EXPERIMENTS ON LEARNING INTERACTIVE DIALOGUES:

PROBLEMS AND PROSPECTS

P.J. Barnard

Rapporteurs: Mr. D.R. Brownbridge
Mrs. J.A. Anyanwu

Abstract:

With the emphasis now on the development of "user-friendly" interfaces, system designers are increasingly turning to human factors practitioners and psychologists to provide advice on the usability of software products. For some aspects of system design, for example, keyboards and displays, a wealth of human factors information is available. Much of this information is derived from behavioural research on users' perceptual and motor skills. For other critical aspects, such as dialogue design and user understanding of system concepts, human factors information is less readily available. For these aspects much depends on users' cognitive skills in understanding, knowledge use, memory, communications and problem-solving. Research on the way these skills come into play in the course of human-computer interaction is in its infancy and this lecture will concentrate on evaluating the progress being made in behavioural experiments on the learning of interactive dialogues by non-expert users.

The general aim of such research is to accumulate evidence and formulate principles concerning user-system behaviour. As with other new areas of research in the behavioural sciences, there are many purely methodological problems to be solved. These concern the definition of experimental treatments, the measurement of behaviour and the interpretation of results. Difficulties of this sort can lead to similar experiments having different apparent implications for system design. A significant part of the problem is that the processes of human cognition are both complex and flexible. The precise course of a particular user-system exchange is dependent upon subtle interplays between the structure and content of the dialogue, user knowledge, the task, and the demands it imposes on the users' understanding and memory. In consequence, conclusions based on a single piece of evidence may not generalise from one context to another. Evidence illustrative of these points will be drawn from studies of command- and menu-driven dialogues. In order to integrate evidence and draw conclusions at an appropriate level for application, we need to develop our theoretical understanding of user cognition.

Experiments on Learning Interactive Dialogues: Problems and Prospects

Philip Barnard
MRC Applied Psychology Unit,
Cambridge

With an increasing commercial emphasis on the design and development of "user-friendly" interfaces, system designers are increasingly looking to human factors practitioners and psychologists to provide them with information, tools and advice about the usability of their products. For some aspects of system design, for example keyboards and displays a wealth of human factors information is available (eg. see Rupp, 1981). For other critical aspects such as dialogue design and user understanding of system concepts, human factors information is less readily available. For these aspects, usability will depend upon the compatibility between the properties of the system and the user's cognitive skills in understanding, knowledge use, memory, communication and problem solving.

Although research on the way in which cognitive skills come into play in the course of human-computer interaction is very much a developing rather than mature tradition, behavioural scientists are currently responding with some vigour to the issues raised and problems posed. A large range of approaches, techniques and tools are being explored both for the purposes of immediate application and for the "longer term" accumulation of knowledge and principles concerning user behaviour. Of these approaches, systematic behavioural experimentation is one means of exploring ideas concerning the factors which may underlie usability. What are the prospects of such longer term experimentation leading to applicable insights into usability?

Problems of interpreting evidence

As with other new areas of research in the behavioural and cognitive sciences there are many purely methodological problems to be tackled and solved. These concern the definition of experimental tasks and treatments, the measurement of behaviour and the interpretation of results. Difficulties of this sort can lead to experiments concerned with similar variables having different apparent implications for system design. The point can be illustrated by reference to three studies of the usability of different kinds of terminology for command dialogues in text-editing. The studies were all concerned in one way or another with the "naturalness" of the terminology for end-users. Making dialogues "more natural" is one potential route for enhancing usability.

In one of the studies (Ledgard et al, 1980), the CDC NOS notational text-editor was compared, in an on-line test, with a more "English-like" equivalent in which abstract symbols were replaced by familiar descriptive words and phrase structures. On both performance and preference measures the "English-like" editing dialogue was favoured. In another study (Scapin, 1981) an off-line memory paradigm for command names and their definitions was used to compare typical system terminology with terminology derived from the textual domain. For naive users, memory for command names and their definitions was found to be better when computer-oriented command names were employed. In third study (Landauer et al, 1983), a cut down version of the UNIX text editor ED was employed with three different vocabularies - the original UNIX names, user-nominated command names, and words whose meaning was totally unrelated to the underlying command actions. These investigators were unable to detect any reliable effect on interactive performance of the different types of names.

To system designers, who are presumably "naive users" of behavioural data, such findings may understandably appear perplexing. To the "expert user" of cognitive experiments they form three pieces of a very large behavioural jigsaw puzzle. In technical terms, it is unfair to compare these experiments and their results in this way. The studies actually made different manipulations, with different command set sizes and user populations as well as employing different measurement techniques. Such technical niceties are, however, unlikely to impress a practical system designer who wants to know how to provide names for the functions implemented in a new system. The important point is that it is not very helpful to define and discuss the usability of namesets in terms of a simple concept such as naturalness. There are many different factors involved which will ultimately contribute to the way in which the user copes with the problems of learning to attach names to functions. The research requirement is to untangle the relationships between factors such as the memory load imposed by different sizes of command set , the semantic interconfusability of lexical items; and how such factors are influenced by task variables and differences between types of users.

There are unlikely to be simple answers to these questions. Many of the factors involved will be interdependent and an effective design solution for one context of use may not be the solution required by another context of use. It is important that we attempt to characterise our answers to the questions in a generalisable manner. If we don't, we will only ever have insights and answers about yesterday's technology and software. This will require a conceptual synthesis not simple adherence to the implications of isolated pieces of evidence. The prospects for a synthesis based solely on the formal properties of dialogue are poor. The same kinds of dialogue structures and vocabularies have different effects in different contexts of use. In order to understand why this should be, we must consider in more detail the precise cognitive demands imposed by the style, structure and content of dialogues used in the different contexts of use.

In the case of learning command names, users are able to recruit their knowledge of natural language terminology to resolve any ambiguities in the mapping between command names and their underlying functions. However, if there is an ambiguous mapping between names and the underlying functions, a cognitive demand is imposed to resolve it. This demand may result in users adopting different strategies for learning with consequential effects on the precise pattern of their system use and for their subsequent ability to recall relevant information. In one on-line experiment (Barnard et al 1982), we studied two types of command names in a text editing environment. Users coped with a set of semantically general command names (eg ADD & EDIT) by consulting HELP facilities more often than they did with a set of semantically specific names (eg INSERT & DELETE) . The latter kind of name furnished clues which enabled them to resolve an appropriate command entry without extensive recourse to the HELP facilities. In an unpublished follow-up study, command sets were studied in which specific and general names occured WITHIN the same set. For command namesets constituted in this way, naive users were able to cope with the general command names without additional recourse to the HELP facilities. Here, the critical factor concerns, not the abstract property of the individual names, but rather on the total cognitive demands imposed by the particular set. Perhaps significant performance decrements occur only when these demands exceed the user's effective representational capabilities.

Task and context dependent effects

Similar points can be made in relation to the demands of learning, using and remembering aspects of the structure of alternative forms of dialogue. For example, in two on-line studies, again carried out in our laboratory, factors associated with the structure of command-argument sequences were examined (Barnard et al, 1981; Hammond et al, 1980). In the first of these experiments users were required to use commands like Delete <arg1><arg2>, where the arguments were numeric and referred to components of a "secret" message which they were decoding. Several commands were required to complete the decoding of each message. One of the arguments always referred to the message "number" and was thus recurrent. The identity of the other argument varied from command to command. The user's task was essentially very simple. Since we were interested in the learning of verb-argument structures, we designed the "experimental" system so that it prompted the user as to which command to use. The essential cognitive demand was to learn and remember the structure of the argument sequences. We compared several rules for positionally formatting the arguments. In two conditions we held the position of the recurrent argument constant, irrespective of its "natural language" relation to the command verb. In another two conditions we placed the arguments either with the direct object of the command verb first (the usual natural language order) or with the effective indirect object first ("unnatural"). When the recurrent argument was held in a constant position, performance was generally better than in the conditions where the position of recurrent argument varied, but the advantage was largely confined to the case where the recurrent argument came first.

In the second experiment we examined the generality of this finding in a more complex task. We expanded the command set and removed the prompts as to which command operation to perform next. We also included two different recurrent arguments in place of the single recurrent argument of the earlier experiment. Finally, we compared two different command vocabularies - a set of semantically general names and a set of semantically specific names. The demands of this task were more like those of a "real" system in that users really did have to understand and remember the full command strings. Under conditions where command selection was enforced the advantage of placing the recurrent argument first rapidly diminished. This held for the specific command vocabulary. With the general command vocabulary we could detect no advantage at all of placing the recurrent argument first. As with the command name experiment, the effects of the superficial properties of the dialogue were context and task dependent.

Task and context dependency are not only obtained with command-driven styles of transaction. We have obtained very similar kinds of effect with menu driven dialogues. In these experiments (see Barnard & Hammond, 1982), we asked naive users to use a data-base system which involved selecting functions (eg DISPLAY) and objects (eg FILE) from a menu. Their task was to answer questions like "Display the file for agent Aquarius. Does it contain a list called travel?". In the first version of the task they were free to choose the dialogue constituents (functions & objects) in any order. We manipulated the structure of the menu (functions above objects or vice versa) and the structure of the question posed. These had the effect of altering the superficial way in which the users encountered task relevant information. We also manipulated the information different groups of users were given about the functions and objects. The initial experimental briefing either included explicit definitions of the functions, but not the objects or vice versa. This manipulation is somewhat similar to the use of general and specific command names in that it should influence the users' initial knowledge of operations or objects. Each of these factors biassed the order in which users tended to select functions and objects. There was an overall bias towards selecting functions first, but this was modulated by the briefing, the structure of the menu, and the structure of the question.

In a second version of the task we imposed a fixed order of menu selection. Our users were first shown a menu of functions, then a menu of objects or vice versa. Under these task conditions the order of selection, the structure of the question and the type of experimental briefing concerning the system functions and objects had little effect on user performance. Again, a change in the task demands modified the detailed characteristics of user system exchanges. Furthermore, comparison of the two task conditions - fixed and free order of selection yielded a somewhat counterintuitive finding. It might be expected that users would find it "easier" to learn their task given freedom to select the function and object constituents in any order. In fact the users made more use of the help facility under these conditions than they did under the fixed order of selection. Additionally, they again made more extensive use of the Help facility when more complicated two-transaction questions were posed in the last few trials of the experiment.

The formulation of behavioural principles

In each of the illustrations given above, the effects of specific properties of command structures and command namesets were complex. The effects were dependent upon the kind of task the user was required to perform and upon the specific cognitive demands of the system context in which the user was attempting to formulate and execute a particular dialogue transaction. From the perspective of application and design, this should not be taken to mean that evidence from behavioural studies "always tells a different story". Similarly, from a research perspective, it should obviously not be taken to mean that user behaviour is unprincipled or unsystematic. The clear implication seems to be that it will be difficult, if not impossible, to formulate meaningful "principles" by referring only to the superficial and logical properties of the different types of dialogues. If not, how are we to formulate our general principles which will enable us to make inferences and take decisions about what properties to include in enhanced or completely novel forms of dialogue? One potential solution is to base the formulation of our principles around an analysis of the cognitive demands likely to be imposed by a user interface and our knowledge of how users' cognitive skills actually respond to those demands.

For example, to formulate our principles concerning the learning of argument structures, we need to consider the kinds of mental representations which the users may be acquiring and accessing. Consider the case of the experiments concerning command argument structures. When the command operations were prompted, users are required to recognise the command name but to recall the argument structure. The command name is the accessing cue for the argument structure. Under these circumstances users may build very simple mental representations focussing on the superficial structure of the command string and its associated argument identities:

```
 COMMAND NAME: ===> <argl> <arg2>
  (recognition)       (response)
```

When command selection is enforced or when a semantically non-specific command vocabulary is employed, the user may not be relying on the retrieval of a simple structural representation. Instead, they may be more likely to generate a mental representation of the conditions under which a particular function is to be used, enabling active retrieval of the both the command name and its required arguments from memory via a more differentiated semantic representation of the underlying function:

```
 COMMAND FUNCTION: ===> name <args>
        |      |
        |      |
        |      |
        v      v
   ARG1   ARG2
```

We may also be able to formulate underlying principles to help us understand why the fixed order of menu selection, in that particular context, was apparently "easier" than the free order of selection. Again we must consider the detailed representational and memory demands. For the fixed order of selection of the verb and object constituents within a transaction, we could suppose that users implicitly come to represent three rules:

FIXED ORDER OF SELECTION

```
  Rule A  Required Action is: <Selection 1><Selection 2>
  Rule B  Selection 1 is: <a function>
  Rule C  Selection 2 is: <an object>
```

Since the menu order is system-driven, there is no explicit demand for the user to have to "retrieve" or decide upon the order of constituents within a transaction. Furthermore, the users experience of the transactions, which we assume are also represented in their own memory "records", would be structurally consistent:

```
  Memory      <Display><File>
              <Compare><List>
              <Delete><File>
                  .
                  .
                  .
```

In constrast, with the free order of memu selection, the rules and users' memory "records" would look rather different on this kind of analysis:

FREE ORDER OF SELECTION

```
  Rule A  Required Action is: <Selection A> & <Selection B>
  Rule B  Selection A is: <Function> OR <Object>
  Rule C  Selection B is: <"not the last value of selection A">
```

For the actual performance of these transactions under conditions of free order of selection, the evidence indicated that the actual order selected was biassed by the initial briefing and by the "surface structure" of task relevant information displayed on the terminal. The actual order of entering constituents varied from one instance to the next.

Accordingly, the users own memory "records" for the individual transactions would be structurally inconsistent:

```
Memory    <Display><File>
          <List><Compare>
          <Delete><File>
          .
          .
```

In effect, cognitive representations involved in the control of the free order transactions would be functionally more complex and the memory "records" which users might access in formulating a course of action would be inconsistent. Of course, for some kinds of dialogues, a cognitively rich representation of this sort may be required to support the user's problem-solving activities. However, when the user's task is relatively straightforward, the richer representation and structural inconsistencies in their memory "records", may give rise to more initial problems in learning and with transfer from simple to complex questions; as was the case in our particular experiment.

Fragmentary ideas of this sort obviously require validation and considerable extension. Nevertheless, they do serve the function of illustrating my own conviction that the principles required for human computer interaction require a sound basis in cognitive theory and task analysis. At present such ideas may serve the rather limited function of altering computer designer's attitudes towards the analysis of usability. However, as the concepts develop in both number and complexity, they may themselves become increasingly "user-unfriendly" to software designers. Just as naive computer users should not need to know about the technicalities of the internal software, software specialists should not necessarily have to wrestle with the detailed concepts and technicalities of the behavioural sciences. The ideas and techniques will thus require translation into usable tools. Some of the issues and problems associated with the translation process will be taken up in my second lecture.

References:

Barnard, P. & Hammond, N. (1982) Usability and its multiple determination for the occasional user of interactive systems. In M.B. Williams (Ed), Pathways to the Information Society: Proceedings of the Sixth International Conference on Computer Communication, 543-548.

Barnard, P., Hammond, N., MacLean, A. & Morton, J. (1982) Learning and Remembering Interactive Commands in a Text-Editing Task. Behaviour & Information Technology. Vol 1, 347-358.

Barnard, P., Hammond, N., Morton, J., Long, J & Clark, I. (1981) Consistency and compatibility in human-computer dialogue. International Journal of Man-Machine Studies, vol 15, 87-134.

Landauer, T., Galotti, K. & Hartwell, S. (1983). Natural command names and initial learning: A study of text-editing terms. Communications of the Association for Computing Machinery, vol 26, 495-503.

Ledgard, H., Whiteside. J., Singer, A., & Seymour, W. (1980). The natural language of interactive systems. Communications of the Association for Computing Machinery, vol 23, 139-150.

Rupp, B. (1981) Visual Display Standards: A review of issues. Proceedings of the SID, Vol 22/1, 63-72.

Scapin, D. (1980) Computer Commands in Restricted Natural Language: Some aspects of Memory and Experience. Human Factors. vol 23, 365-375.