Experimental adaptive interface

M V Mason and R C Thomas

Software often has to interact with end users. It is thus of interest to software designers to consider the incorporation of useful man–machine interface techniques into their systems. The present article seeks to transfer knowledge of adaptable interfaces to the information retrieval community. An experiment at Leeds University into adaptive interfaces for the online UNIX manual is described, together with an overview of some of the design issues. The relevance of dialogue monitoring as a source of data for determining the model of user behaviour is discussed as is the design of dialogue generating programs that can modify their output to suit the user model.*

Keywords: computer software, man–machine interface, information retrieval

INTRODUCTION

For any human–computer system to enjoy success, the design of an effective man–machine interface (MMI) is an essential prerequisite. A wide range of information systems including expert systems, database systems and electronic publishing systems need to make use of appropriate interface engineering principles in order to produce a system that is acceptable to users.

An example where considerable effort has been spent on interface design for database applications is the ROBOT system[1], which features a natural language query interface. The BCS Query Languages Group[2] also discusses techniques for effective query language design. Many of these reported techniques are standard MMI approaches, for example menu selection, which are widely used in non-database systems[3].

Given the need in all information systems, including databases, to engineer high-quality interfaces and that to do so requires effective use of MMI design principles, this article seeks to introduce to the information retrieval community some techniques concerned with the design of adaptable/adaptive interfaces. Design issues relevant to the construction of this form of interface will be discussed in relation to our experiments on the provision of an adaptive interface to the online UNIX manual. Prior to the discussion, however, we briefly review work on electronic reference book systems as it provides the context for our experiment.

Department of Computer Studies, University of Leeds, UK

*UNIX is a trademark of Bell Laboratories

ELECTRONIC REFERENCE BOOK SYSTEMS

There is some discussion in the literature concerned with the implementation of electronic books. Opportunities in this area range from the simulation of paper books in electronic form[4] to the exploitation of the computer-based media to manipulate book information in hitherto impossible ways[5].

We are particularly concerned with the discovery of general engineering principles required in the design of electronic reference books. In particular, our research efforts concentrate on examination of the applicability of the electronic medium to online software manuals.

Authors such as Sohr[6] and Maynard[7] point out that for many software systems, paper reference manuals ideally need to meet a variety of differing information needs (e.g., some users require reference information only, whilst others require detailed help information). Invariably, an attempt to satisfy all user needs fails — in essence paper software manuals cannot be 'all things to all men'. Preliminary investigations into the use of the online UNIX manual at Leeds University suggested that this basic problem of paper software manuals also existed in that electronic manual (see section on monitoring data).

What is ideally required is a manual that can alter its characteristics to suit the particular needs of users consulting it. We refer to such a manual as an adaptable manual.

 0144–817X/84/030162–06$03.00 © 1984 Butterworth & Co (Publishers) Ltd

It is proposed that the construction of a fully adaptable manual will need to incorporate mechanisms which:

- automatically adapt information search methods,
- automatically tailor knowledge in the book according to specific information needs (e.g., the level of detail in displays of reference information changes to suit users with widely different levels of technical knowledge),
- permit users to organize the layout of information displayed on the screen as they require.

The technology of adaptable/adaptive interfaces provides some of the tools needed to build an adaptable manual, but there are other research issues such as appropriate knowledge representation formalisms and text generation mechanisms that lie outside the scope of the current project.

We now examine some of the key concepts in man–machine interface work that have proved useful in the project.

AN OVERVIEW OF ADAPTATION AT THE INTERFACE

A central concept to the study of the man–machine interface is that of the user model. In this context, the term 'user model' refers not to some scheme for user categorization, e.g., casual or dedicated user, but rather to the user's conceptualization or personal image of the system with which he is interacting[8].

Authors such as Nickerson[9] and Young[10] point out that well formulated user models, where the user has a clear conceptualization of the interactive system, positively assist him during interaction, and that conversely he is actually hindered by poorly formulated models. Indeed, Moran[11] believes that during the interface design process the system designer should construct a conceptual model of the system for the user to learn. The interface should possess features such as an adequate degree of interactive feedback that will allow the user to assimilate an appropriate conceptual model of the system as easily and as effortlessly as possible.

Sometimes a poor user model can have disastrous consequences as exemplified by Weizenbaum's[12] famous program ELIZA, which was capable of simulating the type of dialogue that might occur between patient and psychiatrist using keyword-based rules to transform input sentences. The resulting dialogue was so convincing that, despite the fact that ELIZA did not attempt to 'understand' user responses, many users conceptualized the system as being intelligent and capable of appreciating their feelings.

The user's conceptualization of a computer system is not static but changes, for example, as the user gains greater experience of the system. Furthermore, many systems have a number of users, each with varying degrees of knowledge, experience and requirements and thus with different user models.

For the purposes of this paper, we wish to distinguish three approaches to adaptation at the interface:

- Adaptable interfaces which provide programmable facilities so that the user can alter the interface to suit himself.
- Adaptive (or self-adaptive) interfaces which are capable of changing themselves automatically in an appropriate fashion. Typically they need to formulate some model of the user which might simply consist of a descriptive set of variables which is sufficient to discriminate between individual users or user types. In a more complicated approach, which is suitable for dialogues with relatively few interactions between user and computer, Rich[13, 14] has built a user-modelling system that embodies stereotypical knowledge concerning human characteristics.
- Adaptive dialogue design in the sense that the interface evolves gradually via the production of a number of system prototypes. For example, when Alsberg *et al.*[15] built a database interface driven by a microcomputer and touch-sensitive display, they found that, even though the system was easy to use, there were some subtle requirements that simply could not be predicted by the designer.

OVERVIEW OF THE CURRENT EXPERIMENT

In this section we describe the major features of our system. Justification for our approach can be gained from examination of the issues presented in the literature review above.

Current system

The information in the existing online Unix manual is organized into eight major sections. For example, section one contains information on the general commands available to users, while section eight has system maintenance commands. The manual information is retrieved when a user invokes the Unix *man* command. So, for example,

```
man pc
```

retrieves the manual article on the pascal compiler called 'pc'. The user may specify an appropriate section number if desired. Thus

```
man 5 termcap
```

retrieves the manual article on the system's terminal capability database (termcap) in section 5, which is denoted by termcap(5). Note though that

```
man termcap
```

retrieves the termcap(3) article. If a section number is not specified, then the manual searches progressively through its sections in order.

Consistent with the Unix philosophy of modifying a command's behaviour by the setting of flags, *man* offers a number of options. For example, the *-k* flag performs a simple keyword search so if the user types

```
man -k pascal compiler
```

then references to all manual articles with the keywords pascal or compiler in the name field will be listed. The implicit inclusive OR has the drawback that references to all compilers are listed.

Other flags enable the user to obtain brief details of command usage and one line descriptions of system commands.

The current interface can be criticized for:

- its lack of cues to help the user conceptualize the structure of the manual as an electronic book with sections,

- its lack of indexing features,
- its poor keyword search,
- its rather incomprehensible article style,
- poor facilities for browsing through a selected article, sometimes called page flipping.

First experimental version

Features

The philosophy of the system prototyping approach to design has been used so that our basic design strategy for the first version has been to implement a small, but one hopes useful, set of features with the intention that this will motivate users into suggesting new facilities that they would find useful. There is some evidence that this is starting to happen.

The first version of the pilot adaptable manual is upward compatible from the old manual in that the same features (such as keyword search) are still available. However, the new interface has been designed to assist the user in conceptualizing the system as an electronic book as well as possessing new features, which include:

- Page flipping. Users can flip backwards and forwards within and between articles.
- Command prompting. When a normal article is displayed, the terminal screen is divided into two windows. The top window displays one page of manual text whilst the bottom window displays a set of command prompts. As the user's experience of the system increases (as judged by the user-modelling mechanism) the command prompt window decreases in size (allowing more manual text to be displayed). In addition, the nature of the prompts changes such that prompts for sophisticated commands are displayed to the more experienced users.
- Indexes. An index for each section of the manual has been provided as another means of locating relevant information. An adaptive dialogue assists the user in searching these indexes. It includes explanation of the screen layout and terms, a prompt window, and the ability to use a menu or flag driven dialogue.
- Glossary. If the user does not know the meaning of a phrase in the window, he can request an explanation from the built-in glossary by simply placing the cursor on the word and pressing 'g'. Any requests which cannot be answered are stored so that the relevant glossary entry can be added later.
- Cross referencing. While reading an article, the user can ask for any manual information on material by placing the cursor on the word and pressing 'm'. If a suitable article exists, it will be displayed once the user has finished reading the current item.
- Help. While viewing a manual article a user can request information without needing entirely to abandon the currently displayed article (unlike in the existing manual). The help information given depends on the system's model of the user, i.e., the help system dialogue is adaptive. There may be some advantage in presenting help information on the use of the adaptable manual in the same way as information on UNIX is presented by the manual. This gives a recursive flavour to the electronic book system. This 'recursive architecture' could conceivably map naturally into systems providing many levels of help information.

Provision of adaptation

The manual's adaptive processes are driven by a simple model of the user that attempts to quantify the user's experience at using the system. Currently such user modelling is based on a combination of rules and weighting of a set of user-descriptive variables.

For example, to select the appropriate article display, users are graded into classes from TYPE1 through to TYPE4. The TYPE1 user is given explanation material and substantial prompting from the system. On the other hand, the TYPE4 user is immediately presented with the text and given only the prompt 'For help type h'. The user modeller initially classifies the user according to a set of hypotheses. An example of such an hypothesis is:

Hypothesis: 'The TYPE2 user should be upgraded to TYPE3'

V1: the user has used the system more than twice ($w = 0.25$)
V2: commands x and y used in last retrieval ($w = 0.20$)
V3: help not used more than once in last retrieval ($w = 0.25$)
V4: the user has used the system within 5 days ($w = 0.30$)

Variables Vi have values of either 1 or -1 depending on whether or not the condition is true or false. A weighted result is calculated (in this case the sum of the wi times Vi) and tested against some prespecified threshold figure. If the result is greater than or equal to the threshold then the user is upgraded; if the result is less than the threshold then the user classification remains unaltered.

Monitoring data

This section gives a brief comparison of the data obtained from monitoring the usage of the standard manual (hereafter referred to as the 'standard manual') with that obtained from monitoring the adaptable manual. In both cases two records were made:

- 'Log of Manual Transactions'. Each transaction record comprised three fields:
  - user name,
  - time at which the command was invoked,
  - command arguments.
- 'Accounting Information'. A log of all system commands permitting the context in which the manual was invoked to be analysed.

Overall manual usage

The usage of the standard manual was recorded over a five week period (i.e. 25 working days) in the first term of the 1983–1984 session. During this period there were 1709 manual consultations with a total of 85 different users making use of the system. The usage of the adaptable manual was also recorded over a five week period in the second term of the session. In this case 3329 consultations took place and the system was used by 134 users.

In terms of the proportion of the UNIX user population that made use of the man command, the mean daily percentage use of the standard manual was 30.9 per cent (standard deviation: 2.9 per cent) whilst the corresponding statistic for the adaptable manual was 40.6 per cent (standard deviation: 6.4 per cent).

These results should be interpreted with great caution. Before the adaptable manual was installed, there were two similar commands: man and help. The latter had two functions: a simple two page help message, and also as an entry into the manual. Thus some manual usage may have been disguised as help usage. When the adaptable manual was installed, the help command was removed. The figures for help were mean of 17.5 per cent (standard deviation: 4.6 per cent).

Unsuccessful transactions

In assessing any interactive system one typical area of interest is the frequency of unsuccessful transactions. These occur either because of a user syntax error or because, even though the syntax is correct, the requested information is not in the database.

For example, consider the command line:

$ man pascal compiler

The standard manual simply informs the user that the article 'pascal compiler' does not exist and then quits. The adaptable manual, on the other hand, automatically invokes the keyword search facility prompting the user with the line:

pc (1) — pascal compiler

The mean daily percentage of unsuccessful transactions from the standard manual was 12.9 per cent (standard deviation: 6.2 per cent) and for the adaptable manual was 13.5 per cent (standard deviation: 4.5 per cent). The increase for the adaptable manual does not appear to be significant and similar types of unsuccessful transaction occurred in both systems. However, the adaptable manual does deal with unsuccessful transactions in a somewhat different fashion from the standard manual.

Search

A significant difference between the manual versions was that the number of transactions devoted to searching for articles increased in the adaptable manual. In the standard manual keyword searching only was available and the mean daily percentage of transactions concerned with this facility was 4.9 per cent (standard deviation: 4.5 per cent). For the adaptable manual, however, as previously stated, an index search facility was included. In this case the mean daily percentage of keyword search transactions was 3.4 per cent (standard deviation: 2.3 per cent) and the corresponding figure for index searching was 7.3 per cent (standard deviation: 3.3 per cent). Indeed, searching by index proved to be a quite popular facility.

Anecdotal evidence

User comments were recorded via the mail system and also noted from informal conversations. It became clear that users were generally not prepared to spend a great deal of time learning how to use the system (after all users consult the manual to facilitate the completion of some main computing task), but they did expect the system to possess many of the features that they had encountered in sophisticated text editors. Typical examples of suggested facilities were cursor motion by word, bookmarking facilities and within-text searching using regular expressions to specify patterns. This undoubtedly causes something of a paradox as users typically invest a substantial amount of time and effort in learning how to use text editors that possess a comparable set of functions. The problems caused by this paradox are of current research interest. However, although users were given no training nor in-depth documentation on how to use the existing adaptable manual there were few reported difficulties concerning system use. In this respect, it seems that the adaptive interface has made a significant contribution and this can be seen as a sign of encouragement in the design of a future adaptable manual system.

DESIGN ISSUES

This section provides an overview of some of the design issues found to be relevant to the adaptable manual system.

Dialogue monitoring

The recording of the interactive exchanges between user and system by dialogue-monitoring software has contributed significantly to the design of the adaptable manual. For accounts of approaches to and benefits from dialogue monitoring in other systems see Abrams *et al.*[16] and Gainer *et al.*[17].

A dialogue monitor can be used for:

- Obtaining information on the use of some existing system in a manner akin to the systems analysis technique of observation. Moreover, insight can be gained into whether or not the user population is homogeneous or heterogeneous in nature and hence indicate the justification for building an adaptive (in the heterogeneous case) or non-adaptive (in the homogeneous case) interface.
- Obtaining evaluation information useful in checking the design of the interface.
- Obtaining feedback information which is required if a system prototyping approach to design is being used.

Development methodology

The intention is that the adaptable manual should be self-adaptive, although the system is also being developed via a system-prototyping strategy (as discussed in the section on adaptation at the interface) in order to obtain smooth interaction at all levels of dialogue.

Dialogue-generating programs

Dialogue design typically involves the specification and implementation of a set of production rules (equivalent to the rewriting of formal grammars) or alternatively a finite state transition network that defines the dialogue.

Guest[18] has investigated the use of two dialogue-design tools at Leicester Polytechnic, one of which relies on a production rule input to generate an interactive dialogue, while the other requires a transition network representation of the interface. From this study there seems to be some evidence to suggest that the network formalism is a somewhat easier approach to interface design. Certainly, all

dialogues in the adaptable manual have been designed in terms of state transition networks.

The use of state transition networks for dialogue design have been reported by a number of researchers such as Parnas[19], Denert[20] and Edmonds[21]. Given certain slight notational differences between the network schemes recommended by different researchers, the state transition network is essentially a collection of arcs and nodes forming a directed graph. At each node, relevant output text is generated by the system. The corresponding response from the user causes the selection of an arc (which the designer has specified as being uniquely associated with the particular form of the input given by the user) emanating from the current node. Subsequently, an arc transition occurs and a fresh set of output text is displayed when the next node in the network is reached. This process is repeated so that the progression of the man–computer dialogue involves successive transitions from node to node in the network.

In the first version of the adaptable manual the interface to each facility is implemented as an individual dialogue-generating program (thus the complete system consists of a collection of such programs). The ability of programs to branch from one to another permits dialogue transition from facility to facility. At the highest level, therefore, the interface to the manual can be regarded conceptually as a virtual transition network with transition arcs connecting the dialogue-generating programs together. Indeed, the latter essentially form subnets of the top level virtual network. The representation of these subnets is explicit and has been implemented in the C programming language.

Currently, the manual facilities which generate adaptive dialogues possess their own individual user-modelling or classifying mechanisms. The attraction of this approach was seen as minimizing the degree of coupling between programs, and making each facility as highly cohesive as possible, as well as being a simple way of managing adaptation (a user may be expert at using one facility but a novice using another).

Given some particular manual facility with an adaptive interface, immediately prior to the user interacting with this facility, the user is inferred as belonging to some appropriate user class. Relevant arcs are then enabled or disabled on the transition network (representing the dialogue for the facility) in order to provide an interface appropriate for that user. During the subsequent interaction further adaptation may be required and this is achieved by the use of a number of rules that use knowledge of the current user class and the network topology to reclassify the user if appropriate.

Performance

The construction of an adaptive interface naturally requires additional processing. This overhead may affect the performance characteristics of the system.

For the adaptable manual system, the additional processing costs can possibly be reduced by restructuring the organization of the manual information. It is likely that some manual articles are more frequently retrieved by users than others. If this can be established from the dialogue-monitoring statistics, then the manual information storage scheme can be recognized to promote particularly speedy retrieval of popular articles.

The other major problem with performance is that the user may perceive the manual to be slow, when in fact it is doing the work of several transactions in the old version. Overall, therefore, it may not be as slow as users sometimes believe.

Terminals

We believe that the electronic book medium should be upward compatible from the paper book medium. One argument to support this is that the user's conceptualization of a system into a user model is partly based on previous experiences and prior ways of thinking. Therefore, it can be expected that there will be a tendency for electronic books to be conceptualized in terms of previous 'natural' (i.e., paper based) ways of reading texts in books. This basic book model provides a convenient foundation on which the engineering of computer-based books can be laid. It is envisaged that the technology of adaptable/adaptive interfaces allows the user's view of the electronic book to be gently modified (at a pace appropriate to the user) such that the user initially interacts with an electronic simulation of a paper book and is gradually revealed new text manipulation and information retrieval facilities that are realizable only in very powerful forms of electronic book.

Given that the initial user view of an electronic book is to be analogous to the paper book the interface should reinforce this conceptualization of the system. To this end, systems such as the ICL PERQ, which have high resolution displays capable of simulating sheets of paper, seem ideal for electronic book application, barring their physical size.

Our experiences in electronic book design also indicate that direct input devices such as the mouse or touch-sensitive keyboards as well as the ability to organize the display screen into a number of windows are also desirable features for this application area.

CONCLUSIONS

It has been pointed out that the system designer of any information system must take particular care in designing the interface. This, in turn, requires the effective utilization of well established MMI techniques.

This article has introduced one particular MMI approach which introduces the notion of adaptation at the man–computer interface. In relation to the task of retrieving information from an online software manual, this adaptive approach appears to have potential and provides opportunities for implementing an enhanced interface that is superior to the existing system.

Interface adaptation seems well suited to complex systems where users vary in their expertise at using the system. Indeed, a system that possesses a static interface initially requires the user to adapt to the rigid properties of the system and this places a considerable workload on the user (which is uncomfortable). Adaptation at the interface, however, allows the distribution of this workload between system and user (both system and user adapt to each other) and, as such, should promote more comfortable and smoother human–computer interaction.

REFERENCES

1 **Harris, L R** 'User oriented database query with the ROBOT natural language query system' *Int. J. Man–Mach. Stud. (GB)* Vol 9 (1977) p 697

2 **BCS Query Languages Group** *Query Languages — A Unified Approach* Heyden, UK (1981)

3 **Martin, J** *Design of Man–Computer Dialogues* Prentice-Hall, UK (1973)

4 **Benest, I D and Jones, G** Computer emulation of books *IEE Int. Conf. Man–Mach. Syst.* (July 1982)

5 **Weyer, S A** 'The design of a dynamic book for information search' *Int. J. Man–Mach. Stud. (GB)* Vol 17 (1982) p 87

6 **Sohr, D** 'Better software manuals' *BYTE* Vol 8 No 5 (1983) p 286

7 **Maynard, J** 'A user driven approach to better user manuals' *Computer* (Jan 1979) p 72

8 **Fitter, M** 'Towards more "natural" interactive systems, *Int. J. Man–Mach. Stud. (GB)* Vol 11 (1979) p 339

9 **Nickerson, R C** 'Why interactive computer systems are sometimes not used by people who might benefit from them' *Int. J. Man–Mach. Stud. (GB)* Vol 15 (1981) p 469

10 **Young, R M** 'The machine inside the machine: User's model of pocket calculators' *Int. J. Man–Mach. Stud. (GB)* Vol 15 (1981) p 51

11 **Moran, T P** 'The command language grammar: A representation for the interface of interactive computer systems' *Int. J. Man–Mach. Stud. (GB)* Vol 15 (1981) p 3

12 **Weizenbaum, J** 'ELIZA — A computer program for the study of the natural language communication between man and machine' *CACM* Vol 9 (1966) p 36

13 **Rich, E** 'Users are individuals: Individualizing user models' *Int. J. Man–Mach. Stud. (GB)* Vol 18 (1983) p 18

14 **Rich, E** 'User modelling via stereotypes' *Cognitive Science* Vol 3 (1979) p 329

15 **Alsberg, P A, Bailey, J F, Brown, D S and Mullen J R** 'Intelligent terminals as user agents' *Proc. IEEE Symp. on Trends and Applications: Mini and Micro Systems* (1976) pp 129–135

16 **Abrams, M D, Lindamood, G E and Pyke, T N** 'Measuring and modelling man–computer interaction *Proc. 1st Ann. SIGME Symp. on Measurement and Evaluation* (1973) p 136

17 **Gainer, E, Kimball, E and Kortesoja, A** 'A statistical evaluation of tactics usage' *Database* Vol 8 No 3 (1977) p 38

18 **Guest, S P** 'The use of software tools for dialogue design' *Int. J. Man–Mach. Stud. (GB)* Vol 16 (1982) p 263

19 **Parnas, D L** 'On the use of transition diagrams in the design of a user interface for an interactive computer system' *Proc. of the Nat. ACM Conf.* (1969) p 379

20 **Denert, E** 'Specification and design of dialogue systems with state diagrams' *Proc. 5th Int. Comput. Symp. (Liege)* (1977)

21 **Edmonds, E A** 'Adaptive man–computer interfaces' in *Computing Skills and The User Interface* **Coombs, M J and Alty J L (Eds)** Academic Press, UK (1981) p 389