Knowledge acquisition and conceptual models: a cognitive analysis of the interface.

Andrew Dillon

The HUSAT Research Centre

The Elms

Elms Grove.

Loughborough,

Leics. LE11 1RG

Understanding how users process the information available to them through the computer interface can greatly enhance our abilities to design usable systems. This paper details the results of a longitudinal psychological experiment investigating the effect of interface style on user performance, knowledge acquisition and conceptual model development. Through the use of standard performance measures, interactive error scoring and protocol analysis techniques it becomes possible to identify crucial psychological factors in successful human computer use. Results indicate that a distinction between "deep" and "shallow" knowledge of system functioning can be drawn where both types of user appear to interact identically with the machine although significant differences in their respective knowledge exists. The effect of these differences on user ability to perform under stress and transfer to similar systems is noted. Implications for the design of usable systems are discussed.

Published in: D.Diaper and R.Winder (eds.) *People and Computers III,* Cambridge, Cambridge University Press, 1987.

Introduction

The study of usable or "friendly" system design has typically attempted to identify specific stylistic features of the interface that affect user performance. As a result of this approach a large body of data now exists on isolated factors of interface usability such as command language syntax (Barnard et al,1982), menu content (Snowberry et al,1983, Savage et al,1982) and system response time (Goodman and Spence,1978, Shneiderman, 1987). However, while data of this nature is preferable to the anecdotal evidence and subjective recommendations employed by designers in the early 1970's, they often fail to provide context independent information that recognises individual user differences and adaptation over time.

The failure of such research to have the desired impact is due in part to the absence of an overall psychological model of the user based on firm empirical evidence within which specific recommendations on interface features may be embraced. The computer interface provides the user with the means of communication with the system. This process of human computer communication is a psychologically complex one that is little understood. Barnard and Hammond (1982) therefore suggest that usability cannot be defined by any one factor or feature but rather is multiply determined by any number of user and task variables. Analysing the user and his/her cognitive skills and interpreting the relationship of interface to use in terms of the cognitive processes underlying interaction would seem to offer the best way of gaining an understanding of users' needs with respect to interface design.

The User as a Psychological Being

The user of an interactive computer system brings a complex range of psychological skills and attributes to bear on his/her interactions with the system such as knowledge, experience, motivation and personality. Effective interactive communication is seen as requiring the user to learn, assimilate and structure relevant information concerning the system and its uses into a meaningful conceptual model of operation (Hammond et al,1982, Norman,1981). The question of how such knowledge is acquired and the influence of particular interface styles on this knowledge acquisition are essentially psychological in nature, and of central importance for the successful design of usable or "user-friendly" systems.

From this perspective the user of a system may therefore be understood as receiving information from the screen, processing it and responding to it in some way i.e. the user is cognitively active. How such processing and responding is influenced by interface style and user ability remains poorly understood.

Surprisingly, while some evidence exists detailing the large differences that exist between experts and novices on particular tasks (e.g. Carroll and Mack,1984, Rosson,1984), there is little data documenting developments and changes in user knowledge over time. Yet users are not static entities and without analysing the development of user proficiency it is impossible to arrive at any real understanding of usability. Stevens (1983) makes the point that it is not "user-friendly" to make a system easy to use now if it is later limiting in terms of the user's increased knowledge over time. Indeed it is even possible that highly supportive interfaces hamper user development by providing the user with an inaccurate model of system activity and an over-reliance on prompts and facilities.

Currently we lack suitable empirical data on the development of user knowledge over time to judge the long term effects of particular interface styles and make informed decisions on such issues. Psychological investigations of the evolving process of interaction are thus essential.

This attempt to "get inside the head" of the user, so to speak marks a move away from the standard analysis of user performance in terms of speed and error rates, towards the attempted analysis of knowledge acquisition and conceptual model development. While the standard measures of speed and accuracy provide objective and reliable indices of performance, they offer no insight into how the user perceives the interface and why s/he interacts as s/he does. They provide a final "score" of performance without describing the performance itself which is surely crucial for any attempt at understanding the process of human-computer interaction. It is the analysis of the process itself that is of concern in the present study.

Overview of the Study

The full study involved a sample of 20 subjects interacting with one of two experimental interfaces for twelve sessions, performing a variety of database management tasks. The two interfaces were developed over a period of time on the basis of survey and experimental data on the topic of interface usability (see Dillon,1986 for complete details). They differed in terms of their inclusion or exclusion of user-friendly features.

Specifically, the "friendly" interface contained query-in-depth help facilities (which allowed the user to access either brief prompts without leaving the current screen, or to obtain more detailed screenfuls of information), provided explicit feedback and employed user oriented language. The "unfriendly" system lacked help facilities, provided no explicit feedback and employed a machine oriented language.

Table 1 (below) summarises these differences.

Cond.1 "Friendly"	Cond.2 "Unfriendly"
Q-I-D Help facilities	No Help Facilities
Explicit feedback	Implicit feedback
Menu driven	Menu driven
User oriented language	Machine oriented language

Table 1. Comparison of features on experimental interfaces.

The difference in the language employed manifested itself through error messages, prompts and system responses following guidelines similar to Shneiderman (1982). So e.g. if the user presses an inappropriate key when selecting an option from menu the "friendly" interface responds with:

"That is not an available option. Please try again"

whereas the "unfriendly" interface responds with:

"Illegal cmd. Err. code 002. Wait"

whereupon both interfaces then return you to the main menu.

Differences in feedback termed explicit and implicit refer to the degree of specificity provided. Explicit feedback occurs in direct response to the previous user action, clearly informing the user of the the effect of his/her input and the current status of the system. Implicit feedback can be best understood as the routine screen changes and system noises that the user invariably interprets as signs of system activity. Every interface to some extent provides implicit feedback.

In this way the interfaces are seen to differ in terms of information provision. This study attempts to understand how important such information is and how, if at all, it effects long term user performance. By so doing it is hoped to arrive at an understanding of the process of interaction from the perspective of the user.

Data Collection and Method

In order to maximise data collection a variety of techniques were employed. These included interactive error analysis (which checked users path to task completion and ability to satisfy input parameters), time, and performance measures and before/after attitude ratings. Verbal protocols were recorded of the user interacting with the system over the course of the trials.

Twenty subjects (10 per condition), all novice users, were randomly assigned to one or other interface. Each Ss was required to interact unaided with the system for 12 trials, each consisting of three routine database tasks at the rate of one trial per day (time of day controlled for arousal effects). Typical interactions required the user to create, access or edit files containing information on books and authors.

A brief introduction to the system was provided by the experimenter to each subject before the commencement of the initial trial. The nature of the study was explained to the Ss.. Pilot studies indicated that naive users could reasonably expect to master either system within 6 or 7 sessions (or less).

Results and Discussion

Although a large amount of data was generated by this study, for the purposes of the present investigation the analysis will concentrate on the verbal protocols over the first 10 trials. The subsequent two trials were run as independent further studies on the effects of the initial 10 trial investigation and will not be reported in detail here.   Suffice to recognise that significant differences were initially observed between conditions , with the users of the "friendly" interface performing faster and more accurately than their counterparts, until after trial 7 when such performance differences disappeared.

These results may be taken to indicate that interface style may only be an important variable during the early stages of use and have little effect over long term use. However the protocol data suggest differently.

Generally these protocols paint a picture of users initially interacting with the system in a tentative fashion, searching the screen for information and attempting input through a mixture of common-sense and trial and error. Interaction at this stage is slow and difficult, as users make numerous false inferences on system activity, perceiving inconsistencies in the computer's responses.

This contrasts with the style of interactive performance that becomes common to all subjects as the trials continue. Here users interact in a comparatively trouble-free manner, smoothly entering commands, clearly understanding system responses and showing an ability to predict system operation several stages in advance. These distinctions concur with those commonly found in the literature e.g. Wilson et al (1985).

In order to attempt quantification of these qualitative differences three raters carried out a detailed protocol analysis. The protocols were parsed into single interactive attempts, operationally defined as any input that ended with the user hitting the carriage return. The Ss articulation at that point was taken in context and grouped by the raters into a classification scheme. Through a process of refinement three broad descriptive categories of interaction emerged. These were:

1 *Confusion:* here the S lacks any understanding of the situation and interacts on the basis of guesses and "what sounds right". Typical articulations at this point are along the lines of "try it and see what happens".

2 *Rationality:* this category represents interactions that are less the product of chance and guesswork and more of reasoned decision making. Ss tending to remark e.g. that such an option is likely to work on the basis of prior experience, or the discounting of alternative options.

3 *Knowledge:* protocol data that fell into this category typically emerged during the later trials when all Ss had some knowledge of the system and its operation. Typically Ss could specify what they were going to do several stages in advance e.g. "to delete that item I must edit the file, so I'll read it from disk, get into edit mode and check it then..." . Such verbalisations tended to precede several rapid interactive cycles.

Each subject's series of protocols were "scored" using this classification (inter-rater reliability >.89). A 2x3x12 (condition x category x trial) analysis of variance was carried out on the data (d.f.=1,500 unless otherwise stated) indicating that all users becoming significantly more knowledgeable ("Friendly" cond., F=608.2 p<.001, "Unfriendly" cond, F=332.7,p<001) and less confused ("Friendly" cond., F=10.48,p<.01, "Unfriendly" cond., F=5.06,p<025) over the course of the trials.

However an interesting finding emerged from the comparative analysis of protocols between conditions. As noted earlier, initial significant differences (in terms of speed and accuracy) between users of either interface disappeared over time. In other words quality of interface appeared to have only a limited effect on user performance. Yet the protocol analysis paints a very different picture. The results of this suggest that significant differences do indeed exist between conditions, not only as expected during the early phases of the study, but also at the end of trial 10 with users of the "friendly" system appearing significantly more knowledgeable than users of the "unfriendly" system (F=14.77,p<.001).

This result suggests that although behavioural measures may suggest that users are performing at similar levels of competency, significant differences can exist at a cognitive or knowledge level. Indeed the subsequent trials (not detailed here) forced users to work under stress and transfer their knowledge to a stylistically different interface. Significant differences reappeared here on all behavioural measures suggesting that the quality of user knowledge is an important aspect of interaction.

The important question here is what do we mean by knowledgeable and how is this knowledge acquired? The protocols indicate that users initially interact on the basis of common sense (i.e. what appears likely to work, or what sounds a likely option on the menu) and through a slow and often difficult process of attempted input sequences begin to associate specific inputs (user actions) with particular outputs (system responses).

In the early stages of interaction it may be as simple as linking depression of the space bar with cursor movement on screen, but the user soon begins to build up more specific task-dependent associations e.g. the inputting of the letter I (representing the Input option on the main menu) with the appearance of the "NAME:" prompt on screen. These specific associations provide the basis of an interactive syntax for the user e.g."if I type R it will give me a filename prompt and if I type a blank it will take me to the menu again..".

Obviously there are a large number of potential associations that any user would need to make in this parrot-like fashion if s/he were to rely solely on this means of building up knowledge. It can be observed from the protocols that this is not the only means of knowledge acquisition and that a second, less rule-based and more inferential type of knowledge exists.

This second type of knowledge can be seen as generalised interactive knowledge of how the system works and what it requires from the user for operation. This is a sort of context-independent knowledge of how to approach the computer and why it behaves as it does. It is this type of knowledge that users rely on to guide their behaviour when uncertain of the required input.

Rather than blindly link input with output in a non- discriminatory fashion, the human user seems to extract more than the basic associative information from that link and begins to perceive more general rules e.g. "the computer provides a prompt which I must respond to.." or "you can only perform any type of function when you have selected it from the menu first.".

Through a process of chunking particular associative sequences together (probably on the basis of their contingency in time and task) and inferring from these groups of "knowledge fragments" (Barnard and Hammond, 1982) the operational logic of the system, the user acquires what can be termed a conceptual model of the system . In other words rather than just acquire a sequence of action-response associations on how to perform an editing function, the user seems to actively seek out from such information a model of the machine that explains how the system works. Thus users begin to see that files may be in use even if they are not currently on screen, or that entering a blank and returning to the main menu is a useful way of extracting oneself from problems.

These two types of knowledge do not exist independently of each other but rather are complimentary and interwoven. The latter type of knowledge seems to emerge from the associative process due to the innate tendency of the cognitively active user to attempt to extract meaning from the computer's responses, relying on the information given by the interface and his own experience and knowledge to make sense of the interactive process. In so doing such knowledge provides a framework and selection rules for the application of the more specific associative knowledge. In this way, users of varying levels of proficiency possess a mixture of both knowledge types.

Obviously where an interface provides the user with clear and precise information the process of acquiring the relevant knowledge needed for successful interaction is enhanced. Therefore it is of little surprise that users of the "friendly" interface with its explicit feedback and user-oriented language, are better able to aquire suitable knowledge. Not only does the explicit feedback and clear language provide a better means for associating input with output leading to faster associative knowledge, but the extra information available from the interface allows the user to extract more meaning from the activity and thus acquire a more accurate mental model of the system.

However, users of the "unfriendly" interface do manage to interact with their system. Indeed behaviourally, they are seen to be virtually indistinguishable from users of the "friendly" computer. Obviously therefore these users can make inferences and draw conceptual knowledge from the informationally poor interface. The protocol analysis though, suggests that the similarities in knowledge between these two user groups are superficial. While both groups may appear to be of equal proficiency their respective level of understanding of the system and its operation differ significantly. This is not surprising since the users of the "unfriendly" interface must rely to a greater extent on their own experience and limited knowledge to "fill the gaps" in their model of the system.

It seems likely therefore that a distinction can be drawn between what I will term here "shallow" and "deep" knowledge. "Shallow" knowledge users possess numerous associative knowledge types that allow them to interact routinely with a system by virtue of their memory of what type of input is required for a particular prompt and in what order task-specific sequences of prompts occur. Such users possess limited knowledge of overall system operation by virtue of their poor or inaccurate conceptual model of the machine.

"Deep" knowledge users on the other hand while possessing a similar if not larger amount of associative knowledge fragments, have a fuller knowledge of the overall system. Such users rely less on the specific prompt of the interface for guidance on interaction and rely on a detailed mental model of the system to guide them.

It seems likely that users who possess "shallow" knowledge would find routine interactions on a regular basis quite simple yet be unable to use such knowledge to tranfer to alternative systems or handle unfamiliar interactions on the same interface. This is natural since such associative knowledge does not afford transfer to alternative domains. Hence the phenomenon of numerous casual users of office equipment who to the unitiated novice or brief observer appear to interact effortlessly and knowledgeably, when in fact they possesses no genuine understanding of the computer and are unable to explain to others what to do (except for the verbatim disclosure of input/prompt sequences).

Users who possess "deep" knowledge of the should be able to handle unfamiliar interactions by relying on their well developed conceptual model of the system and its overall manner of operation to guide their interactions. Rather than responding to the "NAME:" prompt as simply a request for input for example, a "deeply" knowledgeable user would perceive that they were in fact in a particular mode of operation that facilitated only a limited type of possible interactions.

Conclusion

These results suggest that we may meaningfully view the user of an interactive system as a psychological entity, perceiving, processing and responding to information. Naive users approach a system with a variety of cognitive capabilities to employ in their interaction and, given the familiarity of the concept of the computer in contemporary life, a rudimentary (if innacurate) understanding or mental model of how such "things" work.

User knowledge seems to be acquired through the acquisition of associative and conceptual knowledge. A users ability to form such knowledge is strongly influenced by the interface as the provider of information that the user requires. For novice users it seem that clear, explicit information best facilitates the acquisition of user knowledge, though in the absence of this, users may still manage to interact "successfully".

However, the distinction between "deep" and "shallow" knowledge would suggest that studies of interaction and interface evaluation that only take behavioral measures of performance into account must be interpreted with caution.

By viewing the user as an information processor and, by extension, the interface as an information provider, it can be seen why interface requirements differ for users of varying tasks and competencies. The information provision an expert word-processor user requires is different from the information a casual spreadsheet user requires.

These findings have implications for the design of usable interfaces. Hansen's (1971) call for us to "know the user" has as much relevance now as it had then, though I would modify it now to "know the user as an information processor". If the end user can be understood in terms of the language s/he employs, the type of information s/he uses, the regularity of his/her interactions, and the feedback s/he requires, the design of an interface to suit that user becomes easier.

Knowledge of this variety can provide the general framework or context within which the more specific details of interface design that have so far occupied the efforts of many researchers may be placed.

Bibliography

1. Barnard, P. and Hammond, N. (1982) Usability and its multiple determination for the occasional user of interactive systems. Hursley Human Factors Report HF059, IBM UK Laboratories.

2. Barnard, P., Hammond, N., Maclean, A. and Morton, J. (1982) Learning and remembering interactive commands in a text-editing task. Behaviour and Information Technology, 1, 4, 347-358.

3. Carroll, J.M. and Mack, R.L. (1984) Metaphor, computing systems, and active learning. International Journal of Man-Machine Studies, 22, 1, 39-58.

4. Dillon, A.P., (1986) User-Friendliness: Psychological Aspects of the Usability of Computers. Unpublished M.A. Thesis, University College Cork, Eire.

5. Goodman, T. and Spence, R. (1978) The effect of computer system response time on interactive computer aided problem solving. ACM SIGGRAPH 1978 Conference Proceedings, 101-104.

6. Hammond, N., Morton, J., and Barnard, P. (1982) Knowledge fragments and user's models of systems, in: Cognitive Engineering: a conference on the psychology of problem solving with computers. Amsterdam 1982, 13-14.

7. Hansen, W.J. (1971) cited in Shneiderman, B., Software Psychology: Human Factors in Computer and Information Systems. Winthrop, Cambridge, 1980.

8. Norman, D.A. Some observations on mental models, in: D. Gentner and A. Stevens (eds.) Mental Models, Lawrence Erlbaum Assoc. Hillsdale, 1981.

9. Rosson, M.B. (1984) The role of experience in text-editing. In B. Shackel (ed.) Proceedings of Interact '84. First IFIP Conference on HCI. London IEE.

10. Savage, R.E., Habinek, J.K. and Barnhart, T.W. (1982) The design, simulation and evaluation of a menu driven user interface. Proc. of the Conference on Human Factors in Computer Systems, Gaithersburg, MD. March 1982, 36-40.

11. Shneiderman, B. Designing the User Interface:Strategies for Effective Human-Computer Interaction. Addison-Wesley Pub. Reading MA. 1987.

12. Shneiderman, B. System message design: guidelines and experimental results. In: A. Badre and B.Shneiderman (eds.) Directions in Human Computer Interaction. Ablex, Norwood, N.J. 1982.

13. Snowberry, K., Parkinson, S.R., and Sisson, N., (1983) Computer display menus. Ergonomics, 26, 699-712.

14. Stevens, G.C. (1983) User Friendly Computer Systems?: a critical examination of the concept. Behaviour and Information Technology, 2,1, 3-16.

15. Wilson, M.D., Barnard, P.J. and Maclean, A. (1985) Analysing the learning of command sequences in a menu system. In S.Cook and P.Johnson, People and Computers: Designing the Interface. Cambridge, 1985.

Acknowledgement

The present work was carried out while the author was a member of the Human Factors Research Group at the Dept. of Applied Psychology, University College Cork, Eire.