When does a Mouse become a Rat? or... Comparing Performance and Preferences in Direct Manipulation and Command Line Environment

K. MORGAN*†, R. L. MORRIS* AND S. GIBBS‡
* *Department of Psychology, University of Edinburgh, Edinburgh, EH8 9JZ*
‡ *Computer Studies Department, Napier Polytechnic, Edinburgh, EH11 4BN*

The question of which type of computer interface is 'better' for different types of users is of great interest to human factors engineers. It is argued that very few valid comparisons have been done on systems whose interfaces have been specifically designed and developed to be functionally identical. This paper covers just one study where such an interface has been developed for an investigation which has looked at the individual subject's psychological, social and environmental differences, and how those differences influence the subjects feelings, performance, and preferences with regard to various computer interaction methods. This paper concentrates on giving a description of the findings on interface preferences and performance.

Received March 1989, revised May 1989

1. INTRODUCTION

In the past, interface evaluation and comparison has mainly concentrated its effort on investigations into aspects of specific interaction methods. These investigations have included studies of UNIX-based errors,[1,2] the investigation of different command line keyword styles,[3] the comparison of sub-versions of the text editor EMACS,[4] the investigation of different versions of a robot control command-line system,[5] and the study of the organisation of menu items.[6] In comparison the field has expended very little effort in direct experimental comparison between systems that are functionally identical but which have different user interfaces. This could be due to a number of factors, the most important of which is probably the time and expense involved in preparing such comparisons.[7] The value of such comparisons can be seen by the relevance of one of the earliest such experiment comparisons,[8] which found that the full page editors were twice as fast as identical line editors.

A large proportion of the direct comparisons that have been done are between Natural Language Interfaces (NLI) and their alternatives. Comparisons between NLI and SQL database query systems[9,10] found that the command line SQL system showed a faster performance. Another study[11] compared NLI, command and menu systems, and could find no difference between any of them in speed, errors, or user attitudes.

A comparison between a form-filling interface and an identical command line system[12] found the form-filling interface to be faster, and noted that 11 of the 12 subjects tested preferred the form-filling method. It can be argued that such direct comparisons allow researchers to judge different interaction methods against one another in a way that is impossible with detailed single interface evaluations. It was for this reason, and because of the lack of clear, direct comparisons between WIMP and command line systems in the current HCI literature that the authors undertook this study.

The project being reported was created to compare computer users' performance and preferences in both command line and WIMP (Windows Icons Mice and Pull down menus) interfaces, using a specifically designed database query system as the target application. The present study is the first in a series of studies designed to explore a number of subject areas; the fear of computers, individuals who have habitually good or poor technology based interactions, how computer interfaces can help reduce naive users reluctance to use computer systems, and how individual personality characteristics influence a users preference of interaction method.

2. METHODS

2.1 Subjects

These were 72 second-year B.A. commerce students at Napier Polytechnic of Edinburgh (Sighthill Court). The sessions were part of their coursework, and attendance was compulsory. This group was computer naive and keyboard literate.

The statistics about this subject group were: Of the 55 who took part, 33 attended both sessions, 42 did the command, 46 did the WIMP. Ten of the subjects were male, 45 were female. Three of the subjects were non-UK nationals (European exchange students). The youngest in the subject group was 18, the oldest 24, four subjects were left handed.

This study involved students using a database program specially designed and developed to include two interfaces; one a command line, and the other a WIMP system. These two interfaces are matched for functionality and as far as possible error opportunity.

2.2 Design of the system

Requirements for the design of the study made it necessary to produce a system that could be inserted into

† Now at School of Information Science, Portsmouth Polytechnic, Milton Site, Locksnay Road, Portsmouth, Hampshire.

a classroom situation and be of educational benefit to the subjects. The authors produced an application that was concerned with the running of a college database, since it was a topic that both sexes would easily relate to.

2.2.1. *Design of commands and syntax*

On the basis of an analysis of the requirements for classes in database use over various levels of expertise, and some informal testing sessions a reduced set of simple commands were developed. These commands are both easily learned (target from the informal tests were 1 h to competence, so as to reduce total session length), and enable sufficiently complex actions to be carried out so that experienced users could still use the system.

2.2.2. *Comparison of the interfaces*

The command line and WIMP interfaces had identical command syntax parsers, and functionality. This complete matching was implemented from the command syntax right down to the exact wording of error messages, and help facilities. The WIMP system comprises a graphical interface with the standard windowing layout of pull down menu bars (see Fig. 1). As an example of how closely the two systems matched each other the delete command is illustrated. To delete in the command line system the following syntax would have to be typed by the user.

```
delete record-type key= key-number
```

An example of this would be '`delete student key= 1`'. In direct comparison, in the WIMP system the user would pull down the options menu, and scroll down until they reached the 'delete...' entry and click the mouse to indicate they required that action. This is analogous to typing '`delete`' in the command line version. The WIMP system will then present the user with a dialogue box (see Fig. 2), which prompts for the record type and the key number of the record to be deleted. Clicking on the 'OK' radio button (when the user is satisfied with their entries to the dialogue box) is equivalent to pressing the return key in the command line system. Both systems

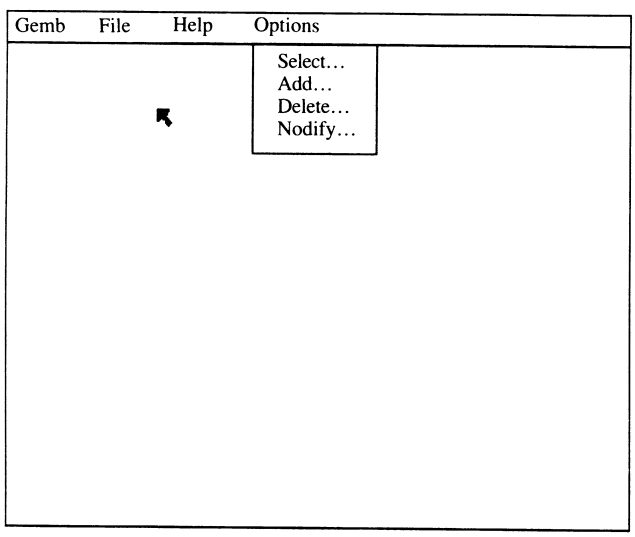

Figure 1. The WIMP system.

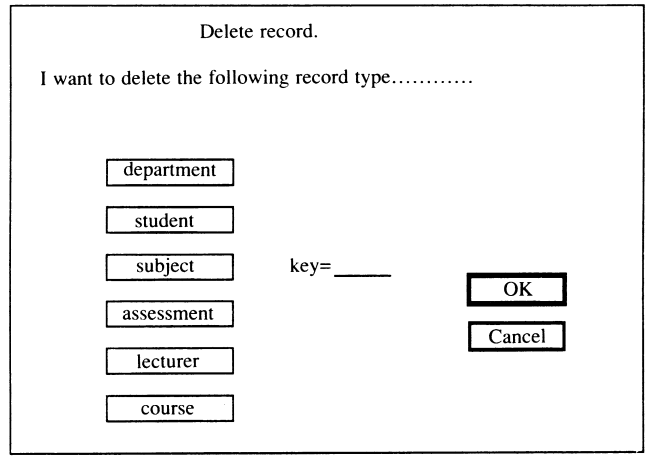

Figure 2. WIMP system with dialogue box.

then prompt the user with identical wording which forces the user to confirm this delete action.

2.3 Procedure

Prior to participation, subjects completed a specially created technology attitudes questionnaire (TAQ2) developed by the authors, which tried to identify any previous computer- or machine-based experience, and other psychological measures of individual differences such as adaptation to change, childhood experiences with machines, and working style.

Subjects were split into groups of 12 and issued unique subject numbers. The subjects experienced both versions of the system using a standard counterbalanced order within subjects design.

Following completion of a session, subjects completed an Evaluation of Interactive Computer Systems questionnaire (UEICS2 – also developed by the authors), which aims to capture the users attitudes towards the interface they have just used. At the end of the whole experiment subjects were asked to comment on their impressions and feelings with regard to the interface.

2.4 Performance measures

Two performance measures were used, as follows.

2.4.1. *Errors*

The system recorded the precise part of the users input that caused the rejection, with details of the actual and mean length of time between each error, and the type of the error (the type and sequence of commands to be issued are preset, and are identical for all users).

2.4.2. *Commands*

The system recorded the details of all the commands issued during each session along with the actual and mean amount of time between each command.

2.5 Before commencing we constructed the following hypotheses:

(1) That WIMP interfaces would be preferred by naive

users, but would have a slower throughput of work and a longer overall session time.

(2) That once competence was reached in the command line interface, it would be preferred by experienced users and naive users who are 'keyboard literate', due to the interface's greater information interchange rate, and the familiarity with the keyboard.

(3) That in all users, over both interfaces, the error rates will be greater in the command line interface throughout the training.

3. RESULTS

A series of specially designed and written analysis packages were created to analyse the data from the experiment. These results were then correlated against the questionnaire data (both TAQ2 and UEICS2), which was stored in a specifically designed interactive questionnaire database (using Symantec's Q & A Version 2.0).

3.1 Hypothesis 1, Part I: WIMP interfaces will have a slower throughput of work

The term 'work' is defined as the number of commands the user issued, and the mean time between commands. Looking at the data generated by the resident recording processes an apparent non-significant superiority in 'work' is shown for the command line system (the command line system shows an average number of commands of 39.667 and an inter-command time of 64.524 s, while the WIMP interface recorded 37.130 as its average number of commands, and inter-command time of 68.135 s). However since the command system had an automatic entry to the help system, if the return button was depressed with no other action, a possible artifact here would be if there was a disproportionate number of help commands being issued in one interface and not in another (help is counted as a command type). To check against this possibility the types of command issued can be seen in the graph 'Average number of command types' (Fig. 3). This histogram shows the number of the various command types issued in both systems ('WIMP' denotes the direct manipulation interface, and 'Command' the command line interface totals). It can be seen that a difference does exist, there being more of the help command type in the command line interface.

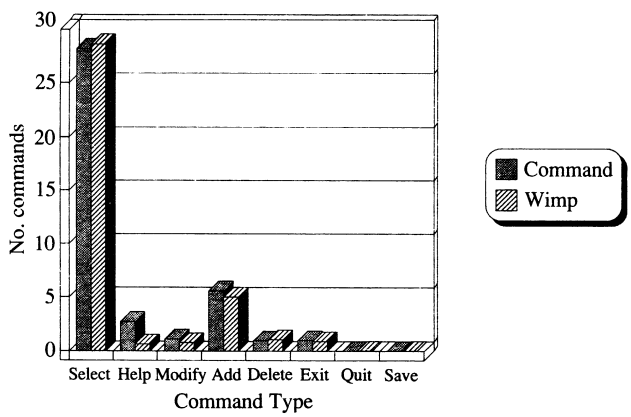

Figure 3. Average number of command types.

Total number of command types issued over both interfaces:

WIMP = 30 helps, Command = 115 helps.

This shows that users of the command system used the on-line help facilities significantly more than users of the WIMP system. This unbalanced use of the help system would have led to misleading totals, and in order to evaluate hypothesis 1 in this study we must correct for this bias. So if we remove the number of helps issued over both interfaces we find:

Total number of commands:

WIMP = 1678 with an N of 46 produces a mean number of commands = 36.478.
Command = 1551 with an N of 42 produces a mean number of commands = 36.928.

From these totals we can adjust the mean time between commands to be:

WIMP session length: 2394.804/36.478 = 65.650 s.
Command session length: 2463.595/36.928 = 66.712 s.

After having corrected for the 'help bias' the difference between the two interfaces is seen to have disappeared for number of commands, and to have exchanged places for the rate of work. Whereas previously the WIMP system had shown a greater time gap between commands (68.155 s to commands 64.524 s), now after adjustment for helps the command interface can be seen to have been slightly slower (although not significantly so). This means that work throughput for the two interfaces was found to be almost identical, therefore part I of hypothesis 1 has not been validated by this experiment.

3.1.1. Time between commands (Fig. 4)

Figure 4 shows the mean time between commands recorded by the system during the sessions for all the subjects over both interfaces (the graph terminates at the end of the session). The graph suggests that the command line system is faster in the early part of the study (as would be expected), but this graph also shows that the difference in the command line systems command rate decreased as the session progressed. There are at least two possible hypotheses that can be put forward to explain this, increased task complexity, and increased

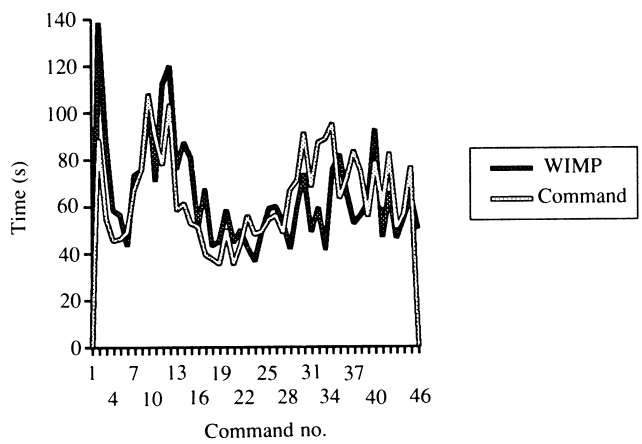

Figure 4. Mean time between commands.

task boredom. Taking these in turn, the tasks that the tutorial sheets required the subjects to undertake became more and more complex as the session progressed. It could be argued that the users of the command line system displayed this increased task complexity by an increased inter-command time gap; this would be used as increased think time by the user. The WIMP interface user also has a rise in inter-command time gap towards the end of the session, but this is very slight. Alternatively the increased task boredom interpretation would state the user of the command system has far fewer stimuli, and is therefore subject to task boredom at an earlier time than the WIMP user who has much greater stimulation for the entire task duration. There is some support for the notion that the task boredom explanation is more likely and the comments given in the post session questionnaires and post experimental essay clearly indicate that most users found the WIMP system more stimulating.

3.2 Hypothesis 1, Part II: WIMP interfaces have a longer session time

Taking the recorded data we find the following

	Mean	N	Min	Max	S.D.	Range
WIMP	2394.804	46	852.000	5633.000	722.742	4781.000
Command	2463.595	42	890.000	3523.000	506.621	2633.000

This shows a non-significant (near MCE) difference of 68 s in session length for the command line system. This means that part two of hypothesis one was also not validated. During this experiment with computer naive keyboard literate users the command interface had a longer session length than the WIMP users.

3.3 Hypothesis 1, Part III: WIMP interfaces will be preferred by naive users

The study has number of points at which the preference of the users can be detected. The last four questions of the post-session questionnaires assess the users' attitudes about the system used in that session. These questions came under the heading 'What were your overall reactions to the system?'. By using a Wilcoxon matched pairs test a measure of user preference can be determined.

Subjects tended to rate the WIMP system as better overall than the command line system ($N = 30$, $T = 138.500$, $Z = 1.933$, $P \sim 0.050$). The WIMP system was also rated as more stimulating ($N = 30$, $T = 111.000$, $Z = 2.499$, $P < 0.02$), as being easier ($N = 30$, $T = 137.000$, $Z = 1.964$, $P < 0.05$), and as having more adequate power ($N = 26$, $T = 108.000$, $Z = 1.714$, $P < 0.10$).

Taking these findings as the 'after use' affect we can clearly conclude that the subjects felt that the WIMP system was better in some very significant ways. It is not surprising that they found the 'feel' of the system more attractive and stimulating, but that they felt that the WIMP system had more 'power' is in direct contradiction to what most people in the HCI field would have anticipated.[13] Power is generally equated with command line environments, and yet in this study we found a strong suggestion that touch typist users (to whom keyboard skills would be second nature), found the WIMP environment more powerful. It will be extremely interesting to see if this continues in the following studies with different subject populations.

3.3.1. Second measure of interface preference

A second measure is from the post-experimental essay that the subjects composed on the subject of which of the two interfaces they preferred and which of the two interfaces they would recommend for business use. This data on the users preference is clouded by a possible artifact. Two things became obvious when the post-experimental essays were read. The first was that the subjects all presented very coherent and comprehensive descriptions of the factors involved in the two systems, even the 17 students who did not take part and never attended any of the sessions. It must be assumed that some collaboration had taken place among the subject population about what to say in this essay. The other factor that seemed to become apparent to S.G. was that the last interface used seemed to be preferred by the subjects. This was confirmed by an analysis, which split the data into presentation order, for details on this analysis see Section 3.6.

3.4 Hypothesis 2, Part I

Once competence is reached in the command line interface, it will be preferred by experienced users and naive users who are 'keyboard literate', due to the interface's greater information interchange rate, and the familiarity with the keyboard.

This subject group was keyboard literate (having been taught touch typing for 2 years previously to the experiment), and yet from the data we find strong evidence that these subjects preferred the WIMP system, thus disproving hypothesis 2. It remains to be seen if an expert keyboard-literate group also prefer the WIMP environment.

3.5 Hypothesis 3

In all users, over both interfaces, the error rates will be greater in the command line interface throughout the training.

If we look at the recorded data for errors we find the following:

	Command	WIMP	T	P
Mean no. errors	29.9	12.9	6.82	$\ll 0.01$
Mean time between errors in seconds	109.2	237.8	−5.45	$\ll 0.01$

From this it can be clearly seen that a highly significant difference exists between the number of errors (and the mean time between errors over both interfaces, since that is a mathematical function of number of errors and session length), the WIMP interface having a highly significant advantage in both cases. This confirms experimental hypothesis 3.

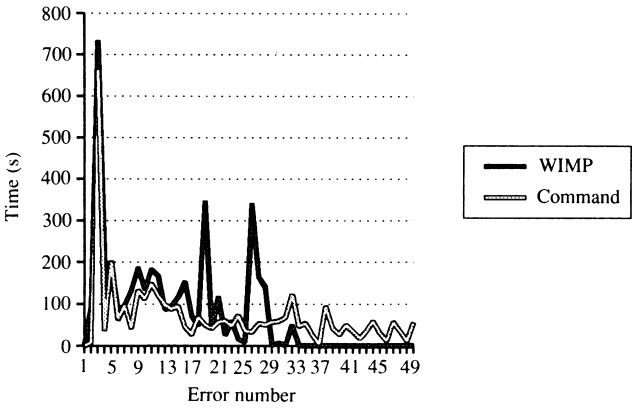

Figure 5. Average time between errors for WIMP and command line interfaces.

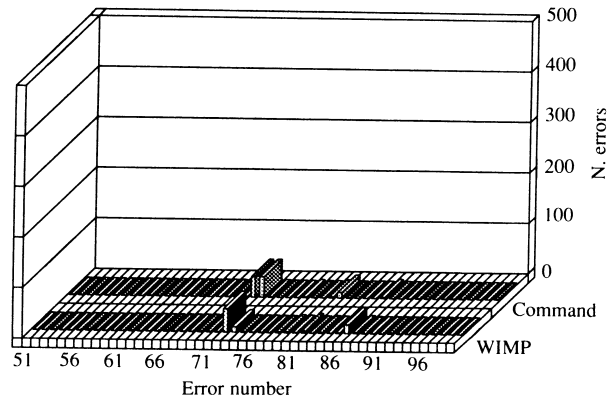

Figure 7. Error types over command and WIMP interfaces.

3.5.1. Time between errors

Figure 5 shows that the average time between errors for the WIMP interface is longer than that for the command line interface. However this difference is reduced as the session length increases. The graph shows a dramatic peak in the beginning of the session, this would correspond with the time between reading and comprehending what was required of the subject from the tutorial sheet. After that initial period (the first to third error) the two graphs settle down to a fairly constant frequency pattern, except for the two very large peaks on the WIMP system (errors 19, 26, 27). Since the largest number of errors made on the WIMP interface was 33, the graphs show a zero error rate from that point on.

3.5.2. Error types over each interface (Fig. 6, 7)

Figures 6 and 7 show the frequency and types of errors produced by both systems. It is split into two parts, part one shows error numbers 1 to 50, and part two shows error numbers between 51 and 100.

(i) *Syntax errors.* Syntax errors are more important and occur where the user has mistyped either a verb, command or condition, or has incorrectly formed the command structure. It may indicate lack of understanding of the required command syntax, or a mistake in command entry. To compare error rates on both systems we consider only those errors which are possible on both systems, thereby eliminating some typing or spelling errors eliminated by the WIMP system. The mean number of such syntax error messages allocated to the WIMP system is 9.608 and 13.875 for command line.

(ii) *Semantic errors.* Semantic errors are those where a user has attempted to carry out an action on the system which did not work, because the user had misunderstood the method required by the system to achieve that act (called an action error within this evaluation system), or the user did not understand what an object, say a name could be or do (called an object error within this evaluation system).

Many 'semantic' error messages also overlap into syntax errors. It is possible to produce a second category for errors called a 'probable category'. This allows us to remove errors messages that are not likely to be semantic errors, such as 'invalid student sex type', where it is highly probable that the user mistyped the gender field (syntax error), rather than the possibility that the user did not understand the concept of a 'sex' (male or female). After doing this for all the errors we find that the mean for WIMP is 4.978 errors and 14.425 for the command line.

Figure 8 shows an almost even match between syntactic and semantic errors in the command line system. This suggests that the lack of a clearly provided 'mental map',[14–16] of how the system works in command line interfaces makes the cognitive load for both syntax and semantic detail about equal. The user is left to create their own mental model of what to use and how they are supposed to achieve the task.

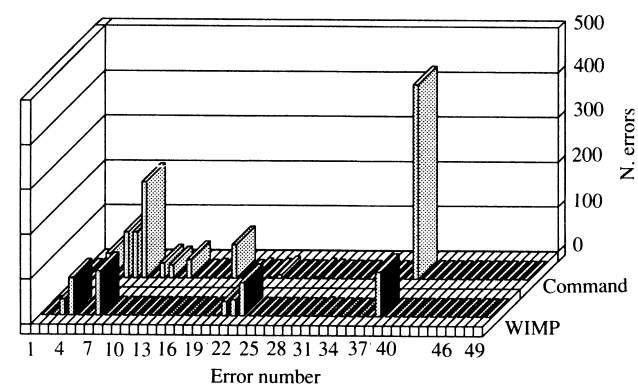

Figure 6. Error types over command and WIMP interfaces.

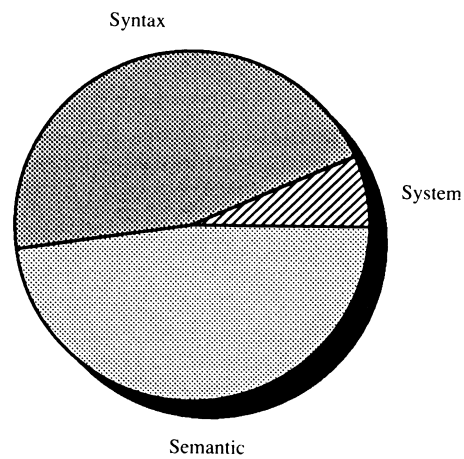

Figure 8. Syntactic and semantic adjusted errors in command line system, showing an almost even match.

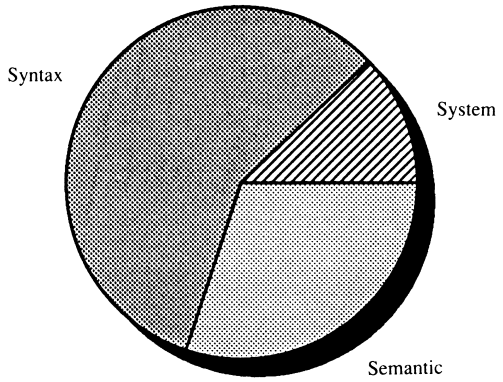

Figure 9. Syntactic and semantic adjusted errors in WIMP-based system, illustrating a large difference.

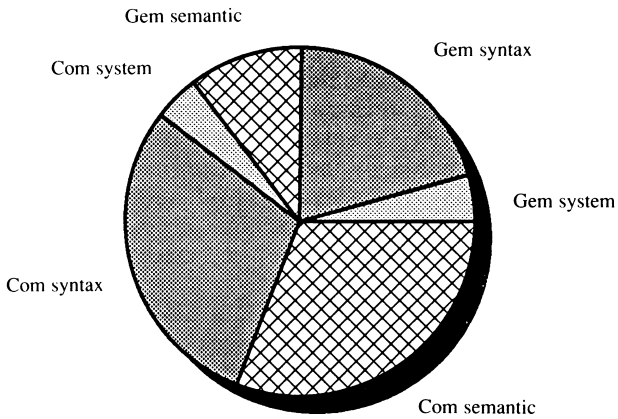

Figure 10. Combined detail from both interfaces (errors adjusted).

In contrast Fig. 9 shows a large difference between syntactic and semantic errors for users of the WIMP based system. We propose that this is caused by the clear and imposed mental model of how the system works that is presented by a graphical interface. Such a system drastically reduces the amount of cognitive load experienced by the user, and this reduces the number of semantic errors.

Figure 10 shows the detail from both interfaces combined. In this diagram, the difference in the number of semantic errors created by each interface is shown clearly. Although the WIMP system is better with regard to all error types, the difference is less marked in the case of syntax errors.

3.6 Order effects

To test if any presentation effects had occurred we segregated the data into presentation order, and carried out a Kruskal–Wallis ANOVA by ranks on the data.

We found slight (non-significant) increases in session length and number of errors made in the second sessions, when compared to the first sessions. We also found an increase in the number of commands issued on both interfaces. This would be expected by a simple learning effect, and the increased relaxation that subjects would experience in their second session. It is proposed that the slight difference found in favour of the WIMP sessions is due to the increased 'user friendliness' of that interface, which would enhance the learning/relaxation effect.

3.6.1. Preferences from the post-experimental essays

Subjects who experienced the WIMP system last, preferred the WIMP system in their essay, and subjects who experienced the command system last preferred the command system ($H(1, N = 31) = 9.475$, $P \ll 0.01$). It was a surprise to find that the command system was preferred by those people who experienced it last, it is of course possible that as the user gained experience with the system they changed to preferring the command system (as would be predicted by hypothesis 2). However our interpretation of this finding is that we are seeing the factor of the most recent experience influencing a compulsory essays content, which is not unknown in an academic situation, and this must call doubt upon the preference data gathered from these essays.

4. SUMMARY

In this paper we have described the results from the in-session recording software from a specifically designed HCI interface comparison tool. The small sample size (55) reflects the increasing difficulty in obtaining truly computer naive user/subjects. Our findings with keyboard literate computer naive users strongly suggest that WIMP interfaces are faster, easier, less error prone, and more stimulating for naive touch typists than a functionally identical command line interface. Further research is needed to see if the WIMP interface's superiority was maintained if the subjects use the system over a long period. In this paper we have also found a marked difference in the types of errors produced in the two systems, even after they have been adjusted to exclude biases.

Acknowledgment

The authors' sincere thanks go to the students who agreed to be subjects in this study, to Napier Polytechnic for the provision of resources required for this study, to Dr Hamish MacLeod of Edinburgh University's Psychology Department for his advice and support, and the referee whose suggestions were most helpful.

REFERENCES

1. R. E. Kraut, S. J. Hanson and J. M. Faber, Command use and interface design. *Proc. CHI '83 Conference on Human Factors in Computing Systems*, 120–123 (1983).
2. S. J. Hanson, R. E. Kraut and J. M. Farber, Interface design and multivariate analysis of UNIX command use. *ACM Transactions on Office Information Systems*, **2** (1), 42–57 (Jan. 1984).
3. H. Ledgard, J. Whiteside, A. Singer and W. Seymour, The natural language of interactive systems. *Communications of the ACM*, **23**, 556–563 (1980).
4. T. R. G. Green and S. J. Payne, Organization and learnability in computer languages. *International Journal of Man Machine Studies*, **21**, 7–18 (1984).
5. J. M. Carroll, Learning, using and designing command paradigms. *Human Learning* **1** (1), 31–62 (1982).
6. L. S. Liebelt, J. E. McDonald, J. D. Stone and J. Karat,

The effect of organization on learning menu access. *Proc. Human Factors Society, 26th Annual Meeting*, 546–550 (1982).
7. O. Iseki and B. Shneiderman, Applying direct manipulation concepts: Direct Manipulation Disk Operating System (DMDOS). *Software Engineering Notes*, **11**, 2 (March 1986).
8. T. Roberts, Evaluation of computer text editors. Ph.D. Dissertation, Stanford University (1980).
9. D. Small and L. Weldon, An experimental comparison of natural and structured query languages. *Human Factors* **25**, 253–263 (1983).
10. M. Jarke, J. A. Turner, E. A. Stohr, Y. Vassiliou, N. H. White and K. Michielson, A field evaluation of natural language for data retrieval. *IEEE Transactions on Software Engineering*, **SE-11** (1), 97–113 (Jan. 1985).
11. A. G. Hauptmann and B. F. Green, A comparison of command, menu selection and natural language computer programs. *Behaviour and Information Technology* **2** (2), 163–178 (1983).
12. W. C. Ogden and J. M. Boyle, Evaluating human computer dialog styles: Command vs. form/fill-in for report modification. *Proc. Human Factors Society, 26th Annual Meeting*. Human Factors Society, Santa Monica, CA, 542–545 (1982).
13. B. Shneiderman and R. Mayer, Syntactic/semantic interactions in programmer behaviour: a model and experimental results. *International Journal of Computer and Information Sciences*, **8** (3), 219–239 (1979). Reprinted in B. Curtis (ed.), *Human Factors in Software Development*, IEEE Computer Society EHO 185–9, 9–23 (1981).
14. T. Nelson, Interactive systems and the design of virtuality. *Creative Computing*, **6** (11), 56ff, (Nov. 1980), and **6** (12). 94ff, (Dec. 1980).
15. C. Rutkowski, An introduction to the Human Applications Standard Computer Interface, Part 1: Theory and principles. *BYTE* **7** (11), 291–310 (Oct. 1982).
16. E. L. Hutchins, J. D. Hollan and D. A. Norman, Direct manipulation interfaces, in D. A. Norman and S. W. Draper (eds), *User Centered System Design: New Perspectives on Human–Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, N.J. (1986).
17. T. S. Tullis, An evaluation of alphanumeric, graphic and color information displays. *Human Factors* **23**, 541–550 (1981).
18. B. Pace, Color combinations and contrast reversals on visual display units. *Proc. of the Human Factors Society, 28th Annual Meeting*. Human Factors Society, Santa Monica, CA, 326–330 (1984).

Announcements

23–26 SEPTEMBER 1991

BMVC 91, organised by the Turing Institute for BMVA – **British Machine Vision Conference,** University of Glasgow

The British Machine Vision Conference is the premier annual UK national conference for Machine Vision and related topics. For the second year this conference has operated as a merger between the BPRA and Alvey Vision conferences.

The emphasis is on UK research being undertaken through national or international collaborative projects, providing a forum for the presentation and discussion of the latest results of investigations. Papers from other nations, especially those collaborating with UK groups, are also being presented.

A printed copy of the *Proceedings* will be available to delegates at the conference, and a selection of the best papers will be published separately in a special issue of *Image and Vision Computing Journal*.

The following is a list of topics:
- Image processing and feature extraction
- Object recognition and scene analysis
- Reconstruction of 3D shape
- Advanced pattern analysis
- Computational issues in visual perception
- Robotic vision and sensor fusion
- Practical applications in machine vision
- Model-based coding
- Architectures for vision systems
- Active vision

Registration: Tanya Oliver, BMVC 91, The Turing Institute, George House, 36 North Hanover Street, Glasgow G1 2AD. Tel: +41 552 6400.

7–9 APRIL 1992

ESCA Workshop, 'Comparing Speech Signal Representations', Sheffield

Raison d'être

There is a growing range of novel techniques for speech analysis. Whilst the utility of an analysis or representation can only be judged with respect to the intended application, it is often desirable to compare techniques applied to the same data. The lack of an agreed common data set and presentation format makes this difficult to achieve in practice.

The aim of this workshop is to overcome these problems by concentrating on a small amount of common material displayed in a common style. Workshop participants will present results of their own analyses of this material.

Topics

Proposals in the following subject areas:
- perception-based analysis;
- production-based modelling;
- time–frequency representations;
- neural network analysis;
- novel visualisation techniques.

Tutorial

There will be tutorial sessions providing an introduction to both traditional and contemporary speech analysis techniques and signal representations, again using the common material.

Speech material

Both workshop and tutorials will focus on the following:
- a single sentence from a standard database;
- the same sentence but with added broadband noise;
- the same sentence re-recorded in a noisy environment;
- one other spontaneously produced sentence recorded in a natural environment.

Copies of this material will be distributed to all participants well in advance of the workshop.

Proceedings

The proceedings will form a special-format book which will consist of tutorial chapters, 4-page textual contributions from participants, commentaries by session chairpersons and a loose-leaf section containing up to 6 pages of displays for each participant.

Fees

Registration fees will be of the order of 250 ECUs (excluding tutorials, meals and accommodation), with reductions for ESCA members. Fees for the tutorial day will be of a similar magnitude.

European Speech Communication Association (ESCA)

ESCA is a non-profit-making organisation for promoting speech communication science and technology in a European context. For membership and other information, please contact the ESCA Secretariat (BP 7, B-1040 Bruxelles 40, Belgium).

Local Organising Committee: Steve Beet, Martin Cooke, Malcolm Crawford, Phil Green, Mike Pont.

SPeech Lab At SHeffield, Portobello Centre, Sheffield University, Sheffield S10 2TN. Tel: (+44) 742 768555, ext. 5577. Fax: (+44) 742 750318. Email: ESCA@UK.AC.SHEFFIELD.PRIMEA.