialogue style as it supports the easy entry of new relations. However a menu style interface would not be suitable, as menu options would have to be actually altered on the menu interface to accommodate new relations.

The study also looked at dealing with the implications of the suggestion that different users of a database system require interfaces with different

dialogue styles. The study looked at whether it was possible to produce an adaptive database system, which automatically presented the different users of the system with the dialogue styles which suited them. To do this, the system has to gather information about the users, and relate this information to the interface dialogue style which is best for a user. The use of a prototype shell for the development of adaptive systems to produce a demonstration adaptive version of the university database system, suggested that such a way of presenting different dialogue styles to different users could be possible. Further work is however needed to examine which aspects of users' dialogues with a computer system can be used reliably to infer information about the users, especially when users are working under natural rather than experimental conditions.

The latter part of the study provided some support for the potential success of the prototype adaptive system shell for presenting interfaces with different dialogue styles to different users of a database system. However a shell needs to be capable of providing users with interfaces which suit users on many different aspects, not just dialogue style. Many aspects of a database system interface could be tailored to suit different users, such as the tasks supported by the interface, the accessibility of certain information in the database, the terms used for database items and operations, and the level of on-line help provided to users. Suggestions about how the shell could deal with these are considered below, but they need to be tested in practice.

To provide users with an interface supporting the range of tasks they require to carry out using a system, information about users' occupations and interests could be stored in the personal profile section of the user

model. Adaptation rules could then be written to link different occupations and interests with the tasks users require; for example, a rule stating that a library customer needs an interface through which they can access book details but not enter book details, whereas a member of the library staff needs an interface through which they can do both. Adaptation rules could also be written to link users' occupations and interests to the suitable availability of information for users. For example a rule could be written stating that users who are interested primarily in books and journals about psychology, need an interface which makes the information about books and journals in this subject area most readily accessible. To provide users with appropriate terms for database items and operations, the personal profile could contain information about users' preferred terms for these. As all the information for the personal profile can simply be requested from the user, there is no problem of inference rules to obtain the information.

For the provision of suitable on-line help to users, the student model contains information about users' knowledge of the system they are using, and the cognitive model can contain information about users' preferred learning styles and strategies. Adaptation rules could link users' levels of knowledge from the student model with the level of help system they need, and users' learning styles and strategies to the style of help system they need. The domain model provides the information about the system on which the content of the help can be based. Inference rules would have to be written to infer users' knowledge about the system and their learning styles and strategies from their interactions with the system. If the on-line help is to be geared to help users with specific goals they are attempting using the system, users' goals would also have to be determined. The architecture developed for the prototype version of the shell may not be complex enough

to handle inferences such as these, and the feasibility of making complex inferences such as these from users' dialogues with a system needs to be thoroughly examined.

Edmonds (1987) suggested that users' goals cannot be adapted to automatically in this way, but that user goals could be adapted to by allowing users to customise the system in some way to suit their own goals. This would mean that inferring users' goals unobtrusively would be unnecessary. Attempts have been made to build adaptive systems which are capable of adapting automatically to suit users' goals, for example an adaptive version of the Telecom Gold electronic mail system (Totterdell and Cooper, 1986) which was developed as part of the Adaptive Intelligent Dialogues (AID) project.

This system was designed to allow users' goals to be inferred. The system was designed to support the recognition of seven distinct goals, by comparing users' actions (ie. the commands they input) with sequences of actions necessary to carry out the goals. If the sequence of user actions matched the sequence of actions necessary to carry out a goal, it could be inferred that the user was attempting this goal. However, when this system was tested out on real users, it was found that inappropriate inferences were made as to users' intentions. It was also pointed out that modelling at this level of detail has high overheads in implementation effort and speed of the system.

Even though a shell can help with the development of an adaptive system, by providing the mechanisms for presenting different interfaces to different users, a large amount of research has to be done initially to determine

which interfaces are needed to suit which users. This is a lengthy process. The amount of research necessary could be reduced by compiling a knowledge base of any information which has already been discovered about how user characteristics relate to the interface design which is best for a user. This knowledge base could then be used to help in the design of the alternative interfaces necessary to suit the users of a system. A useful addition to the shell would be a mechanism to allow the system to assess whether any changes it has made to an interface have in fact improved user performance (Benyon et al., 1990). This self-evaluation mechanism could then automatically modify any adaptation rules which appeared faulty. This would allow the shell to generate information about which interface is best for which user, which could be fed back into the knowledge base of how user characteristics relate to the best interface for a user. The shell could therefore help in determining which type of interface suits which type of user, not just in presenting the appropriate interface to the appropriate user.

Further work needs to be carried out to test the prototype adaptive system produced here against a system in which users are able to choose between the two interfaces for themselves, to see if users do prefer an automatically adaptive system. The success of systems such as the Apple Macintosh, which frequently offer the user a choice of operating an application using commands and function keys or a mouse and menu, suggests that users may be happy with making the choice between alternative interface dialogue styles for themselves. However, if this is the case, the shell would still be useful as a system development tool for helping to determine the types of interfaces necessary to suit a range of users, by using the shell to try out different interfaces on test users and assess their suitability.

References

Abstract Ability Test. MFER-NELSON, ASE Division.

Benyon, D., Milan, S. and Murray, D. (1987). Modelling users' cognitive abilities in an adaptive system. Paper presented at the fifth Empirical Foundations of Software Sciences (EFISS) conference, Risø National Laboratory, Denmark.

Benyon, D., Murray, D. and Jennings, F. (1990). An adaptive system developer's tool-kit. In Diaper, D. ed., Interact '90.

Booth, B., Fowler, C. and Macaulay, L. (1987). An investigation into business information presentation at human-computer interfaces. In Bullinger, H.J. and Shakel, B. eds., Interact '87. North-Holland.

Davis, J.S. (1989). Usability of SQL and menus for database query. Int. J. Man-Machine Studies, 30, 447-455.

Edmonds, E.A. (1987). Editorial: Adaptation, response and knowledge. Knowledge-Based Systems, 1(1).

Egan, D.E. (1988). Individual differences in human-computer interaction. In Helander, M. ed., Handbook of Human-Computer Interaction. Elsevier.

Egan, D.E. and Gomez, L.M. (1985). Assaying, isolating and accommodating individual differences in learning a complex skill. In Dillon, R.F. ed., Individual Differences in Cognition 2. New York: Academic Press.

Everest, G.C. (1986). Database Management: Objectives, System Functions and Administration. McGraw-Hill.

Fowler, C.J.H., Macaulay, L.A. and Siripoksup, S. (1987). An evaluation of the effectiveness of the adaptive interface module (AIM) in matching dialogues to users. In Diaper, D. and Winder, R. eds. People and Computers 3. Cambridge University Press.

Fowler, C.J.H. and Murray, D. (1988). Gender and cognitive style differences at the human-computer interface. In Bullinger, H.J. and Shakel, B. eds., Interact '87. North-Holland.

Garceau, S., Oral, M. and Rahn, R.J. (1988). The influence of data presentation mode on strategic decision making performance. Computer Operations Research, 15, 479-488.

Greenberg, S. and Witten, I. H. (1985). Adaptive personalized interfaces: a question of viability, Behaviour and Information Technology, 4(1), 31-45.

Myers Briggs Type Indicator, Oxford Psychologists Press.

Norcio, A.F. (1989). Adaptive Human-Computer Interfaces: A Literature Survey and Perspective. IEEE Transactions on Systems, Man and Cybernetics, 19(2).

Rich, E. (1979). User modelling via stereotypes. Cognitive Science, 3, 329-354.

Robertson, I.T. (1985). Human information-processing strategies and style. Behaviour and Information Technology, 4(1), 19-29.

Rogers, Y. (1990). Evaluation of the user interfaces for NPL project. NPL Technical Memorandum.

Schneiderman, B. (1986). Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley.

Spatial Ability Test ST7. Saville and Holdsworth Ltd.

Totterdell, P. and Cooper, P. (1986). Design and evaluation of the AID adaptive front-end to Telecom Gold. In Harrison, M.D. and Monk, A.F. eds. People and Computers: Designing for Usability. Cambridge University Press.

Van der Veer, G.C. (1990). Human-Computer Interaction: Learning, Individual Differences and Design Recommendations. Haveka B.V.

Van der Veer, G.C., Tauber, M., Wearn, Y. and van Muijlwijk, B. (1985). On the interaction between system and user characteristics. Behaviour and Information Technology, 4(4), 289-308.

Verbal Ability Test. NFER-NELSON, ASE Division.

Vicente, K.J., Hayes, B.C. and Williges, R.C. (1987). Assaying and isolating individual differences in searching a hierarchical file system. Human Factors, 29, 349-359.

Vicente, K.J. and Williges, R.C. (1988). Accommodating individual differences in searching a hierarchical file system. International Journal of Man Machine Studies, 29, 647-668.

Wahlster, W. and Kobsa, A. (1986). Dialogue-based user models. Cited in Norcio (1989).

Wearn, Y. (1989). Cognitive Aspects of Computer Supported Tasks. Wiley.

Appendices

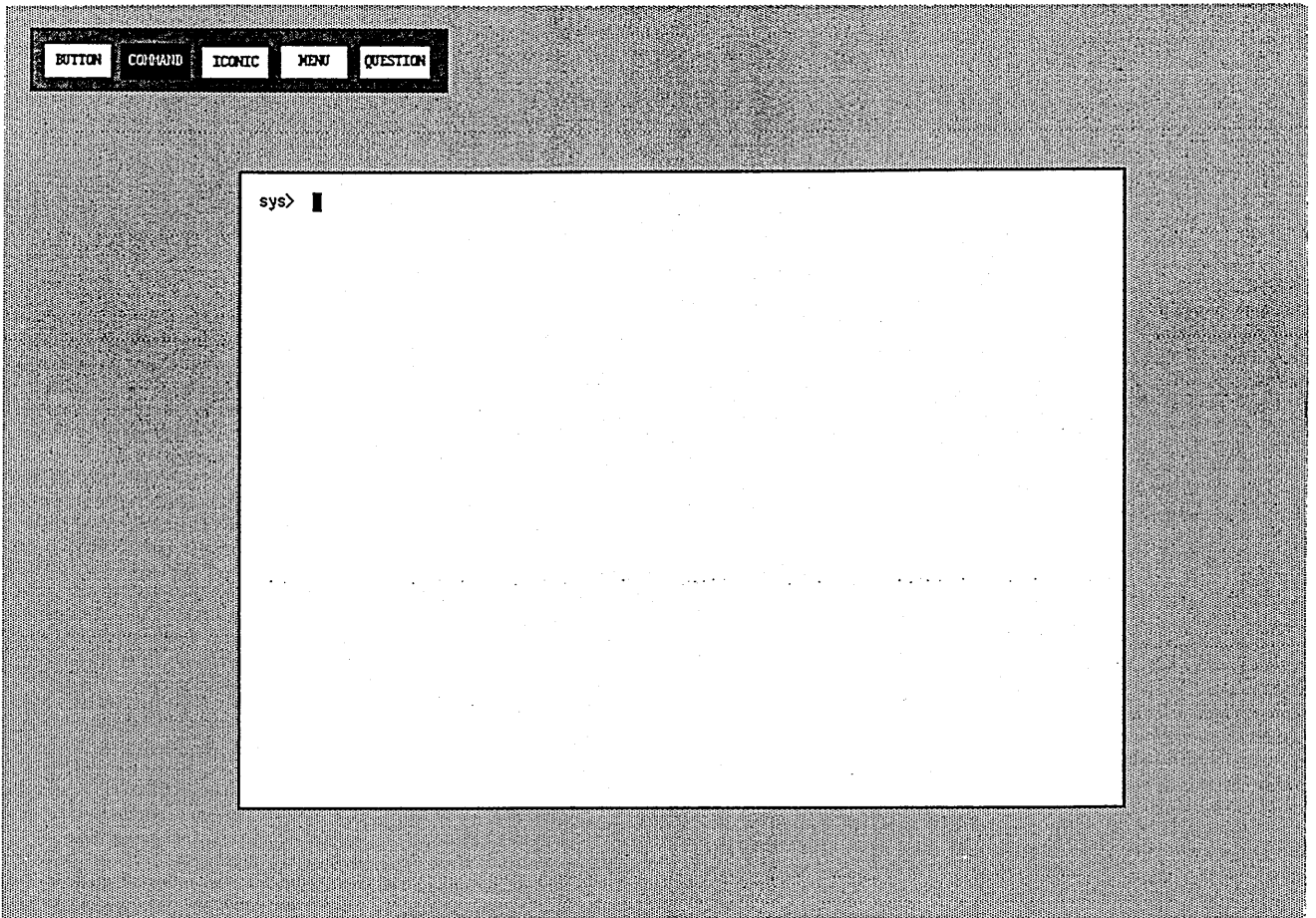
1. Experiment One: Test System	94
2. Experiment One: Computing Experience Questionnaire	111
3. Experiment One: Example Test Session Tasks	113
4. Experiment One: Instruction Sheet	115
5. Experiment One: Raw Results	116
6. Experiment Two: Test System	124
7. Experiment Two: Computing Experience Questionnaire	133
8. Experiment Two: Example Test Session Tasks	134
9. Experiment Two: Instruction Sheet	136
10. Experiment Two: Raw Results	137
11. Experiment Two: ANOVA Summary Tables	141
12. Adaptive System Shell: Figures 7-11	144

Appendix 1

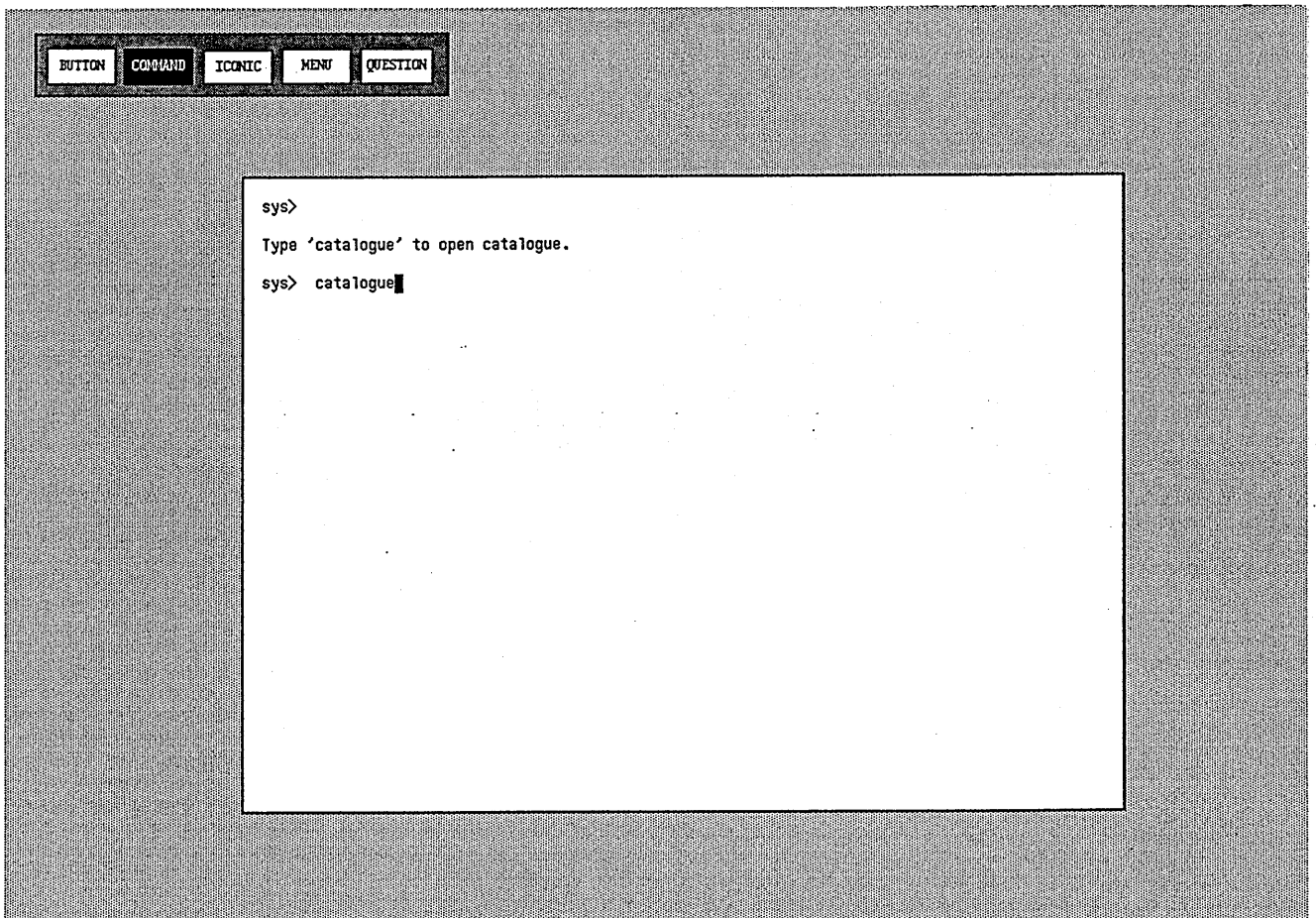
Experiment One: Test System.

1. command interface
2. question interface
3. menu interface
4. button interface
5. iconic interface

1. command interface



1.1



1.2

```
ctg>  
Type 'help' to enter help facility.  
Type 'close' to exit catalogue.  
ctg> help
```

1.3

```
HELP  
Enter the name of the item type you are interested in  
(one of the following list)  
  
blouses  
carpets  
curtains  
dresses  
fridges  
hifis  
mens_jeans  
mens_tshirts  
shirts  
skirts  
suits  
televisions  
vacuum_cleaners  
washing_machines  
womens_jeans  
womens_tshirts  
  
hlp> vacuum_cleaners
```

1.4

BUTTON

COMMAND

ICONIC

MENU

QUESTION

HELP

Item specification must be of the following form:
(vacuum_cleaners <cost> <colour> <power>)

<cost> is the cost required in pounds,
and can be one of the following:
<100 <150

<colour> is the colour required,
and can be one of the following:
white grey black

<power> is the power required in watts,
and can be one of the following:
800 1000 1200

Example: (vacuum_cleaners <100 grey 1000)

hlp>

Invalid item type. Please respecify.
Or type 'quit' to exit help facility.

hlp> quit

1.5

BUTTON

COMMAND

ICONIC

MENU

QUESTION

ctg> (vacuum_cleaners <100 grey 1000)

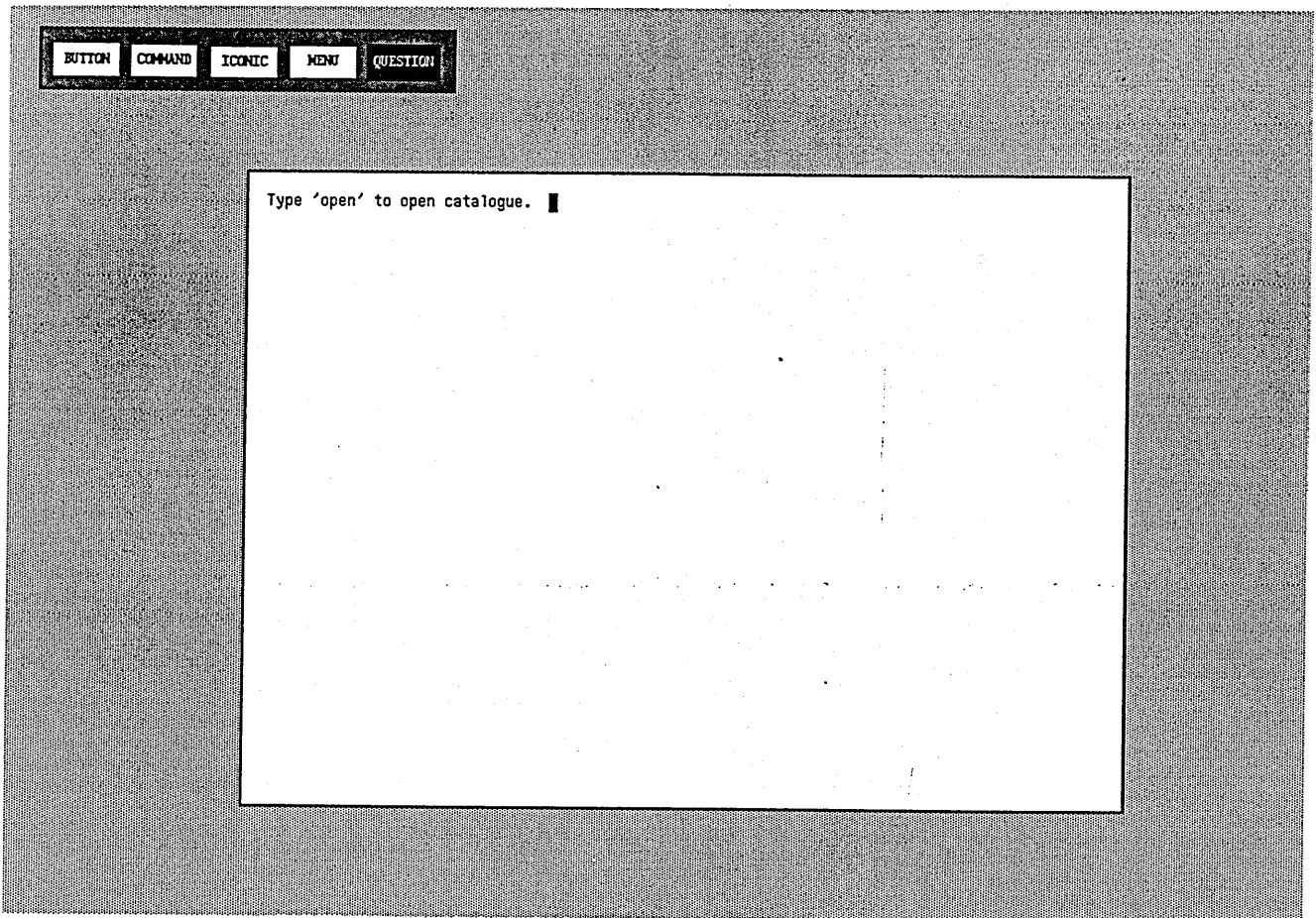
1.6

List of available vacuum cleaners:

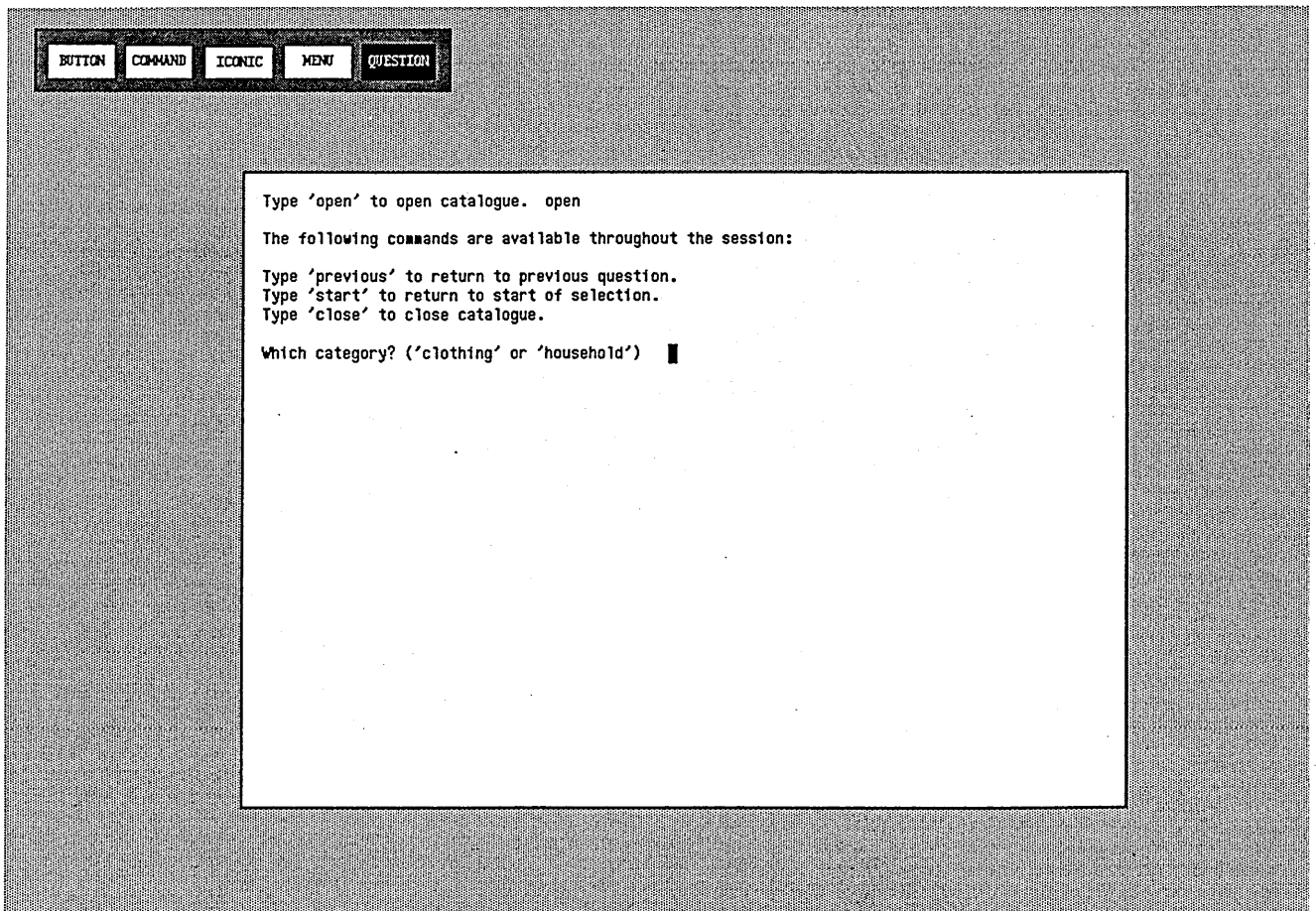
CAT. NO.	DESCRIPTION	COLOUR	POWER (WATTS)	COST (POUNDS)
P88851	(ELECTRONIC CYLINDER)	GREY	1000	89.99

ctg> █

2. question interface



2.1



2.2

Type 'open' to open catalogue. open

The following commands are available throughout the session:

Type 'previous' to return to previous question.
Type 'start' to return to start of selection.
Type 'close' to close catalogue.

Which category? ('clothing' or 'household') household

Which category? ('audio', 'electrics' or 'furnishings') electrics

Which category? ('fridges', 'vacuum-cleaners' or 'washing-machines') vacuum-cleaners

Which cost category (in pounds)? ('<100' or '<150') <100

Which colour? ('white', 'grey' or 'black') grey

Which power (in watts)? ('800', '1000' or '1200') 1000

2.3

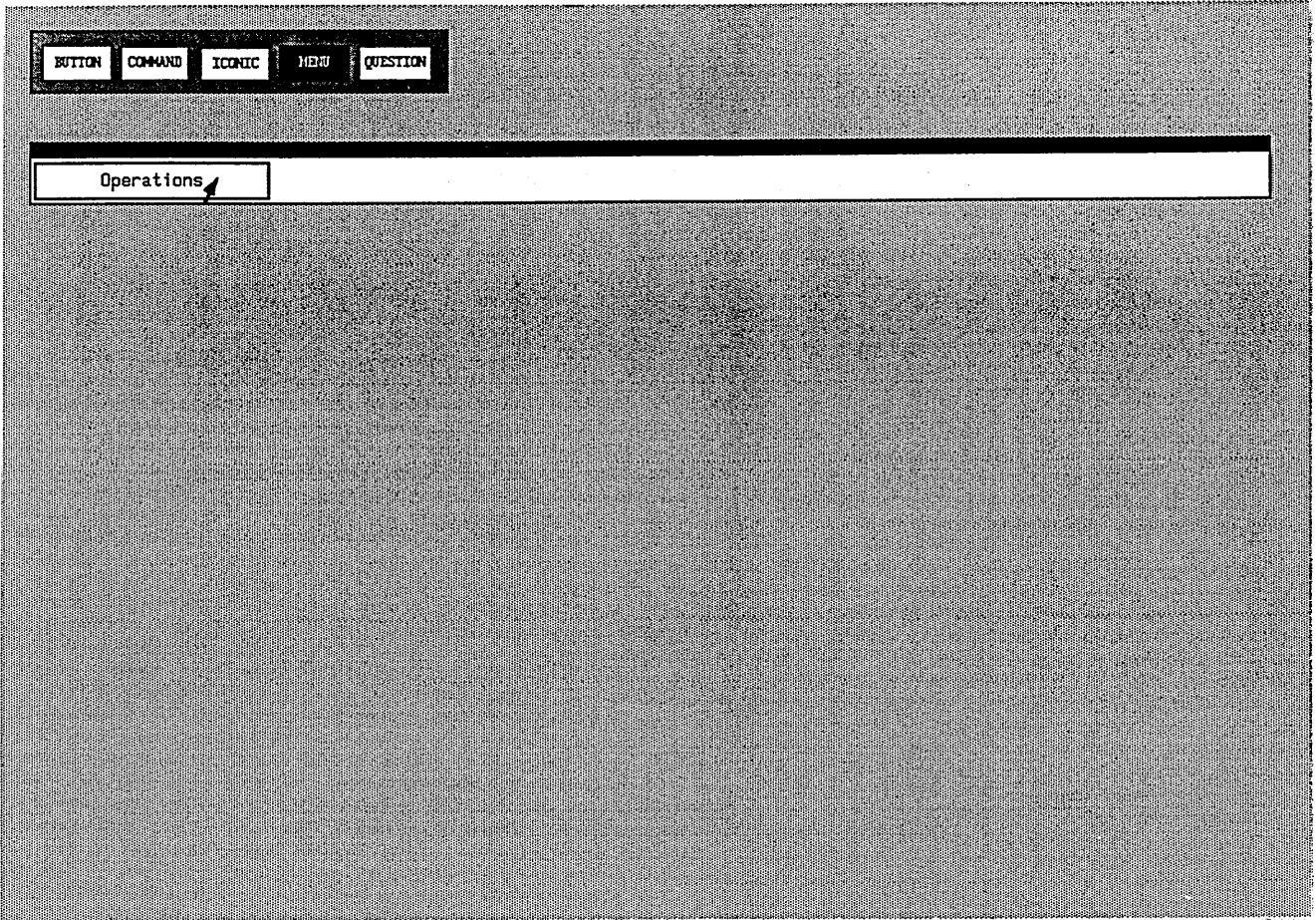
List of available vacuum cleaners:

CAT. NO.	DESCRIPTION	COLOUR	POWER (WATTS)	COST (POUNDS)
PB0051	(ELECTRONIC CYLINDER)	GREY	1000	89.99

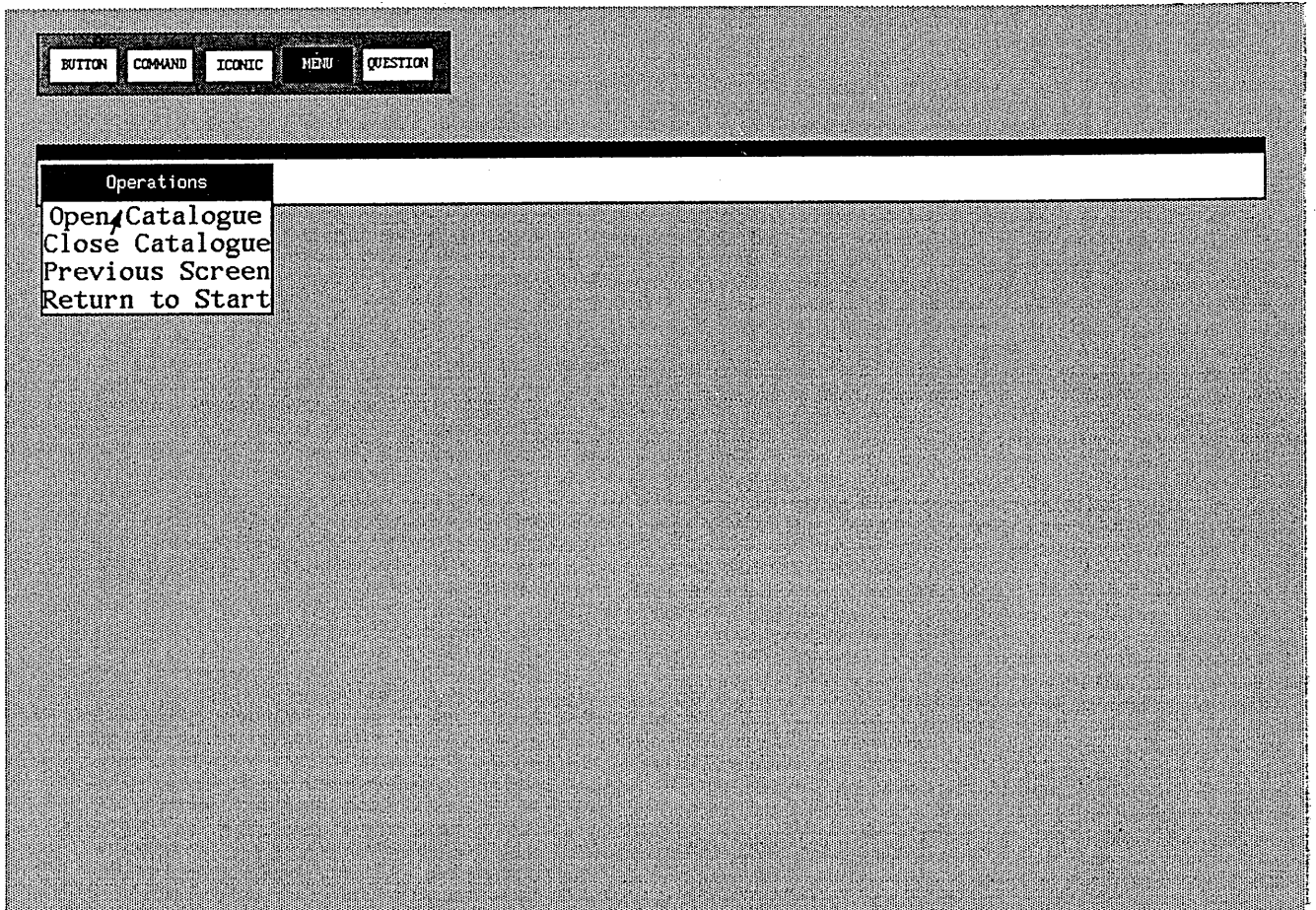
Use 'close', 'start' and 'previous' commands as appropriate. █

2.4

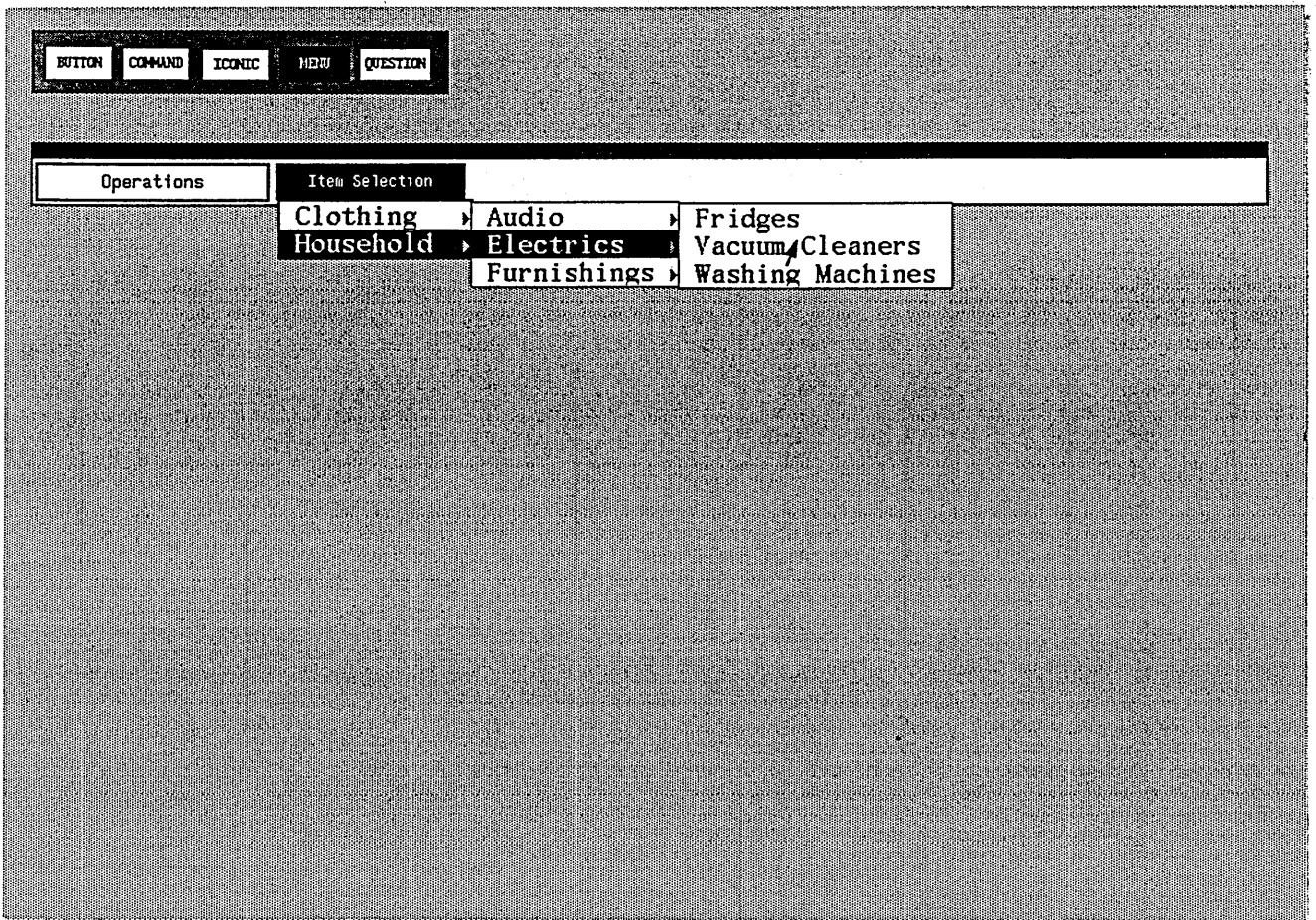
3. menu interface



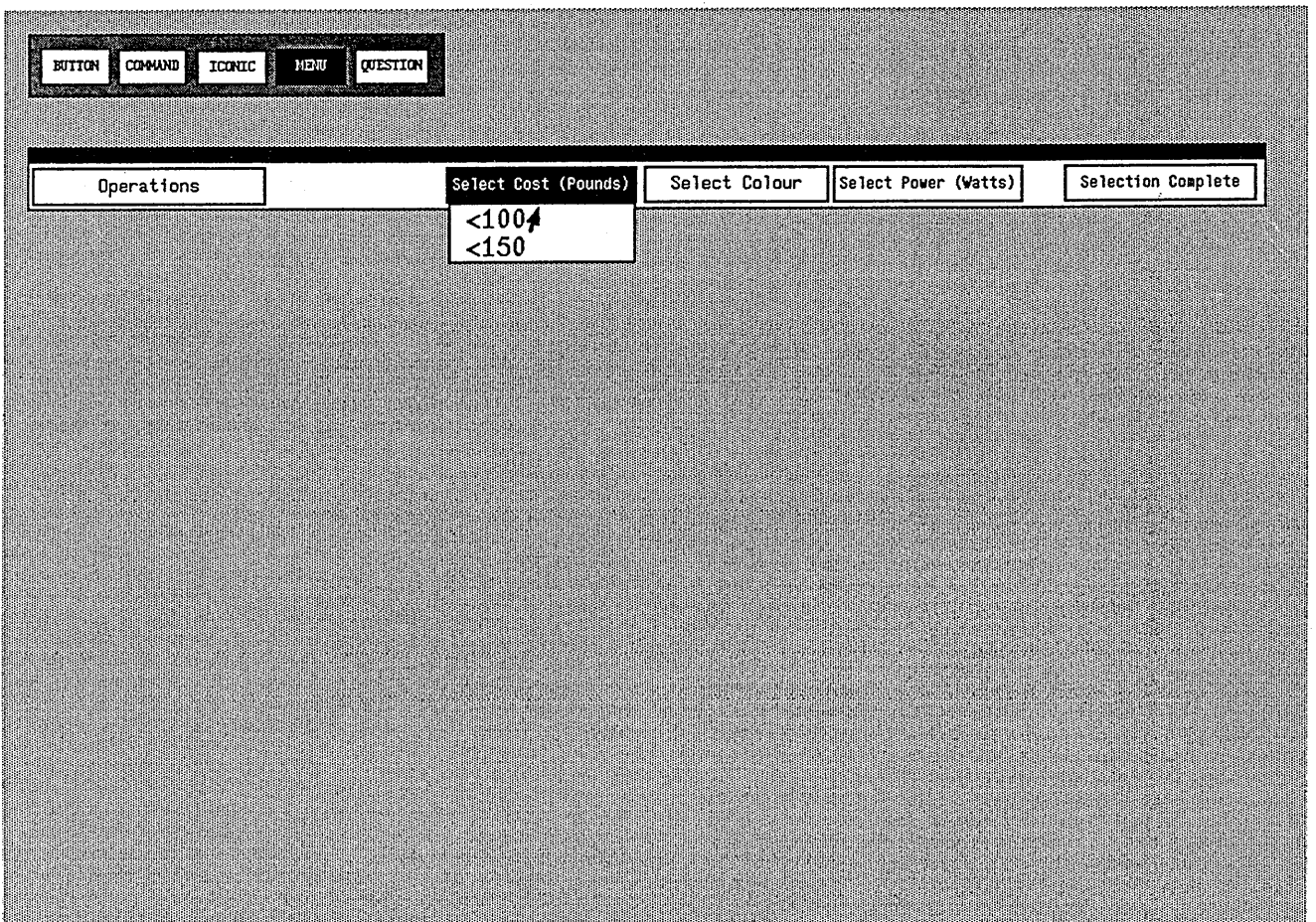
3.1



3.2



3.3



3.4

BUTTON COMMAND ICONIC MENU QUESTION

Operations Select Cost (Pounds) Select Colour Select Power (Watts) Selection Complete

WHITE
GREY
BLACK

3.5

BUTTON COMMAND ICONIC MENU QUESTION

Operations Select Cost (Pounds) Select Colour Select Power (Watts) Selection Complete

800
1000
1200

3.6

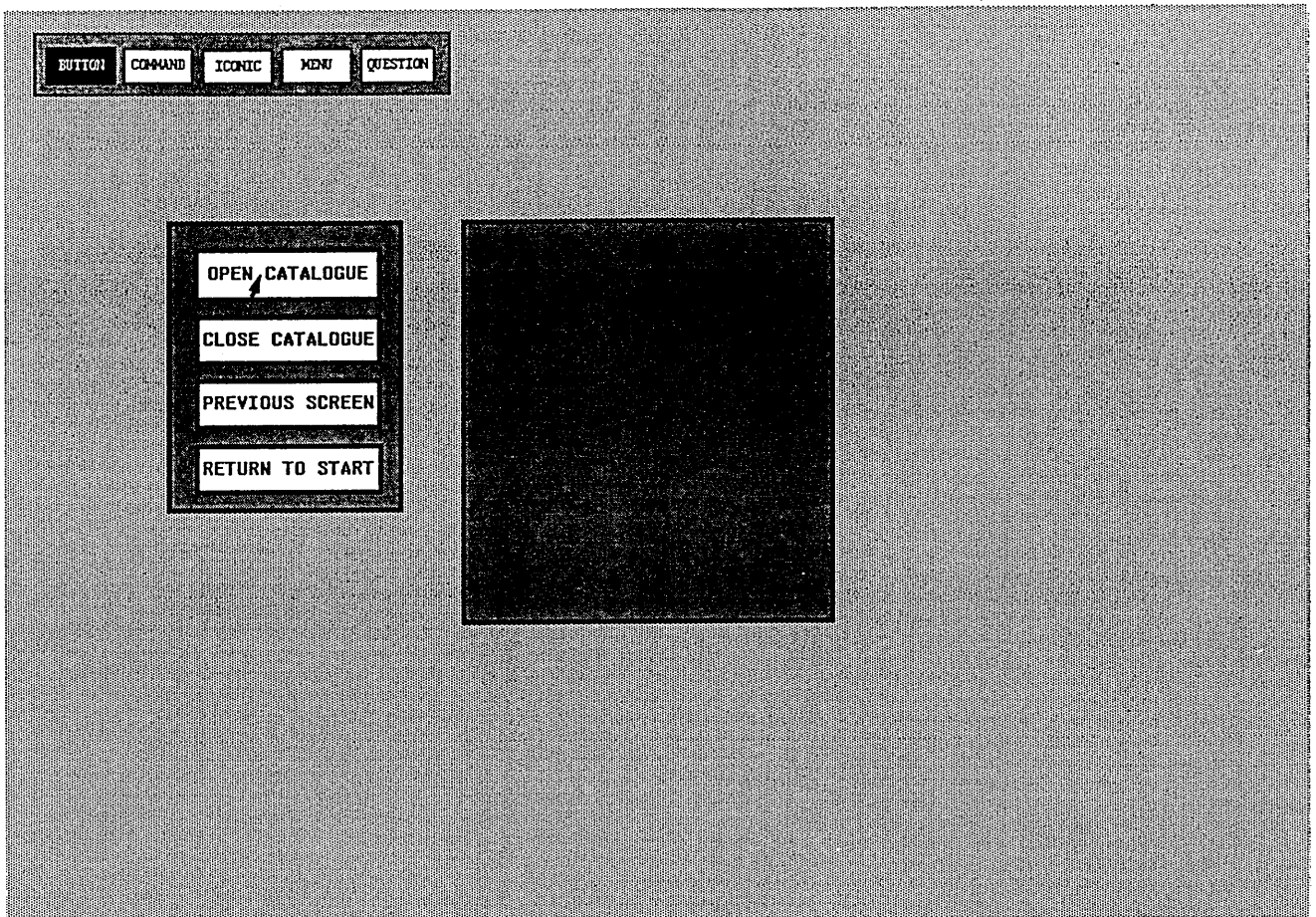
Operations

List of available vacuum cleaners

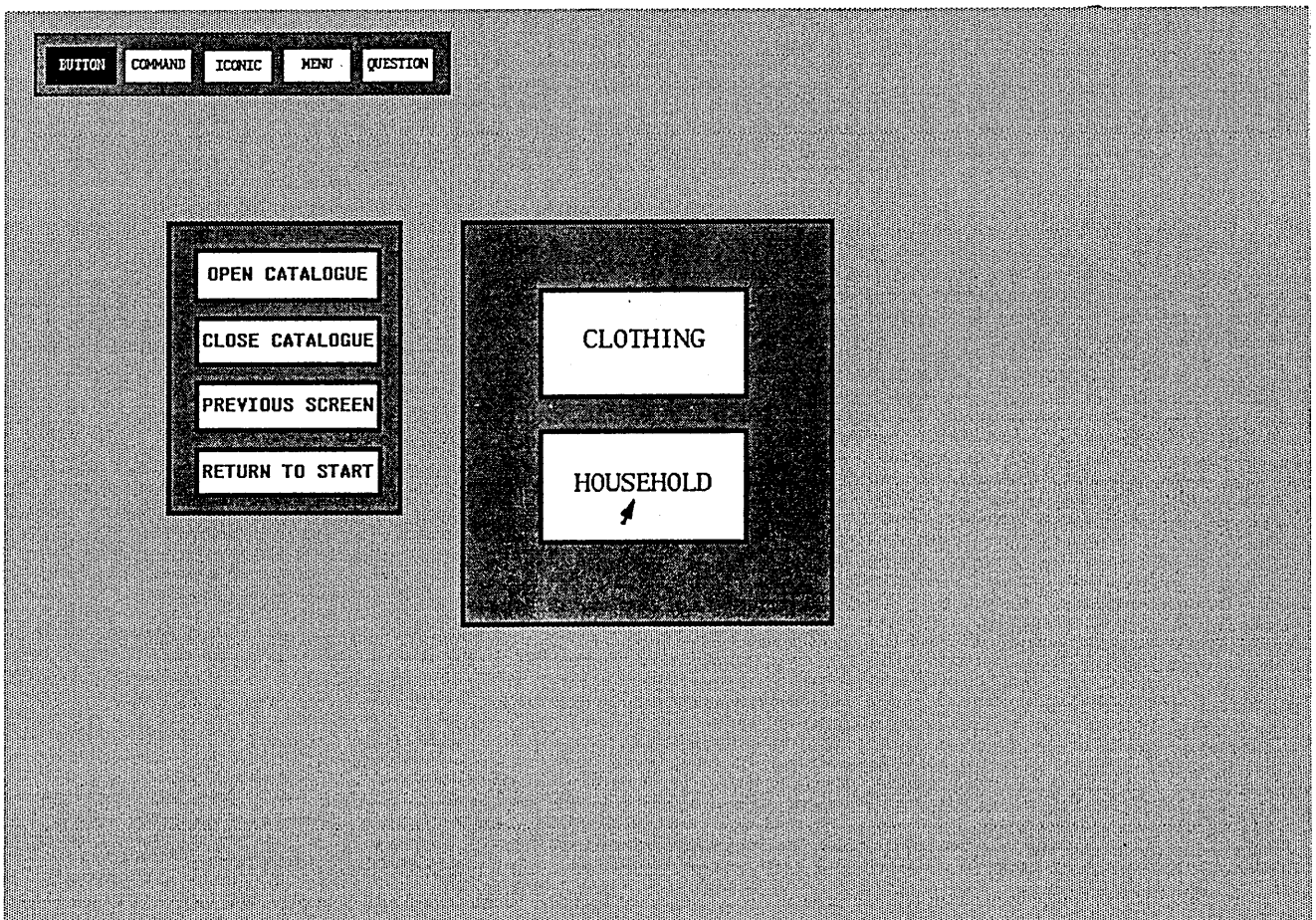
CAT. NO.	DESCRIPTION	COLOUR	POWER (WATTS)	COST (POUNDS)
PB0051	(ELECTRONIC CYLINDER)	GREY	1000	89.99

3.7

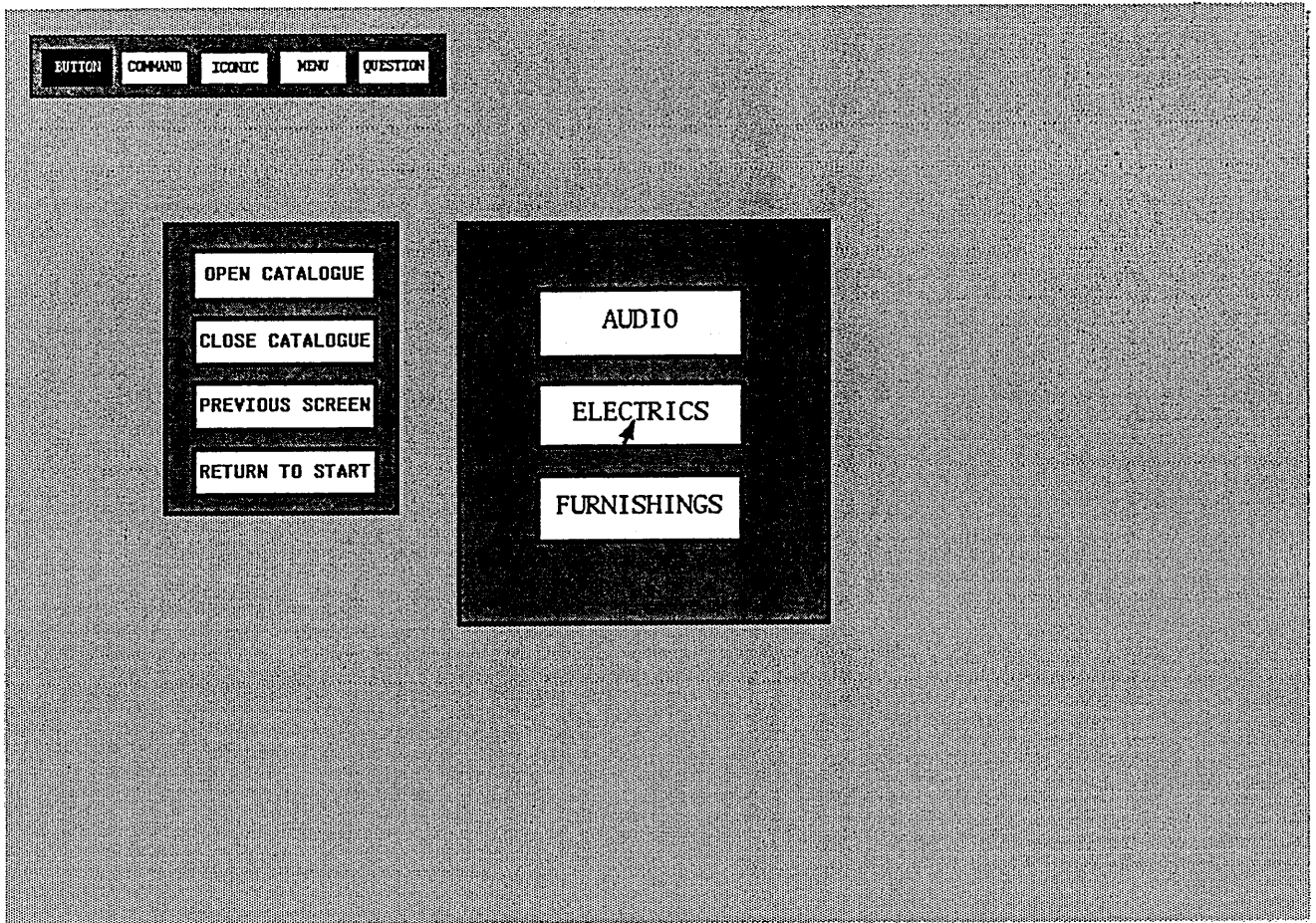
4. button interface



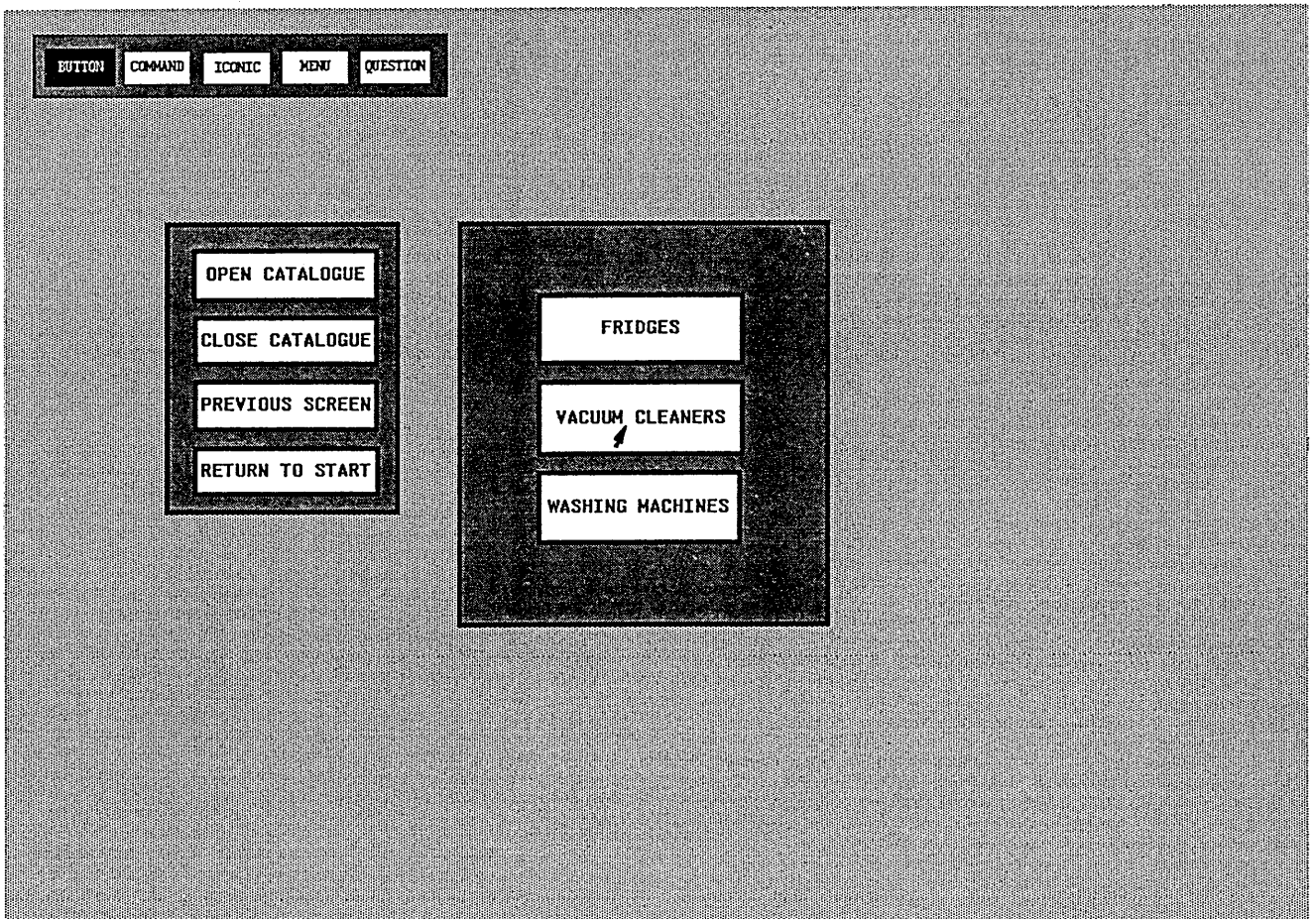
4.1



4.2



4.3



4.4

OPEN CATALOGUE
 CLOSE CATALOGUE
 PREVIOUS SCREEN
 RETURN TO START

Select Cost (Pounds) Select Power (Watts)
 <100 800
 1000
 <150 1200
 Select Colour
 WHITE
 GREY
 BLACK PRESS TO ENTER

4.5

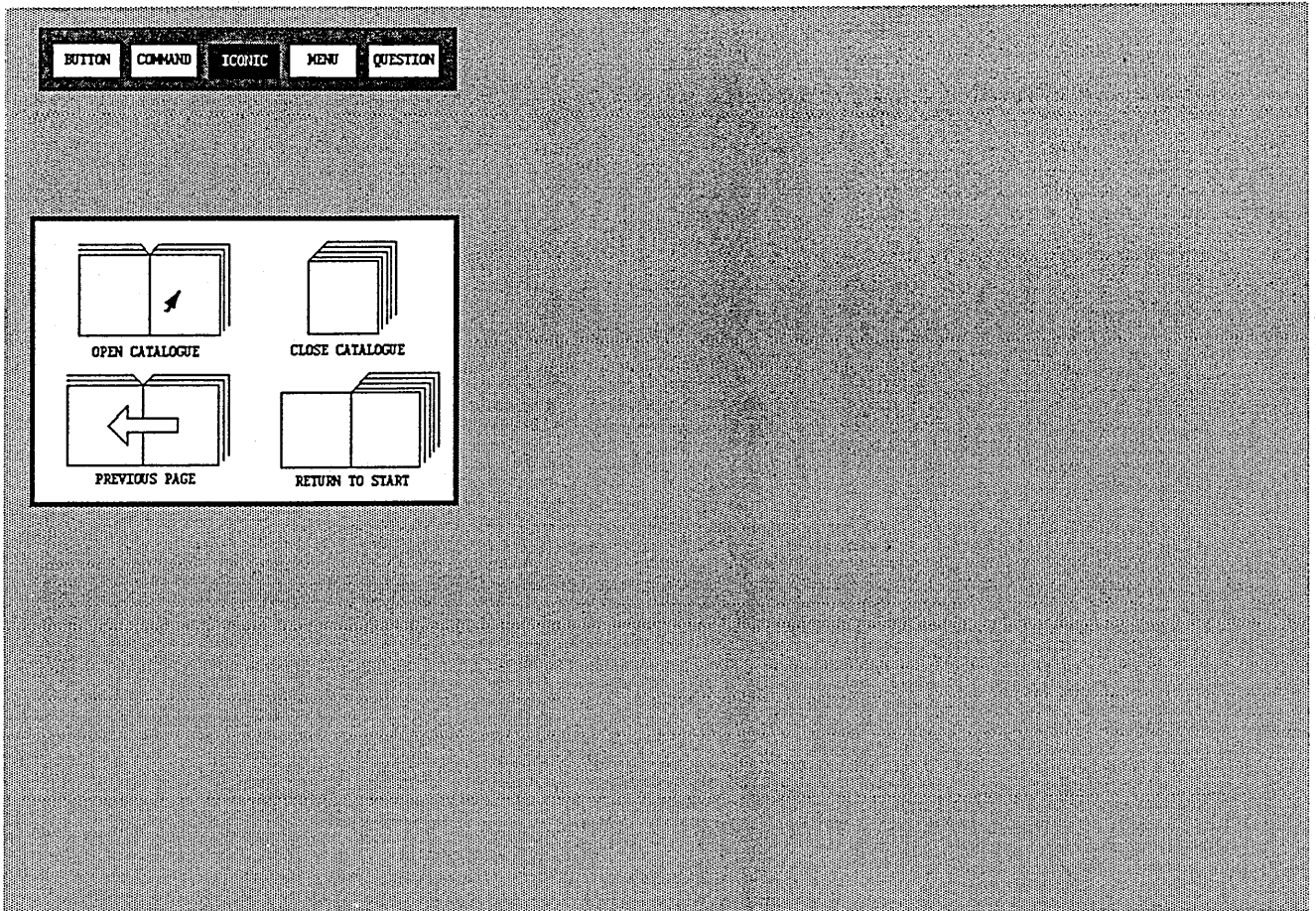
OPEN CATALOGUE
 CLOSE CATALOGUE
 PREVIOUS SCREEN
 RETURN TO START

List of available vacuum cleaners

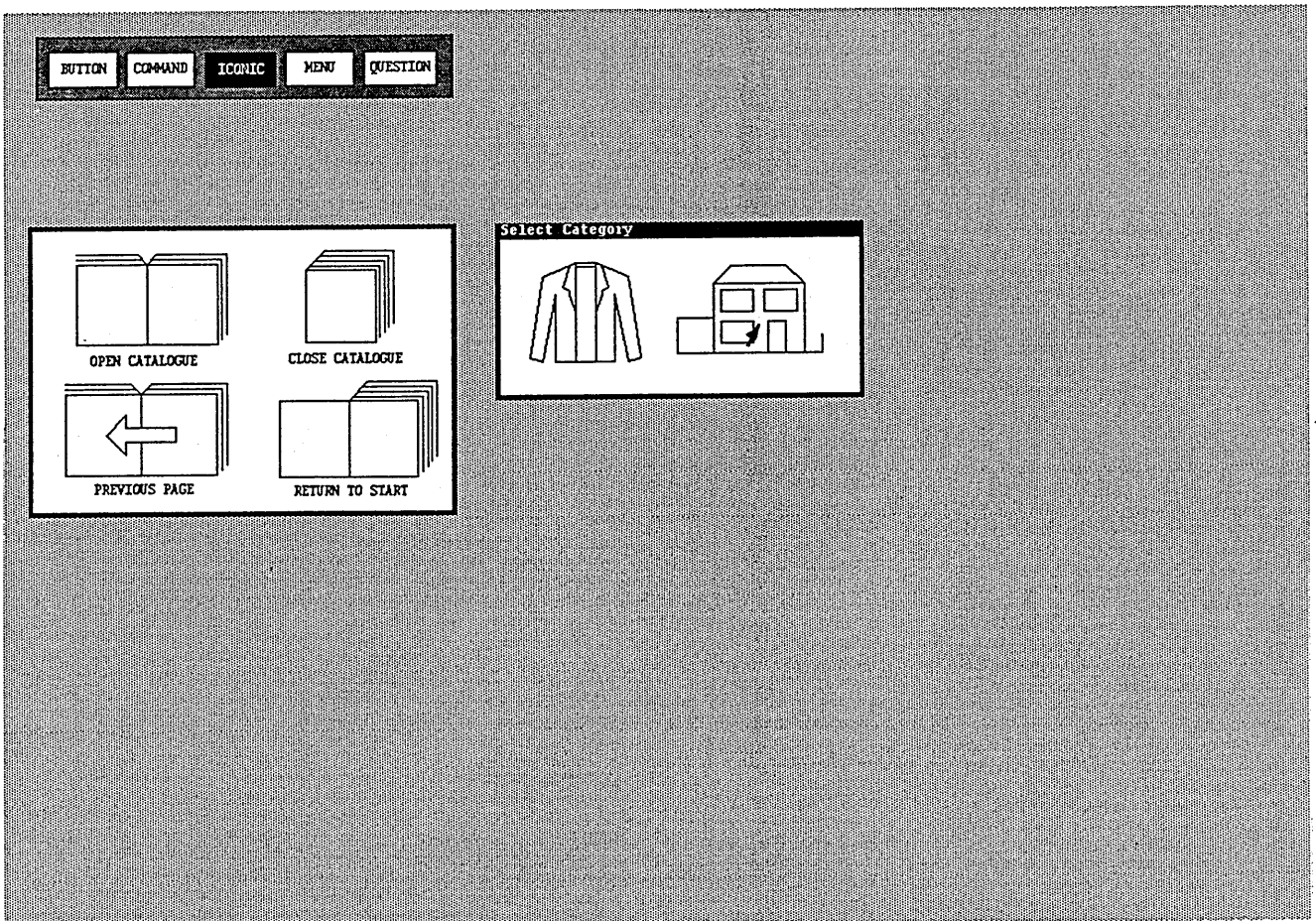
CAT. NO.	DESCRIPTION	COLOUR	POWER (WATTS)	COST (POUNDS)
PB8051	(ELECTRONIC CYLINDER)	GREY	1000	89.99

4.6

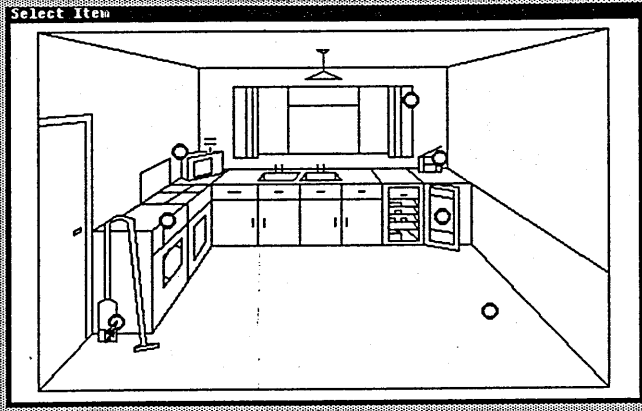
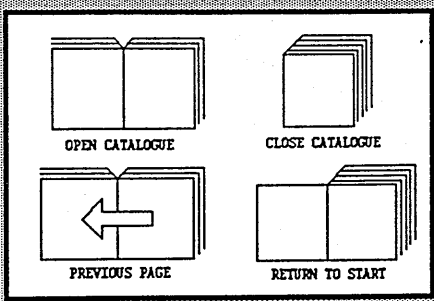
5. iconic interface



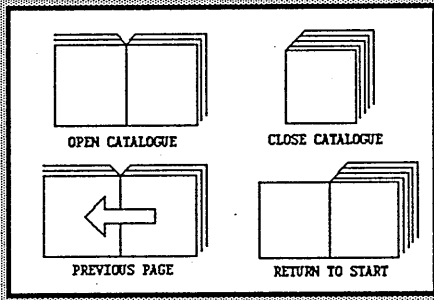
5.1



5.2



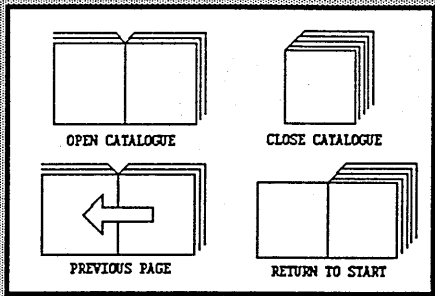
5.3



VACUUM CLEANERS

Select Cost (Pounds)	Select Power (Watts)
<100	800
<150	1000
	1200
Select Colour	
WHITE	
GREY	
BLACK	PRESS TO ENTER

5.4



List of available vacuum cleaners

CAT. NO.	DESCRIPTION	COLOUR	POWER (WATTS)	COST (POUNDS)
PB8851	(ELECTRONIC CYLINDER)	GREY	1888	89.99

Appendix 2

Experiment One: Computing Experience Questionnaire.

Please classify your level of experience with the following types of computer systems as none, some or a lot:

1. Command language systems. Ie. systems in which the dialogue with the computer involves typing in commands to a system prompt. (eg. E-mail on the VAX, MS-DOS).

none / some / a lot *

2. Mouse and menu systems. Ie. systems in which the dialogue with the computer involves selecting items from menus using a mouse. (eg. Microsoft Word, and most Apple Macintosh applications).

none / some / a lot *

3. Iconic systems. Ie. systems in which the dialogue with the computer involves using a mouse to click on pictures representing objects or operations. (eg. MacPaint and MacDraw on the Apple Macintosh).

none / some / a lot *

4. Question and answer systems. Ie. systems in which the dialogue with the computer involves typing in answers to specific questions or answers to prompts for specific information. (eg. many commercial data entry systems).

none / some / a lot *

5. Mouse and button systems. Ie. systems in which the dialogue with the computer involves clicking on named buttons. (eg. many Hypercard applications).

none / some / a lot *

* Delete as applicable.

Appendix 3

Experiment One: Example Test Session Tasks.

1. How many types of men's jeans are available, which cost less than £35, are blue in colour and have a waist size of 34 inches?

No. _____

2. Are there any vacuum cleaners available, which cost less than £100, are white in colour and have a power of 1000 watts?

Yes / No *

3. How many types of women's t-shirts are available, which cost less than £15, are navy in colour and size 14-16?

No. _____

4. Are there any carpets available, which cost less than £10 per yard, are brown in colour and 13 feet in width?

Yes / No *

5. Find the catalogue number of a shirt which costs less than £15, is grey in colour and collar size 15.5 inches.

Cat. No. _____

6. Find the catalogue number of a dress which costs less than £40, is red in colour and size 12.

Cat. No. _____

7. How many types of television are available, which cost less than £250, are black in colour and have a screen size of 14 inches?

No. _____

* Delete as applicable.

Appendix 4

Experiment One: Instruction Sheet.

Instructions

This experiment is being carried out to discover the relative merits of five different types of interface to a computer system.

The computer system allows you to obtain lists of items which are available from a shopping catalogue. You specify the type of item you are interested in, and three attributes the items must have. For example, you may specify that you are interested in vacuum cleaners, which cost less than £100, are grey in colour and have a power of 1000 watts. The system presents you with a list of items available from the catalogue which fit this description.

You will be presented with each of the five interfaces to the system in turn. For each interface you will be given a practice session in which you can try out the interface and then a test session in which you will be asked to carry out a series of tasks using the interface.

Do you have any questions?

Subject No.	Spatial Score (1-40)	Verbal Score (1-60)	Abstract Score (1-115)	STM Score (1-10)	Thinking/ Feeling (F68-T68)	Command Experience (1-3)
1	28	43	82	4	T19	2
2	17	*	70	3	T43	2
3	32	32	76	1	T13	2
4	30	39	90	6	T3	1
5	12	40	35	5	T11	2
6	14	34	63	5	T13	1
7	32	45	88	5	F11	1
8	34	35	101	7	F25	2
9	25	40	92	2	T17	1
10	21	41	69	4	T5	1
11	17	38	66	3	F35	1
12	7	38	60	5	F15	2
13	23	42	86	5	T29	2
14	28	35	84	4	F23	2
15	22	36	99	8	T13	2
16	36	44	79	5	T39	1
17	23	35	90	5	T5	1
18	12	28	54 ⁺	5	T11 ⁺	2
19	19	42	68	4	T1	1
20	26	30	73	6	F27	2
21	35	38	97	6	T51	1
22	10	14	74	2	T43	2
23	19	*	78	6	T23	1
24	34	39	52	5	T11	1

contd/

Subject No.	Question Experience (1-3)	Menu Experience (1-3)	Button Experience (1-3)	Iconic Experience (1-3)	Command Time (secs)	Command Ease (1-5)
1	2	2	2	2	341	2
2	2	2	2	2	440	5
3	2	1	1	1	257	1
4	2	2	2	2	325	3
5	2	1	2	2	576	2
6	2	1	2	2	232	3
7	1	1	1	1	284	1
8	3	2	3	3	253	2
9	1	3	3	3	314	1
10	1	1	1	1	392	3
11	3	2	2	2	270	2
12	3	3	3	3	384	2
13	3	2	3	3	250	3
14	3	2	2	2	353	2
15	3	2	3	2	388	2
16	2	3	3	3	214	5
17	2	1	2	1	395	4
18	2	1	2	1	432	5
19	2	2	2	2	380	3
20	2	1	1	1	309	5
21	2	1	3	2	212	3
22	2	2	2	2	285	4
23	2	1	2	2	352	5
24	3	2	2	3	229	3

contd/

Subject No.	Command Enjoyment (1-5)	Command Syntactic Errors	Command Command Use Errors	Command Uses of Help	Question Time (secs)	Question Ease (1-5)
1	2	1	0	2	318	1
2	5	1	1	4	457	3
3	1	1	2	1	371	2
4	4	0	0	2	441	2
5	2	1	3	8	422	5
6	3	1	0	1	287	2
7	2	0	0	2	378	5
8	2	0	0	7	299	1
9	3	1	0	6	316	4
10	4	4	0	5	335	2
11	3	2	0	4	379	4
12	3	2	0	2	449	3
13	4	0	0	7	344	1
14	4	2	0	6	492	4
15	1	2	0	9	297	1
16	4	0	0	2	252	2
17	5	0	0	5	446	2
18	4	2	2	4	475	3
19	4	2	0	6	381	3
20	5	0	0	6	388	4
21	3	0	0	0	377	1
22	4	1	1	1	459	3
23	5	1	0	7	490	4
24	2	4	2	2	251	4

contd/

Subject No.	Question Enjoyment (1-5)	Question Syntactic Errors	Question Command Use Errors	Menu Time (secs)	Menu Ease (1-5)	Menu Enjoyment (1-5)
1	4	0	0	310	2	4
2	5	0	0	435	2	3
3	2	0	0	525	3	5
4	3	0	0	455	3	3
5	5	2	0	356	4	4
6	3	0	0	332	1	1
7	5	0	0	350	4	4
8	1	0	0	317	1	4
9	5	0	0	352	3	3
10	3	0	0	279	2	3
11	4	1	1	420	3	3
12	3	2	0	0	4	4
13	1	0	0	294	4	5
14	4	0	0	359	3	3
15	3	0	0	311	1	2
16	5	0	0	334	3	3
17	4	0	0	343	2	4
18	3	0	2	491	2	3
19	4	1	0	313	1	2
20	4	0	0	313	2	3
21	2	0	0	298	3	3
22	4	0	0	285	2	3
23	4	0	0	342	3	3
24	4	1	0	314	3	4

contd/

Subject No.	Menu Syntactic Errors	Menu Command Use Errors	Button Time (secs)	Button Ease (1-5)	Button Enjoyment (1-5)	Button Syntactic Errors
1	0	0	226	2	2	0
2	0	1	467	1	1	0
3	0	0	220	3	3	0
4	0	0	315	1	2	0
5	0	0	289	1	1	0
6	0	1	199	1	1	0
7	0	0	253	1	2	0
8	0	0	259	1	3	0
9	0	0	237	1	1	0
10	0	0	265	1	2	0
11	0	0	253	2	2	0
12	0	0	334	2	2	0
13	0	0	205	1	2	0
14	0	0	216	1	1	0
15	0	0	193	1	2	0
16	0	0	171	1	1	0
17	0	0	232	2	3	0
18	0	1	247	1	2	0
19	0	0	221	1	1	0
20	0	0	246	2	2	0
21	0	0	239	1	1	0
22	0	0	250	1	2	0
23	0	0	350	1	1	0
24	0	0	202	3	3	0

contd/

Subject No.	Button Command Use Errors	Iconic Time (secs)	Iconic Ease (1-5)	Iconic Enjoyment (1-5)	Iconic Syntactic Errors	Iconic Command Use Errors
1	0	193	1	1	0	0
2	3	249	1	2	1	0
3	0	269	3	4	0	0
4	0	346	2	2	0	0
5	0	197	4	3	0	0
6	0	208	1	2	0	0
7	0	248	4	2	1	0
8	1	206	3	2	0	0
9	0	319	3	2	2	0
10	1	221	1	2	2	1
11	0	278	3	3	0	0
12	1	0	4	4	0	1
13	0	208	4	4	0	0
14	0	283	2	2	0	0
15	0	185	1	1	0	0
16	0	170	4	2	0	0
17	0	238	1	1	0	0
18	0	249	2	1	0	0
19	0	262	1	1	0	0
20	0	239	1	1	0	0
21	0	202	2	2	0	0
22	0	190	2	2	0	0
23	0	268	2	2	2	0
24	0	178	3	3	0	1

Notes:

For all experience, ease and enjoyment ratings, 1 denotes high.

Time and error values are for the complete test session with each interface.

Missing data:

@ Timing device failed.

* Subject gave up part way through test, finding it difficult.

Subject included in low ability group.

+ Subject completed test late. Value missing from some analyses.

Appendix 6

Experiment Two: Test System.

1. command interface

2. menu interface

1. command interface

SYS> █

students				
name	id	reg	course	grade
ACKEROYD	S1	1989	COMPUTING	66
ELLIS	S2	1990	MATHS	72
...				

staff			
name	id	course	position
FRENCH	T1	MATHS	SENIOR_LLECTURER
HEATHCOTE	T2	MATHS	LECTURER
...			

1.1

SYS> udb

UDB> select name from staff

List of staff:

FRENCH
HEATHCOTE
JONES
LAI
REDHEAD

UDB> select name from staff where course=maths
Invalid input.

UDB> help █

students				
name	id	reg	course	grade
ACKEROYD	S1	1989	COMPUTING	66
ELLIS	S2	1990	MATHS	72
...				

staff			
name	id	course	position
FRENCH	T1	MATHS	SENIOR_LLECTURER
HEATHCOTE	T2	MATHS	LECTURER
...			

1.2

HELP

[Type 'quit' to exit help system]

Enter code for help required:

1. Opening university database.
2. Querying university database.
3. Exiting university database.

HLP> 2

students

name	id	reg	course	grade
ACKEROYD	S1	1989	COMPUTING	66
ELLIS	S2	1990	MATHS	72
				...

staff

name	id	course	position
FRENCH	T1	MATHS	SENIOR_LLECTURER
HEATHCOTE	T2	MATHS	LECTURER
			...

1.3

Querying the database.

To query the university database, a statement of the following form is typed in at the UDB> prompt:

```
select <attribute> from <table> where <condition1> and <condition2>
```

The 'where' and 'and' sections of the statement are optional.

<table> specifies the table which the required information is to be obtained from. The tables are given beside this screen. <table> can be either 'students' or 'staff'.

<attribute> specifies the type of information required about the students or staff. This can be either 'name' or 'id'.

<condition1> and <condition2> specify any conditions the students or staff must fulfil to be selected. The condition takes the form <attribute> =, < or > <value>. <attribute> can be 'reg', 'course', 'grade' or 'position'; and <value> is the required value for the attribute.

Examples of statements:

To list the names of all students:
select name from students

To list the id numbers of all the students taking the maths course:
select id from students where course = maths

To list the id numbers of all the students taking maths who have a grade of over 60%:
select id from students where course = maths and grade > 60

Control-R can be used to retrieve the previous statement entered.
Control-B can be used to move the cursor backwards to edit a statement line.
Control-F can be used to move the cursor forwards to edit a statement line.

Hit any key to continue. █

students

name	id	reg	course	grade
ACKEROYD	S1	1989	COMPUTING	66
ELLIS	S2	1990	MATHS	72
				...

staff

name	id	course	position
FRENCH	T1	MATHS	SENIOR_LLECTURER
HEATHCOTE	T2	MATHS	LECTURER
			...

1.4

HELP

[Type 'quit' to exit help system]

Enter code for help required:

1. Opening university database.
2. Querying university database.
3. Exiting university database.

HLP> quit

students

name	id	reg	course	grade
ACKEROYD	S1	1989	COMPUTING	66
ELLIS	S2	1990	MATHS	72
				...

staff

name	id	course	position
FRENCH	T1	MATHS	SENIOR_LLECTURER
HEATHCOTE	T2	MATHS	LECTURER
			...

1.5

SYS> udb

UDB> select name from staff

List of staff:

FRENCH
HEATHCOTE
JONES
LAI
REDHEAD

UDB> select name from staff where course=maths

Invalid input.

UDB> help

UDB> select name from staff where course = maths

List of staff:

FRENCH
HEATHCOTE
JONES

UDB> █

students

name	id	reg	course	grade
ACKEROYD	S1	1989	COMPUTING	66
ELLIS	S2	1990	MATHS	72
				...

staff

name	id	course	position
FRENCH	T1	MATHS	SENIOR_LLECTURER
HEATHCOTE	T2	MATHS	LECTURER
			...

1.6

```

SYS> udb
UDB> select name from staff

List of staff:

FRENCH
HEATHCOTE
JONES
LAI
REDHEAD

UDB> select name from staff where course=maths
Invalid input.
UDB> help
UDB> select name from staff where course = maths

```

List of staff:

```

FRENCH
HEATHCOTE
JONES

```

```

UDB> select name from staff where course = maths and position = lecturer

```

List of staff:

```

HEATHCOTE
JONES

```

```

UDB> █

```

students				
name	id	reg	course	grade
ACKEROYD	S1	1989	COMPUTING	66
ELLIS	S2	1990	MATHS	72
...				

staff			
name	id	course	position
FRENCH	T1	MATHS	SENIOR_LLECTURER
HEATHCOTE	T2	MATHS	LECTURER
...			

1.7

```

SYS> udb
UDB> select name from staff

List of staff:

FRENCH
HEATHCOTE
JONES
LAI
REDHEAD

UDB> select name from staff where course=maths
Invalid input.
UDB> help
UDB> select name from staff where course = maths

```

List of staff:

```

FRENCH
HEATHCOTE
JONES

```

```

UDB> select name from staff where course = maths and position = lecturer

```

List of staff:

```

HEATHCOTE
JONES

```

```

UDB> quit

```

```

SYS> █

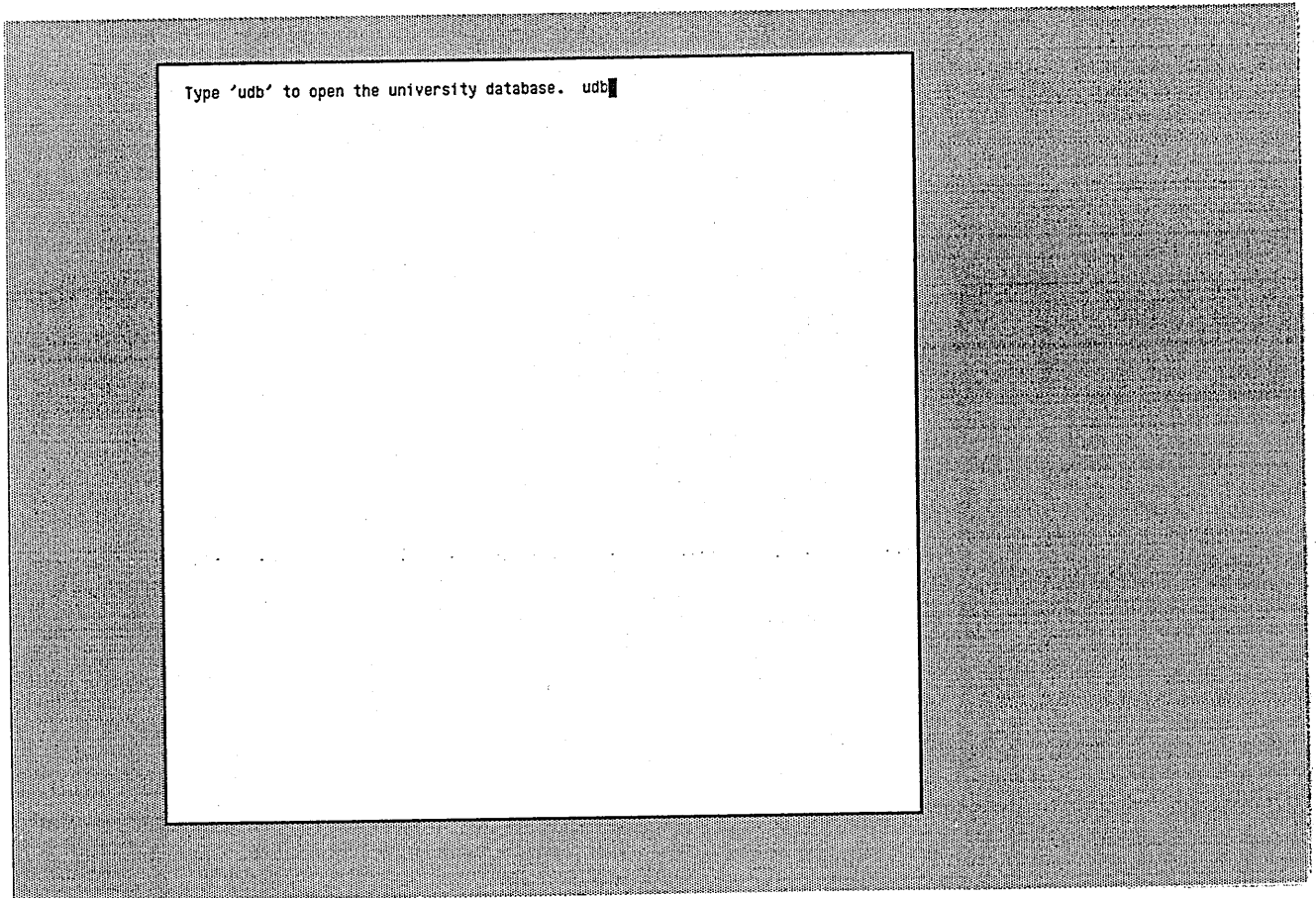
```

students				
name	id	reg	course	grade
ACKEROYD	S1	1989	COMPUTING	66
ELLIS	S2	1990	MATHS	72
...				

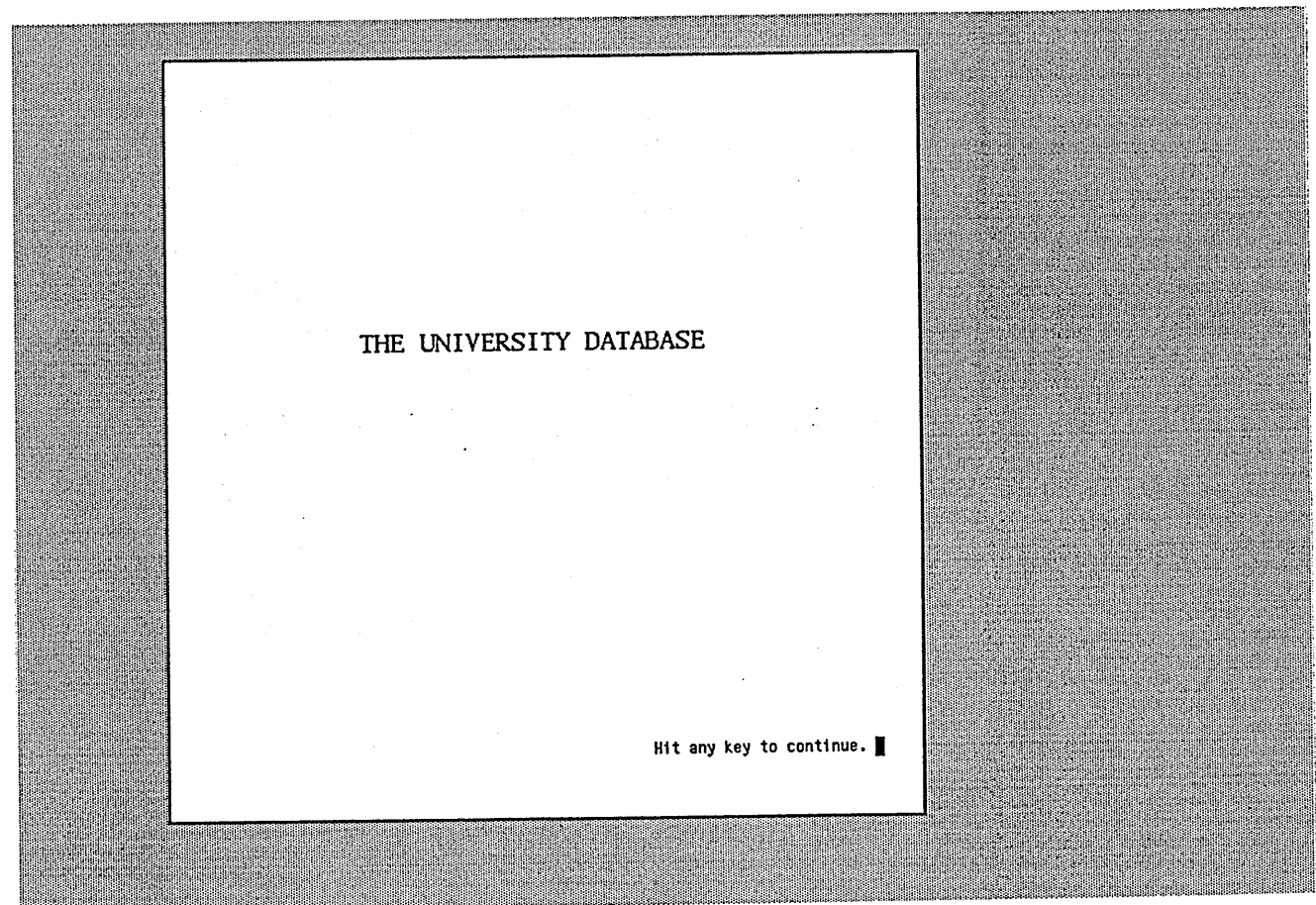
staff			
name	id	course	position
FRENCH	T1	MATHS	SENIOR_LLECTURER
HEATHCOTE	T2	MATHS	LECTURER
...			

1.8

2. menu interface



2.1



2.2


```
University database                [Type 'quit' to exit the database]

The university database allows you to obtain information about:

  1. students
  2. staff

Enter code for category you are interested in (1e. '1' or '2'): 1
```

2.3

```
Students                          [Type 'quit' to exit database]
                                   [Type 'start' to return to start]
                                   [Type 'back' to return to previous question]

Do you want a list of:

  1. names
  2. identification numbers

Enter code for category (1e. '1' or '2'): 1
```

2.4

Students

[Type 'quit' to exit database]
[Type 'start' to return to start]
[Type 'back' to return to previous question]

Do you want a list of:

1. names
2. identification numbers

Enter code for category (ie. '1' or '2'): 1

What conditions must the students fulfil to be selected for the list:

1. none
2. particular registration date
3. particular course
4. particular grade

Enter code for category/categories, separated by a space (eg. '4', '2 3'): 2 4

2.5

Students

[Type 'quit' to exit database]
[Type 'start' to return to start]
[Type 'back' to return to previous question]

Do you want a list of:

1. names
2. identification numbers

Enter code for category (ie. '1' or '2'): 1

What conditions must the students fulfil to be selected for the list:

1. none
2. particular registration date
3. particular course
4. particular grade

Enter code for category/categories, separated by a space (eg. '4', '2 3'): 2 4

Enter registration date condition (eg. '=1989', '<1990', '>1988'): =1990

Enter grade condition (eg. '<68', '=78', '>78'): >68

Press return to see list. █

2.6

List of students:

[Type 'quit' to exit catalogue]
[Type 'start' to return to start]

ELLIS
THOMPSON

Use 'start' and 'quit' commands as appropriate. start: █

2.7

University database

[Type 'quit' to exit the database]

The university database allows you to obtain information about:

1. students
2. staff

Enter code for category you are interested in (ie. '1' or '2'): █

2.8

Appendix 7

Experiment Two: Computing Experience Questionnaire.

Please rate your level of experience with using interfaces akin to the two interfaces in this experiment:

	a lot				none
command	1	2	3	4	5
menu	1	2	3	4	5

Appendix 8

Experiment Two: Example Test Session Tasks.

1. List the identification numbers of all the students.
2. List the identification numbers of all the students on the maths course.
3. List the identification numbers of all the students on the computing course.
4. List the identification numbers of all the students on the computing course who registered in 1990.
5. List the names of all the members of staff.
6. List the names of all the members of staff who teach on the computing course.
7. List the names of all the members of staff who are lecturers on the computing course.
8. List the names of all the members of staff who are senior lecturers on the computing course.
9. List the names of all the members of staff who are senior lecturers on the maths course.
10. List the names of all the students.

11. List the names of all the students who registered after 1988 on the maths course.

12. List the names of all the students who registered after 1988.

Appendix 9

Experiment Two: Instruction Sheet.

Instructions

This experiment is being carried out to discover the merits of two different interfaces to a database system.

The database system allows you to access information about students and staff working at a university.

You will be presented with each of the interfaces in turn. For each interface you will be given a practice session in which you can try out the interface and then a test session in which you will be asked to carry out a series of tasks using the interface.

Do you have any questions?

Appendix 10

Experiment Two: Raw Results.

Subject No.	Spatial Score (1-40)	Command Experience (1-5)	Menu Experience (1-5)	Command Time (secs)	Command Ease (1-5)	Command Enjoyment (1-5)
1	35	1	2	265	2	1
2	36	4	4	228	2	2
3	20	1	4	269	2	2
4	23	3	3	411	3	4
5	22	4	3	421	2	1
6	34	4	2	274	4	3
7	32	2	3	243	2	2
8	25	1	2	275	2	2
9	19	2	3	252	1	2
10	34	3	3	248	1	2
11	18	5	5	497	2	4
12	28	2	2	290	2	3
13	17	4	3	509	2	2
14	26	4	1	338	3	3
15	15	4	3	490	3	2
16	30	2	4	304	3	2
17	7	3	3	300	3	2
18	34	1	4	181	2	2
19	31	4	4	308	2	3
20	16	3	1	333	3	4
21	23	3	2	452	3	2
22	15	2	5	312	1	1
23	29	3	4	313	2	2
24	28	4	4	380	3	3
25	13	4	3	560	3	2
26	18	5	5	670	3	2
27	27	4	3	337	1	2
28	7	2	4	356	2	2
29	10	2	2	380	1	2
30	14	2	4	258	2	2

contd/

Subject No.	Command Errors	Menu Time (secs)	Menu Ease (1-5)	Menu Enjoyment (1-5)	Menu Errors
1	0	347	2	2	0
2	0	291	3	3	0
3	0	354	3	3	0
4	0	585	5	5	0
5	2	309	1	1	0
6	1	424	3	4	0
7	0	306	2	3	0
8	0	429	4	3	0
9	0	448	3	4	0
10	0	338	2	4	0
11	3	333	1	3	0
12	1	361	1	4	0
13	2	459	3	4	1
14	1	474	2	2	0
15	2	432	4	3	0
16	0	425	2	2	0
17	1	347	2	4	0
18	0	304	5	5	0
19	0	396	3	4	0
20	0	399	1	2	0
21	1	487	2	3	1
22	1	404	1	1	0
23	0	341	3	3	0
24	1	435	4	3	0
25	1	473	2	3	0
26	6	531	1	3	0
27	2	427	1	3	1
28	1	511	4	3	1
29	0	410	1	4	0
30	1	334	3	4	0

Notes:

For experience, ease and enjoyment ratings, 1 denotes high.

Time and error values are for the complete test session with each interface.

Appendix 11

Test Session Times:

2(spatial)x2(interface) ANOVA for all subjects.

	df	SS	MS	F	P
Between	29				
S (spatial)	1	44946	44946	3.65	0.066
Ss within S	28	344573	12306		
Within	30				
I (interface)	1	41139	41139	12.08	0.002
IxS	1	27229	27229	8.00	0.009
IxSs within S	28	95346	3405		

2(experience)x2(interface) ANOVA for low spatial ability subject group.

	df	SS	MS	F	P
Between	11				
E (experience)	1	81201	81201	8.72	0.014
Ss within E	10	93151	9315		
Within	12				
I (interface)	1	24	24	0.02	0.898
IxE	1	64481	64481	46.07	0.000
IxSs within E	10	13995	1400		

2(experience)x2(interface) ANOVA for high spatial ability subject group.

	df	SS	MS	F	P
Between	11				
E (experience)	1	14114	14114	2.72	0.130
Ss within E	10	51883	5188		
Within	12				
I (interface)	1	59601	59601	86.58	0.000
IxE	1	43	43	0.06	0.808
IxSs within E	10	6884	688		

Ease Ratings:

2(experience)x2(interface) ANOVA for low spatial ability subject group.

	df	SS	MS	F	P
Between	11				
E (experience)	1	0.37	0.37	0.29	0.599
Ss within E	10	12.75	1.28		
Within	12				
I (interface)	1	0.38	0.38	0.65	0.438
IxI	1	3.38	3.38	5.87	0.036
IxSs within E	10	5.75	0.58		

2(experience)x2(interface) ANOVA for high spatial ability subject group.

	df	SS	MS	F	P
Between	11				
E (experience)	1	0.17	0.17	0.12	0.736
Ss within E	10	13.83	1.38		
Within	12				
I (interface)	1	0.67	0.67	0.73	0.414
IxI	1	0.17	0.17	0.18	0.679
IxSs within E	10	9.17	0.92		

Enjoyment Ratings:

2(experience)x2(interface) ANOVA for low spatial ability subject group.

	df	SS	MS	F	P
Between	11				
E (experience)	1	0.00	0.00	0.00	1.000
Ss within E	10	13.00	1.30		
Within	12				
I (interface)	1	6.00	6.00	13.85	0.004
IxI	1	0.67	0.67	1.54	0.243
IxSs within E	10	4.33	0.43		

2(experience)x2(interface) ANOVA for high spatial ability subject group.

	df	SS	MS	F	P
Between	11				
E (experience)	1	0.67	0.67	0.75	0.405
Ss within E	10	8.83	0.88		
Within	12				
I (interface)	1	4.17	4.17	10.00	0.010
IxE	1	0.67	0.67	1.60	0.235
IxSs within E	10	4.17	0.42		

Errors:

2(experience)x2(interface) ANOVA for low spatial ability subject group.

	df	SS	MS	F	P
Between	11				
E (experience)	1	7.04	7.04	7.75	0.190
Ss within E	10	9.08	0.91		
Within	12				
I (interface)	1	12.04	12.04	12.79	0.005
IxE	1	7.04	7.04	7.48	0.021
IxSs within E	10	9.42	0.94		

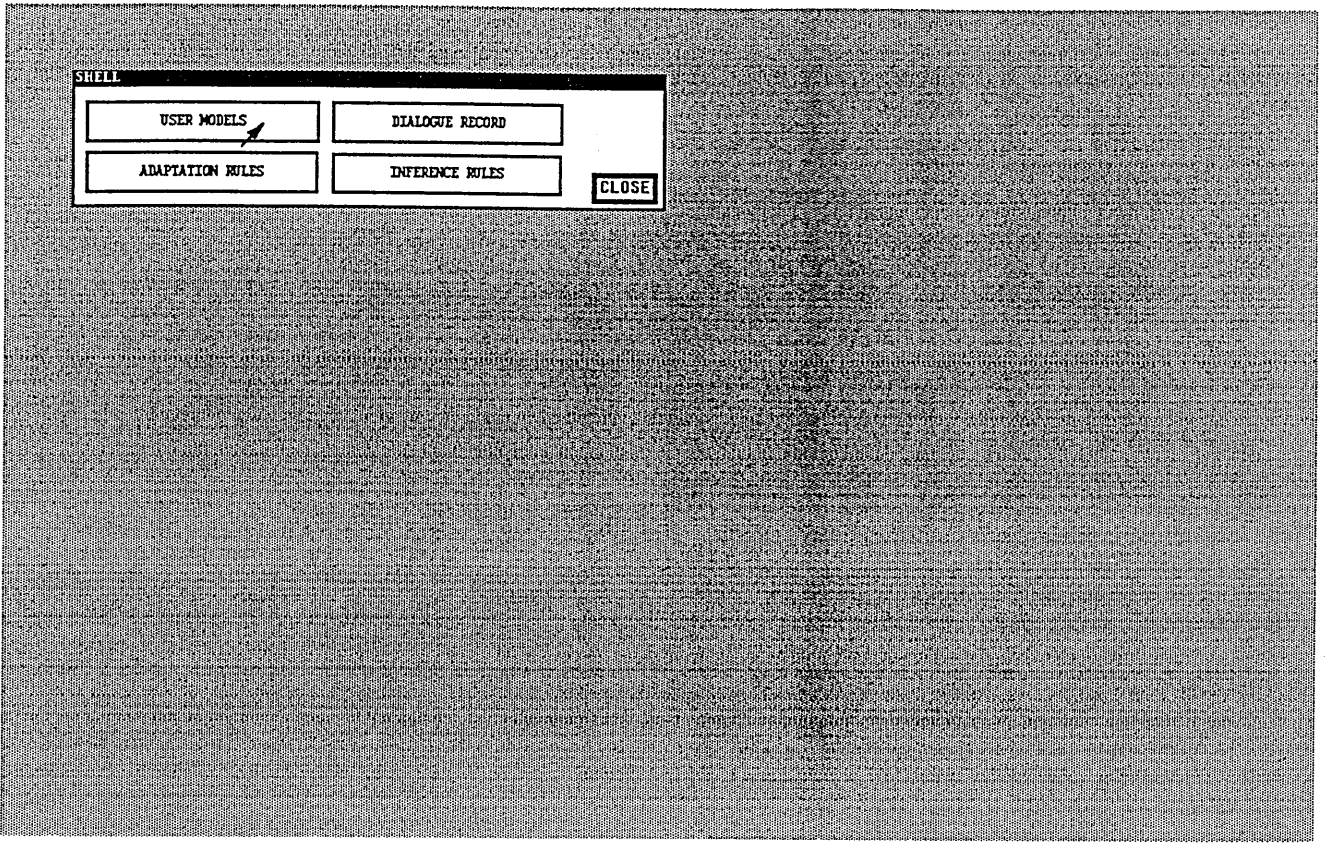
2(experience)x2(interface) ANOVA for high spatial ability subject group.

	df	SS	MS	F	P
Between	11				
E (experience)	1	1.04	1.04	3.05	0.111
Ss within E	10	3.42	0.34		
Within	12				
I (interface)	1	1.04	1.04	9.62	0.011
IxE	1	0.37	0.37	3.46	0.092
IxSs within E	10	1.08	0.11		

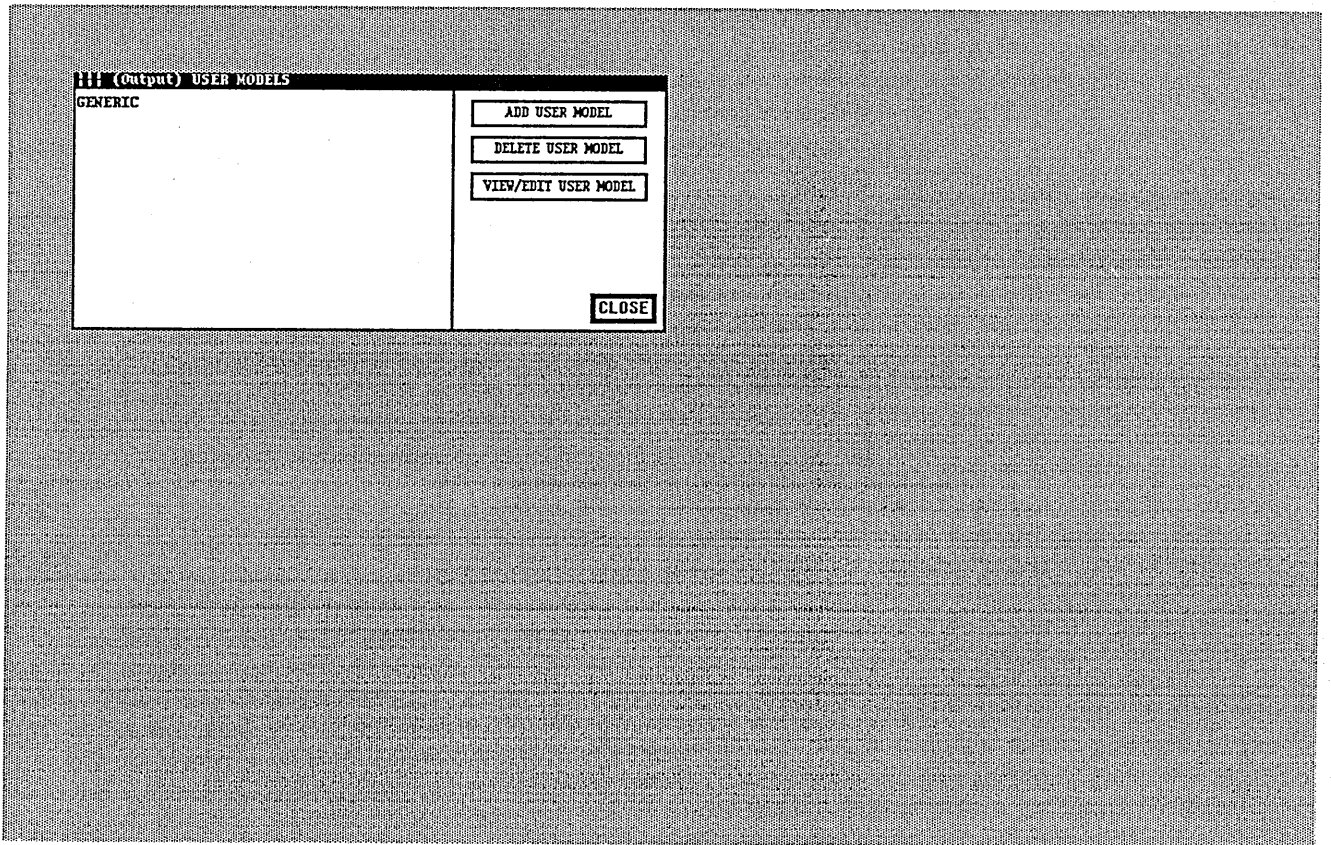
Appendix 12

Adaptive System Shell: Figures 7-11.

7. User Model
8. Adaptation Rules
9. Dialogue Record
10. Inference Rules
11. Management System



7.1



7.2

Figure 7. User Model.

USER MODELS	
<p>GENERIC</p>	<p>ADD USER MODEL</p> <p>DELETE USER MODEL</p> <p>VIEW/EDIT USER MODEL</p> <p>CLOSE</p>

Enter name of user model you wish to view/edit.

generic

7.3

GENERIC USER MODEL	
<p>Personal Profile</p> <div style="border: 1px solid black; height: 100px; width: 100%;"></div> <p>ADD CHARACTERISTIC</p> <p>DELETE CHARACTERISTIC</p> <p>EDIT CHARACTERISTIC</p>	<p>Cognitive Model</p> <div style="border: 1px solid black; height: 100px; width: 100%;"></div> <p>ADD CHARACTERISTIC</p> <p>DELETE CHARACTERISTIC</p> <p>EDIT CHARACTERISTIC</p> <p>CLOSE</p>

7.4

GENERIC USER MODEL

Personal Profile	Cognitive Model
<p>COMMAND.EXP</p>	
<p>ADD CHARACTERISTIC</p> <p>DELETE CHARACTERISTIC</p> <p>EDIT CHARACTERISTIC</p>	<p>ADD CHARACTERISTIC</p> <p>DELETE CHARACTERISTIC</p> <p>EDIT CHARACTERISTIC</p>
	CLOSE

Enter name of characteristic (name must be an atom).
 command.exp

Enter alternative values for characteristic.
 Entry must be of the following form:
 (<value1> <value2> ... <valuen>)
 Each value must be written as an atom.
 (expert high medium low none)

7.5

GENERIC USER MODEL

Personal Profile	Cognitive Model
<p>COMMAND.EXP</p> <p>Value: EXPERT HIGH MEDIUM LOW NONE</p>	
<p>ADD CHARACTERISTIC</p> <p>DELETE CHARACTERISTIC</p> <p>EDIT CHARACTERISTIC</p>	<p>ADD CHARACTERISTIC</p> <p>DELETE CHARACTERISTIC</p> <p>EDIT CHARACTERISTIC</p>
	CLOSE

7.6

GENERIC USER MODEL

Personal Profile	Cognitive Model
<p>COMMAND.EXP Value:</p> <p>EXPERT HIGH MEDIUM LOW NONE</p> <p>COMPUTER.USE Value:</p> <p>OCCASIONAL FREQUENT</p>	<p>SPATIAL Value:</p> <p>LOW HIGH</p>
<p>ADD CHARACTERISTIC</p> <p>DELETE CHARACTERISTIC</p> <p>EDIT CHARACTERISTIC</p>	<p>ADD CHARACTERISTIC</p> <p>DELETE CHARACTERISTIC</p> <p>EDIT CHARACTERISTIC</p> <p>CLOSE</p>

7.7

+++ (Output) USER MODELS

<p>GENERIC</p> <p>ADD USER MODEL</p> <p>DELETE USER MODEL</p> <p>VIEW/EDIT USER MODEL</p> <p>CLOSE</p>	<p>Enter name for new user model (name must be an atom).</p> <p>default</p> <p>Is this a default user model? (y/n)</p> <p>y</p>
--	---

7.8

USER MODELS

GENERIC

|

DEFAULT

/ \ / \ / \ / \

USER1 USER2 USER3 USER4 USERS

ADD USER MODEL

DELETE USER MODEL

VIEW/EDIT USER MODEL

CLOSE

Enter name of user model you wish to view/edit.

default

7.11

DEFAULT USER MODEL

Personal Profile

COMMAND.EXP Value:

EXPERT

HIGH

MEDIUM

LOW

NONE

COMPUTER.USE Value:

OCCASIONAL

FREQUENT

ENTER VALUES

Cognitive Model

SPATIAL Value:

LOW

HIGH

ENTER VALUES

CLOSE

7.12

!!! (Output) USER MODELS

```

  GRAPH TD
    GENERIC --> DEFAULT
    DEFAULT --> USER1
    DEFAULT --> USER2
    DEFAULT --> USER3
    DEFAULT --> USER4
    DEFAULT --> USER5
  
```

ADD USER MODEL
 DELETE USER MODEL
 VIEW/EDIT USER MODEL

Enter name of user model you wish to view/edit.
 user1

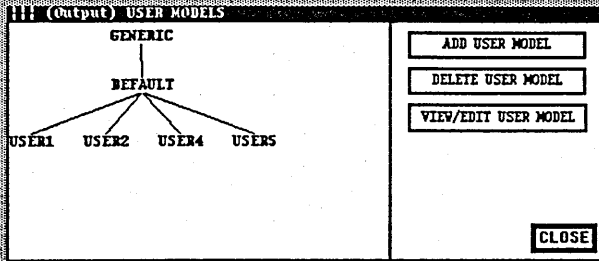
CLOSE

7.13

USER1 USER MODEL

Personal Profile		Cognitive Model	
COMMAND.EXP	Value: EXPERT HIGH MEDIUM LOW NONE	SPATIAL	Value: LOW HIGH
COMPUTER.USE	Value: OCCASIONAL FREQUENT		
ENTER VALUES		ENTER VALUES	
		CLOSE	

7.14



7.17

GENERIC USER MODEL

Personal Profile

COMMAND.EXP Value: EXPERT, HIGH, MEDIUM, LOW, NONE

COMPUTER.USE Value: OCCASIONAL, FREQUENT

Buttons: ADD CHARACTERISTIC, DELETE CHARACTERISTIC, EDIT CHARACTERISTIC

Cognitive Model

SPATIAL Value: LOW, HIGH

Buttons: ADD CHARACTERISTIC, DELETE CHARACTERISTIC, EDIT CHARACTERISTIC, CLOSE

Enter name of characteristic you wish to edit.
computer.use

Enter new alternative values for characteristic.
Entry must be of the following form:
(<value1> <value2> ... <valueN>)
Each value must be written as an atom.
(rare occasional frequent)

7.18

GENERIC USER MODEL

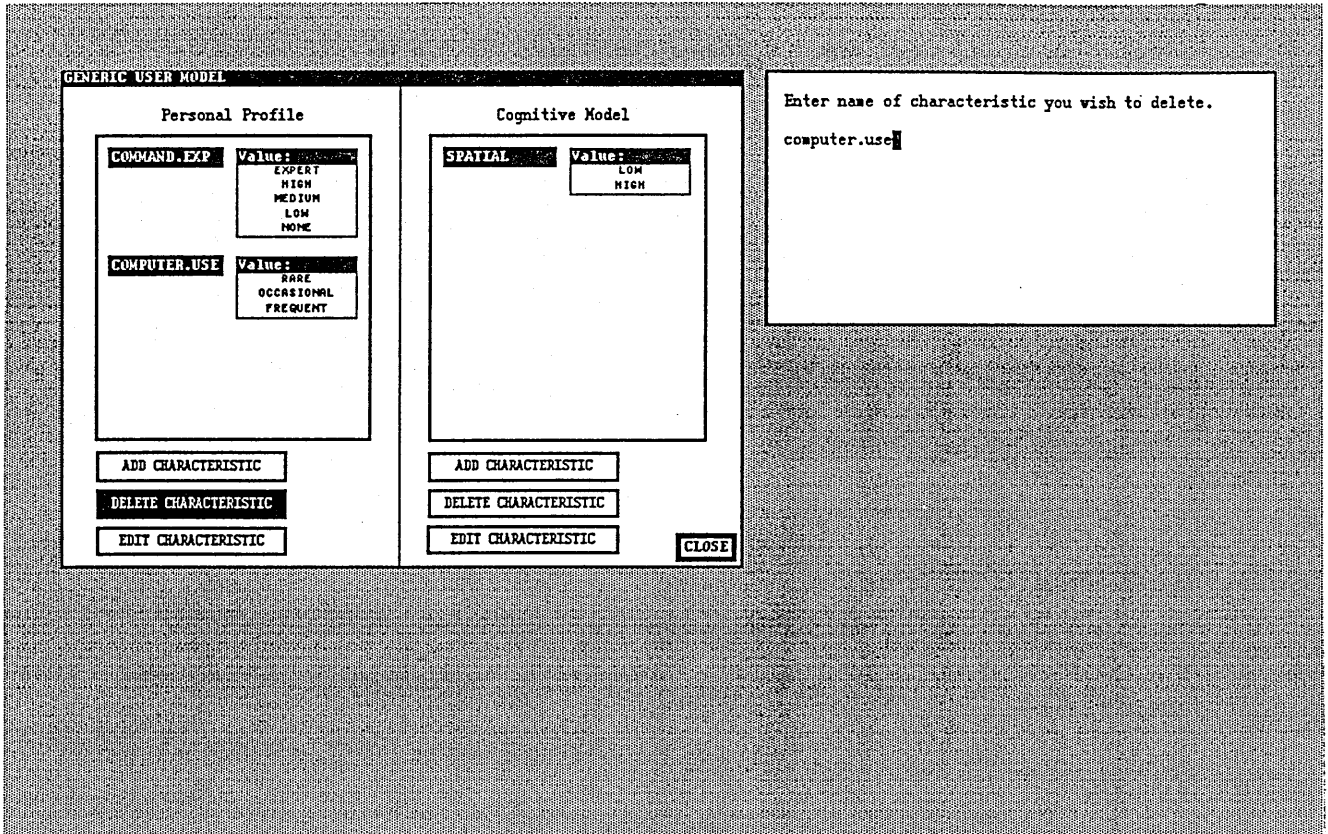
Personal Profile		Cognitive Model	
COMMAND.EXP	Value: EXPERT HIGH MEDIUM LOW NONE	SPATIAL	Value: LOW HIGH
COMPUTER.USE	Value: RARE OCCASIONAL FREQUENT		
ADD CHARACTERISTIC		ADD CHARACTERISTIC	
DELETE CHARACTERISTIC		DELETE CHARACTERISTIC	
EDIT CHARACTERISTIC		EDIT CHARACTERISTIC	
			CLOSE

7.19

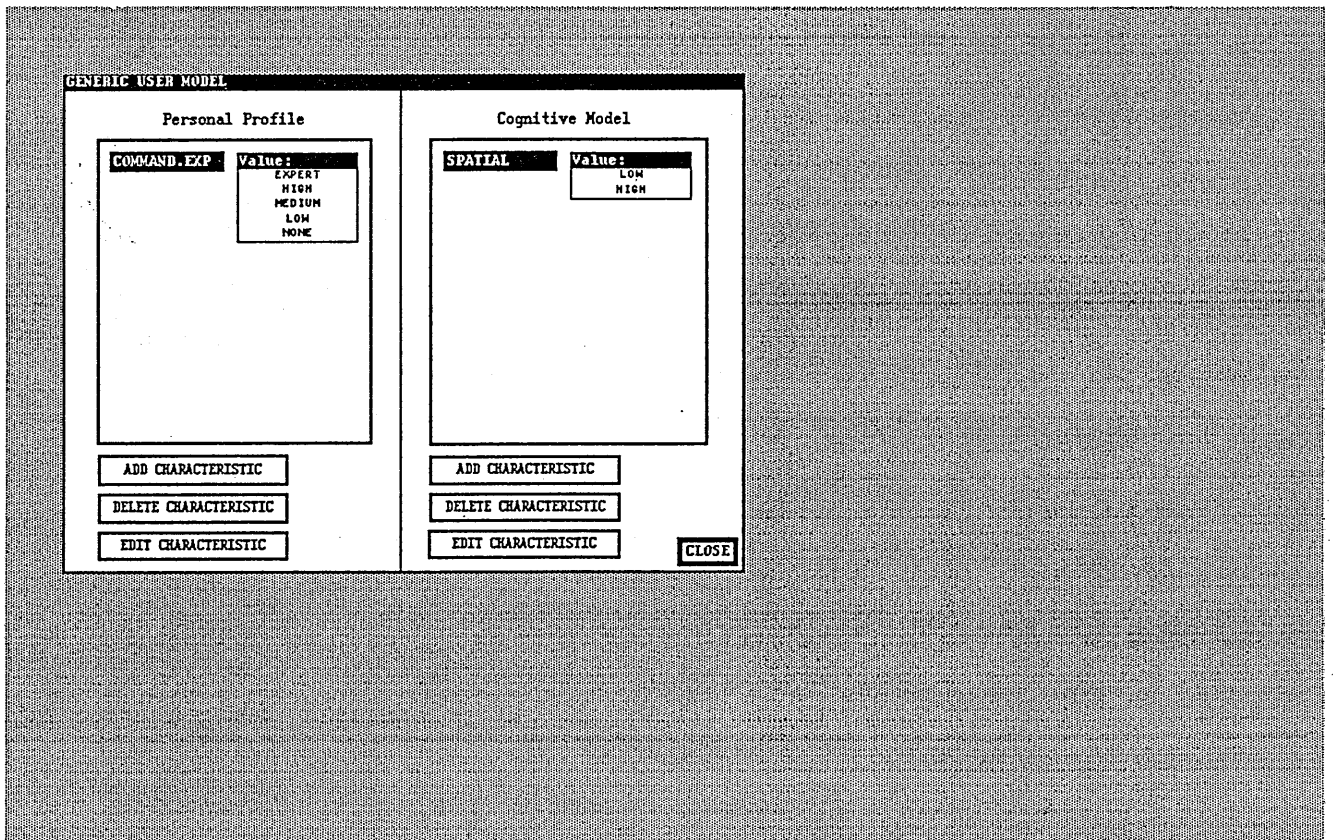
USER1 USER MODEL

Personal Profile		Cognitive Model	
COMMAND.EXP	Value: EXPERT HIGH MEDIUM LOW NONE	SPATIAL	Value: LOW HIGH
COMPUTER.USE	Value: RARE OCCASIONAL FREQUENT		
ENTER VALUES		ENTER VALUES	
			CLOSE

7.20



7.21



7.22

USER: USER MODEL

Personal Profile		Cognitive Model	
COMMAND EXP	Value: EXPERT HIGH MEDIUM LOW NONE	SPATIAL	Value: LOW HIGH
<input type="button" value="ENTER VALUES"/>		<input type="button" value="ENTER VALUES"/>	
		<input type="button" value="CLOSE"/>	

7.23

(Output) ADAPTATION RULES
 ADAPTATION RULES

8.1

(Output) ADAPTATION RULES
 ADAPTATION RULES---- HIGHSP

Enter name of rule (name must be an atom).

highsp

Enter weighting of rule (1-5, 1=high, 5=low).

1

8.2

Figure 8. Adaptation Rules.

ADAPTATION RULES
 ADAPTATION RULES - - - HIGHSP

Rule must be of the following form:

```

(IF
  (THE VALUE OF <USER MODEL CHARACTERISTIC> IS <VALUE>)*
THEN
  (THE INTERFACE OF USER IS <INTERFACE NAME>)*
  
```

* the rule can contain any number of clauses of this form.

Example:

```

(IF
  (THE VALUE OF SPATIAL IS LOW)
  (THE VALUE OF COMMAND.EXP IS NONE)
THEN
  (THE INTERFACE OF USER IS MENU))
  
```

Enter rule. Type \$ to end.

```

(if
  (the value of spatial is high)
  then
  (the interface of user is command))$
  
```

8.3

ADAPTATION RULES
 ADAPTATION RULES

EXPERTEXP
 HIGHEXP
 HIGHSP
 LOWSP/LOWEXP/FREQUSE
 LOWSP/LOWEXP/OCCUSE
 LOWSP/NOEXP/FREQUSE
 LOWSP/NOEXP/OCCUSE
 MEDIUMEXP

Enter name of rule (name must be an atom).

lowsp/noexp/occuse

Enter weighting of rule (1-5, 1=high, 5=low).

1

8.4

Output ADAPTATION RULES

ADAPTATION RULES

- EXPERTXP
- HIGHEXP
- HIGHSP
- LOWSP/LOWEXP/FREQUSE
- LOWSP/LOWEXP/OCCUSE
- LOWSP/NOEXP/FREQUSE
- LOWSP/NOEXP/OCCUSE
- MEDIUMEXP

ADD RULE

DELETE RULE

VIEW/EDIT RULE

CLOSE

Rule must be of the following form:

```
(IF
 (THE VALUE OF <USER MODEL CHARACTERISTIC> IS <VALUE>)*
 THEN
 (THE INTERFACE OF USER IS <INTERFACE NAME>)*)
```

* the rule can contain any number of clauses of this form.

Example:

```
(IF
 (THE VALUE OF SPATIAL IS LOW)
 (THE VALUE OF COMMAND.EXP IS NONE)
 THEN
 (THE INTERFACE OF USER IS MENU))
```

Enter rule. Type \$ to end.

```
(if
 (the value of spatial is low)
 (the value of command.exp is none)
 (the value of computer.use is occasional)
 then
 (the interface of user is menu))
```

8.5

Output ADAPTATION RULES

ADAPTATION RULES

- EXPERTXP
- HIGHEXP
- HIGHSP
- LOWSP/LOWEXP/FREQUSE
- LOWSP/LOWEXP/OCCUSE
- LOWSP/NOEXP/FREQUSE
- LOWSP/NOEXP/OCCUSE
- MEDIUMEXP

ADD RULE

DELETE RULE

VIEW/EDIT RULE

CLOSE

Enter name of rule you wish to view/edit.

mediumexp. Type \$ to finish.

```
(IF
 (THE VALUE OF COMMAND.EXP IS MEDIUM)
 THEN
 (THE INTERFACE OF USER IS COMMAND))
```

8.6

(Output) ADAPTATION RULES

ADAPTATION RULES

- EXPERTXP
- HIGHEXP
- HIGHSP
- LOWSP/LOWEXP/FREQUSE
- LOWSP/LOWEXP/OCCUSE
- LOWSP/NOEXP/FREQUSE
- LOWSP/NOEXP/OCCUSE
- MEDIUMEXP

ADD RULE

DELETE RULE

VIEW/EDIT RULE

CLOSE

Enter name of rule you wish to delete.

mediumexp

8.7

(Output) ADAPTATION RULES

ADAPTATION RULES

- EXPERTXP
- HIGHEXP
- HIGHSP
- LOWSP/LOWEXP/FREQUSE
- LOWSP/LOWEXP/OCCUSE
- LOWSP/NOEXP/FREQUSE
- LOWSP/NOEXP/OCCUSE

ADD RULE

DELETE RULE

VIEW/EDIT RULE

CLOSE

8.8

: (Output) DIALOGUE RECORD	
DIALOGUE.RECORD	<input type="button" value="ADD RECORD"/> <input type="button" value="DELETE RECORD"/> <input type="button" value="VIEW RECORD"/> <input type="button" value="CLOSE"/>

9.1

: (Output) DIALOGUE RECORD	
DIALOGUE.RECORD	<input type="button" value="ADD RECORD"/> <input type="button" value="DELETE RECORD"/> <input type="button" value="VIEW RECORD"/> <input type="button" value="CLOSE"/>

Enter characteristic of dialogue to be recorded.
 Entry must be of the following form:
 <attribute>.of.<aspect>
 For example: number.of.tasks, name.of.interface.
 number.of.errors

9.2

Figure 9. Dialogue Record.

!!! (Output) DIALOGUE RECORD	
DIALOGUE.RECORD — NUMBER.OF.ERRORS	<input type="button" value="ADD RECORD"/> <input type="button" value="DELETE RECORD"/> <input type="button" value="VIEW RECORD"/> <input type="button" value="CLOSE"/>

9.3

!!! (Output) DIALOGUE RECORD	
DIALOGUE.RECORD — <ul style="list-style-type: none"> NAME.OF.INTERFACE NUMBER.OF.ERRORS NUMBER.OF.TASKS 	<input type="button" value="ADD RECORD"/> <input type="button" value="DELETE RECORD"/> <input type="button" value="VIEW RECORD"/> <input type="button" value="CLOSE"/>

9.4

(Output) DIALOGUE RECORD

DIALOGUE.RECORD — NAME.OF.INTERFACE
 DIALOGUE.RECORD — NUMBER.OF.ERRORS
 DIALOGUE.RECORD — NUMBER.OF.TASKS

ADD RECORD

DELETE RECORD

VIEW RECORD

CLOSE

Enter name of user whose dialogue record you wish to view.

user4

NUMBER.OF.ERRORS = >1
 NUMBER.OF.TASKS = 7
 NAME.OF.INTERFACE = COMMAND

Hit any key to continue. █

9.5

(Output) DIALOGUE RECORD

DIALOGUE.RECORD — NAME.OF.INTERFACE
 DIALOGUE.RECORD — NUMBER.OF.ERRORS
 DIALOGUE.RECORD — NUMBER.OF.TASKS

ADD RECORD

DELETE RECORD

VIEW RECORD

CLOSE

Enter name of dialogue characteristic you wish to delete.

number.of.errors █

9.6

Output DIALOGUE RECORD

DIALOGUE RECORD

NAME OF INTERFACE

NUMBER OF TASKS

ADD RECORD

DELETE RECORD

VIEW RECORD

CLOSE

9.7

*** (Output) INFERENCE RULES ***

INFERENCE RULES

ADD RULE

DELETE RULE

VIEW/EDIT RULE

CLOSE

10.1

*** (Output) INFERENCE RULES ***

INFERENCE RULES----LOWSP/LOWEXP

ADD RULE

DELETE RULE

VIEW/EDIT RULE

CLOSE

Enter name of rule (name must be an atom).
 lowsp/lowexp
 Enter weighting of rule (1-5, 1-high, 5=low).

1

10.2

Figure 10. Inference Rules.

(Output) INFERENCE RULES
 INFERENCE.RULES- - - -LOWSP/LOWEXP

ADD RULE

DELETE RULE

VIEW/EDIT RULE

CLOSE

Rule must be of the following form:

```
(IF
 (THE <ATTRIBUTE> OF <DIALOGUE RECORD ASPECT> IS <VALUE>)*
 THEN
 (THE VALUE OF <USER MODEL CHARACTERISTIC> IS <VALUE>)*)
```

* the rule can contain any number of clauses of this form.

Example:

```
(IF
 (THE NAME OF INTERFACE IS COMMAND)
 (THE NUMBER OF ERRORS IS >1)
 THEN
 (THE VALUE OF SPATIAL IS LOW)
 (THE VALUE OF COMMAND.EXP IS LOW))
```

Enter rule. Type \$ to end.

```
(if
 (the name of interface is command)
 (the number of tasks is 12)
 (the number of errors is >1)
 then
 (the value of spatial is low)
 (the value of command.exp is low))
```

10.3

(Output) INFERENCE RULES
 INFERENCE.RULES- - - -LOWSP/LOWEXP

ADD RULE

DELETE RULE

VIEW/EDIT RULE

CLOSE

Enter name of rule you wish to view/edit.

lowsp/lowexp. Type \$ to finish.

```
(IF
 (THE NAME OF INTERFACE IS COMMAND)
 (THE NUMBER OF TASKS IS 12)
 (THE NUMBER OF ERRORS IS >1)
 THEN
 (THE VALUE OF SPATIAL IS LOW)
 (THE VALUE OF COMMAND.EXP IS LOW))
```

10.4

(Output) INFERENCE RULES

INFERENCE RULES-----LOWSP/LOWEXP

ADD RULE

DELETE RULE

VIEW/EDIT RULE

CLOSE

Enter name of rule you wish to delete.

lowsp/lowexp

10.5

(Output) INFERENCE RULES

INFERENCE RULES

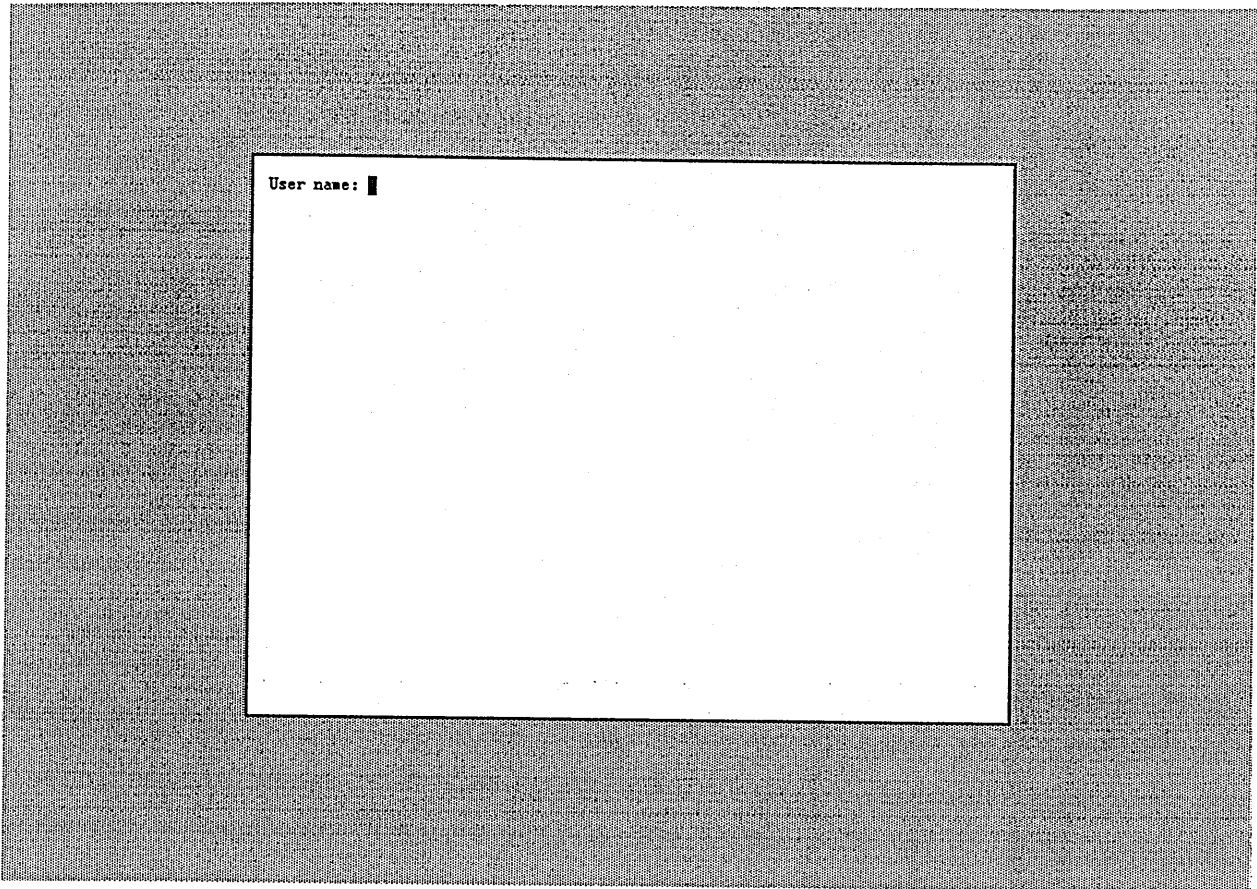
ADD RULE

DELETE RULE

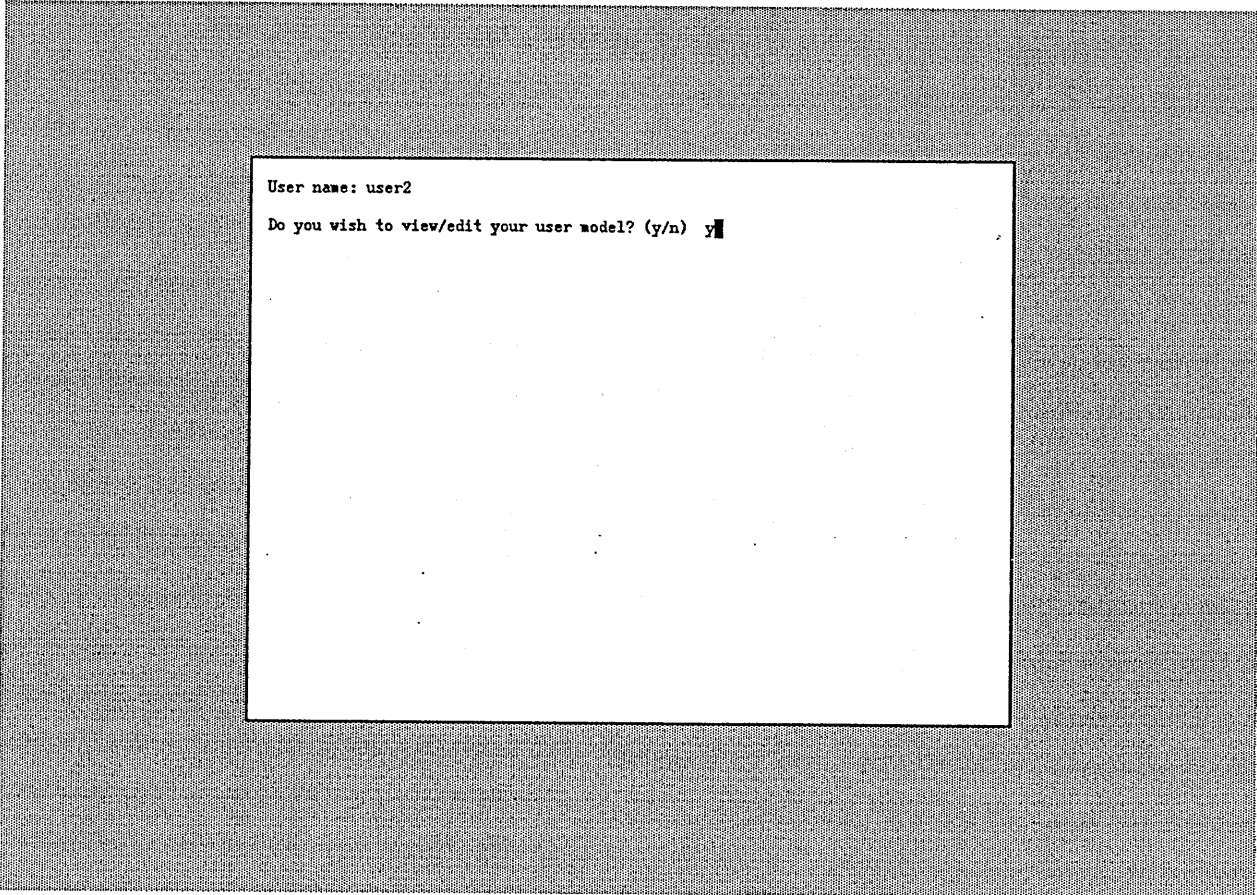
VIEW/EDIT RULE

CLOSE

10.6



11.1



11.2

Figure 11. Management System.

USER2 USER MODEL

Personal Profile	Cognitive Model
<p>COMMAND.EXP Value:</p> <p>EXPERT HIGH MEDIUM LOW NONE</p> <p>COMPUTER.USE Value:</p> <p>OCCASIONAL FREQUENT</p> <p>ENTER VALUES</p>	<p>SPATIAL Value:</p> <p>LOW HIGH</p> <p>ENTER VALUES</p> <p>CONTINUE</p>

11.3

USER2 USER MODEL

Personal Profile	Cognitive Model
<p>COMMAND.EXP Value:</p> <p>EXPERT HIGH MEDIUM LOW NONE</p> <p>COMPUTER.USE Value:</p> <p>OCCASIONAL FREQUENT</p> <p>ENTER VALUES</p>	<p>SPATIAL Value:</p> <p>LOW HIGH</p> <p>ENTER VALUES</p> <p>CONTINUE</p>

11.4


```
SYS> █
```

11.5

```
SYS> udb
UDB> select name from staff
List of staff:
FRENCH
HEATHCOTE
JONES
LAI
REDHEAD
UDB> select name from staff where course=computing
Invalid input.
UDB> select name from staff where course = computing
List of staff:
LAI
REDHEAD
UDB> █
```

11.6

MESSAGE FROM SHELL

Your user model has been altered by the system. The new values in your user model suggest that an alternative interface would be more suitable for you.

If you are not happy with any of the system generated values in your user model, you can change them. The system will then reconsider its choice of interface for you.

If you do not change any of the values in your user model, the alternative interface will automatically be presented to you.

Press return to view/edit your user model. █

11.7

USER MODEL

Personal Profile	Cognitive Model																		
<table border="1"><tr><td>COMMAND.EXP</td><td>Value:</td></tr><tr><td></td><td>EXPERT</td></tr><tr><td></td><td>HIGH</td></tr><tr><td></td><td>MEDIUM</td></tr><tr><td></td><td>LOW</td></tr><tr><td></td><td>NONE</td></tr></table>	COMMAND.EXP	Value:		EXPERT		HIGH		MEDIUM		LOW		NONE	<table border="1"><tr><td>SPATIAL</td><td>Value:</td></tr><tr><td></td><td>LOW</td></tr><tr><td></td><td>HIGH</td></tr></table>	SPATIAL	Value:		LOW		HIGH
COMMAND.EXP	Value:																		
	EXPERT																		
	HIGH																		
	MEDIUM																		
	LOW																		
	NONE																		
SPATIAL	Value:																		
	LOW																		
	HIGH																		
<table border="1"><tr><td>COMPUTER.USE</td><td>Value:</td></tr><tr><td></td><td>OCCASIONAL</td></tr><tr><td></td><td>FREQUENT</td></tr></table>	COMPUTER.USE	Value:		OCCASIONAL		FREQUENT													
COMPUTER.USE	Value:																		
	OCCASIONAL																		
	FREQUENT																		
<input type="button" value="ENTER VALUES"/>	<input type="button" value="ENTER VALUES"/>																		
	<input type="button" value="CONTINUE"/>																		

11.8

THE UNIVERSITY DATABASE

Hit any key to continue. █

11.9

University database

[Type 'quit' to exit the database]

The university database allows you to obtain information about:

1. students
2. staff

Enter code for category you are interested in (ie. '1' or '2'): █

11.10

USER2: USER MODEL

Personal Profile	Cognitive Model
<p>COMMAND.EXP Value: EXPERT HIGH MEDIUM LOW NONE</p> <p>COMPUTER.USE Value: OCCASIONAL FREQUENT</p> <p>ENTER VALUES</p>	<p>SPATIAL Value: LOW HIGH</p> <p>ENTER VALUES</p> <p>CONTINUE</p>

11.11

```
List of students:
ACKEROYD
ELLIS
GILLIES
REEVES
THOMPSON

UDB> select name from students where grade > 68

List of students:
ACKEROYD
ELLIS
GILLIES
THOMPSON

UDB> █
```

11.12