# Experience with Adaptive Interfaces

D. BENYON* AND DIANNE MURRAY†

* *Faculty of Mathematics, Open University, Milton Keynes MK7 6AA*

† *National Physical Laboratory, Teddington, Middlesex W11 OLW*

***Over the last few years, adaptive interfaces have become a focus of research. In HCI one system is Monitor, which formed the basis of an adaptive interfaces project funded by the National Physical Laboratory. This paper describes our experiences in working on this project and the lessons that have been learnt about adaptive interfaces and the Monitor system. We then look at the experiments which have been conducted to establish the feasibility of modelling and using users cognitive characteristics to provide adaptivity. The paper concludes with some thoughts on the direction which future research should take.***

*Received June 1988*

## 1. INTRODUCTION

One view of human–computer interaction (adapted from Norman[6]) is shown in Figure 1. This highlights several important points. Firstly, the system consists of many tasks. For example a word-processing system includes the tasks print, index, spellcheck and edit. In turn edit includes enter, delete word, copy, move and so on. Secondly, there will be many (not necessarily simultaneous) users of the system. These are individuals who have differing skills and experience. Frequently users vary in their requirements of the system. For example, a manager and a secretary may use the same word-processing system, but they have very different needs and will typically use different functions. Thirdly, Fig. 1 emphasizes the difference between the human–computer *interaction* and the human-computer *interface*. Interaction includes all aspects of the environment such as the working practices, office layout, provision of help and guidance, and so on. The interface is the parts of the system with which the user comes into contact 'physically, perceptually or conceptually. Physical components include the use of the keyboard, type of screen and pointing device, perceptual features are concerned with the layout of the screen and use of colour or highlighting. Conceptual aspects relate to the users *mental model* of the system. Principally this is their understanding of the tasks and concepts which constitute the system. Notice that the variety of users typically share the same interface.

The final feature of Fig. 1 is the people who have an interest in the interaction, but who reside outside the system. These people typically include the system designer and system manager and may include other interested parties such as observers, psychologists and so on. These people all have different views of the interaction (because they are interested in different aspects of the interaction).

Users are often categorised into broad groups such as expert or novice users or frequent or infrequent users and HCI guidelines have been developed to assist designers in producing systems appropriate for these classes. However, we believe that whilst such categories are useful in focusing attention on the variety of interfaces required, such classifications are too broad. Users are individuals.

We also know that users change their behaviour from a primarily problem-solving activity to a routine skill[23] as their understanding of a system grows. Users learn about the system and gradually move from novice to expert user. In fact most users lie between novice and expert in a class of users called intermediate users. Moreover, this expertise relates to particular tasks which the system can perform and not to the system as a whole.

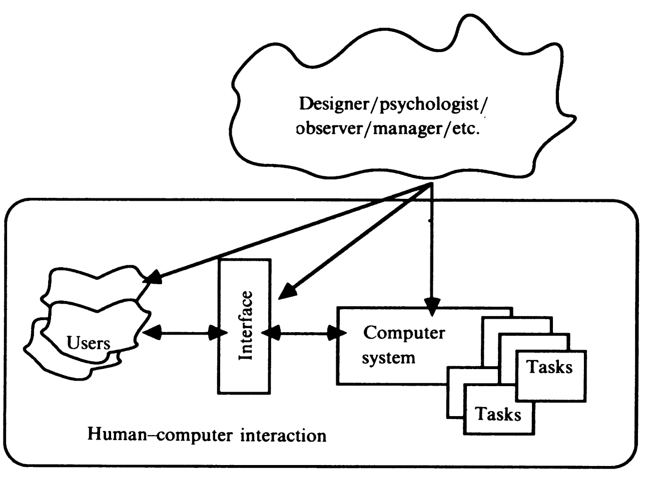


**Fig. 1. Model of human–computer interaction.**

Users differ in many other ways. Their previous knowledge affects the way in which they construct their mental model of the system, as does their goal in using the system.[6] Their mood influences how they perceive the system and their personality contributes to what they find agreeable or disagreeable about the system.

The problem which HCI designers are faced with is how to accommodate this variety and the changeability of the users. Any system potentially needs a variety of interfaces in order to provide a system powerful and flexible enough for the expert users yet simple enough for the novices. How can systems be designed to deal with the differences in background, ways of working and other personal preferences?

One solution is to build an adaptive capability into the application system, or into the interface system, which can then adjust the functionality of the system or facets of the interface to suit the individual.[1] Such adaptivity should benefit all users since the computer system being used will be tailored to individual requirements and skills. This is quite different from previous approaches to system design where dialogue and interaction were fixed and pre-specified and where the human had to adapt to the requirements and conventions of an inflexible computer system.

As may be expected, this is not straightforward. What aspects of the user should the system adapt to? How often should it adapt? Should the user be in control of the adaptativity or should it be automatic? Once these questions have been answered, how can the necessary knowledge be elicited and represented in the system? How effective is an adaptive system likely to be? Will it break other fundamental guidelines of HCI such as consistency of the interface? The remainder of this paper describes our experiences with these and related questions.

## 2. ADAPTIVE INTERFACES

Adaptive interfaces are a particular type of adaptive system. In another paper[2] Benyon discusses some of the lessons which can be learned from studying systems in general. Systems adapt to other systems in order to facilitate interaction between those systems. The following types of adaptivity can be identified.

* (automatic) adaptation
* customisation
* non adaptation
* evolution
* adaptability

Automatic adaptation is characterised by the system providing a response to another system. For example, thermostats automatically adapt to a change in temperature. Customisation requires action by one system to alter the state of another system. For example we can turn the volume up on a television set to adapt the system to our desired hearing level. Other systems are characterised by a lack of adaptivity. Perhaps the system has been designed for a particular purpose, but is unsuitable for other purposes. When two non-adaptive systems attempt to interact (for example a hammer and a window), the result is likely to be unsatisfactory at best (and may be destructive of one or both systems).

Evolution is concerned with adaptivity over time and the inheritance of preferred characteristics from one generation to the next. Biological systems are the clearest example, but the development of software systems is another. Adaptability is a measure of flexibility. An adaptable system can be used for many purposes, but it is not capable of changing those purposes. The hammer is adaptable, but it is not adaptive.

We are interested in adaptive systems, which change their state according to the requirements of the interaction and do not require (unlike customisation) the other system to actively participate in the changes. The main reason that we choose to ignore customisation is that in order to customise a system, the user must learn how the customisation system operates. This detracts from the central purpose of the system. It is the automatic adaptation of systems to 'the changing needs of users over time and to individual users or classes of users'[4] which we need. This distinction is important since many systems facilitate customisation but few, as yet, can provide knowledge-based and automatic adaptation.

There are, however, several levels and types of adaptation. Edmonds[3] describes five areas; user errors, user characteristics, user performance, user goals and the information environment. He suggests that adaptation is most easily accomplished automatically with reference to user errors. Adaptation to user goals and to the environment is difficult to achieve automatically, but is relatively straightforward through customisation. The actual manner and type of response is important here, however, as adaptation by the system to the environment is common at the physical interface level but not at the user's conceptual task level.

Totterdell, Norman and Browne[5] have examined adaptivity in natural and computer systems, and present the following taxonomy:

*Level 1 Systems* – produce a change in output in response to a change in input

*Level 2 Systems* – include an evaluation function which selects from a range of possible outputs.

*Level 3 Systems* – include a mechanism for monitoring the effect of the selected strategy on the environment and altering the evaluation function appropriately.

*Level 4 Systems* – possess an internal model of the environment in addition to an evaluation function and use the predictive capability of the model to select an appropriate response.

*Level 5 Systems* – inherit knowledge from previous generations and adapt before they come into existence (evolution). The designer of such a system needs, therefore, to concentrate on the strategies for adaptation in addition to the mechanisms.

The authors argue that a system adapts to its environment and discuss the trade-off between maintaining a model of that environment, and the need to have an adaptive capability. At some point adaptation is bounded by the limits of the environment model which is maintained. This environment model may be viewed as a number of systems.

In terms of the systems which interest us, we currently impose the limit of an application. The system is designed to support an application and will be unable to adapt outside of that application. However, we argue that the system should be able to accommodate changes arising from different users of the system or the changing needs of individuals over time. In particular, the system should be able to accommodate the needs of individuals as they move from a primarily problem-solving activity to a routine cognitive skill.

We seek a level 4 system in the above taxonomy which must, therefore, possess a model of the systems with which it will interact (its environment) and the evaluation functions to select an appropriate response to the range of inputs.

## 3. THE COMPONENTS OF ADAPTIVE INTERFACES

An adaptive interface system needs to model the systems with which it will interact. What are those systems? The diagram in Fig. 1 shows the main components of HCI – the users, designers/managers/observers and the system (consisting of several tasks) itself. Although the designer/manager/observer view(s) of the system is important, we do not require the system to be capable of adapting to this view. Hence the system does not need to model this aspect.

There is clearly a need to model the users, but we need to consider which aspects we want to model. For example, we do not wish to model the physical characteristics of users, because we do not wish the system to be able to (automatically) adapt to the differences of individuals in this respect. Customisation is often more suitable here, particularly at the physical level of the interaction such as screen and keyboard layout. We return to the user model in Section 3.1.

The adaptive interface also needs a model of the system itself, so that it can adapt appropriately. We call this the 'task model' and discuss it in more detail in Section 3.2.

There is one further aspect of the environment which must not be overlooked – namely that the user and the task interact. Indeed it is this feature of the environment – an individual performing a task – which we would like our system to adapt to. Hence the need for an 'interaction model'. We return to this in Section 3.3.

We consider the term 'model' in line with Murray[7] to be B's representation of an entity, A, for the purpose of C. The models we require, therefore, are the system's models of the user(s), the tasks and the user-task interactions. The purpose of these models is to facilitate the provision of adaptive interfaces.

### 3.1 User Model

We argued earlier that for systems to interact, there must be adaptation by one or both systems. Humans are extremely adaptive, but there is evidence to suggest that some aspects are more adaptive than others. Van der Veer *et al.*[8] have argued that cognitive styles and personality factors are characteristics of users which are particularly relevant to HCI. They have analysed these factors in terms of how easy they are to change. Their analysis is shown Fig. 2.

In the upper part of the figure are personality and cognitive characteristics which are often the focus of commercial personality tests. For example negative fear of failure measures an individual's concern over not succeeding in a task, impulsivity/reflexivity measures the degree to which an individual will consider his/her actions before undertaking a task, and field independence relates to an individual's ability to transfer knowledge from one situation to another. The lower part of the figure indicates those personal characteristics which are less resistant to change. The individual's perception of their own general knowledge and problem solving abilities is argued to be more resistant to change than their learning style. A serialistic learner prefers to learn facts first and then to develop general concepts whereas a holistic learner tends to prefer an overview before filling in the details. People are typically relatively adaptable in respect of learning style, but still have preferred ways of learning.

This analysis provides the basis for a user model which captures the essential characteristics of individual user and offers suggestions as to where and how the system should adapt. If individual characteristics are resistant to change, then the system must adapt if the interaction is to be satisfactory. Consider a user confounded by his/her word processor. S/he can attempt to solve the problem by reading the (highly non-adaptive) manual, give up the current goal or make use of a more adaptive system. In most current interactions, the only adaptive system is another human who the user can question about the system. The hypothesis presented here is that the user will make use of the adaptive system if it is available or give up the goal unless s/he is highly motivated and willing to adapt to the non-adaptive source of help (the manual).

Notice that the analysis does not imply that personality and cognitive characteristics are completely non-adaptive. We recognise that people have preferred styles and ways of working, but under certain circumstances they can adapt these. However, if people have preferred working styles then they should be able to make use of these rather than having to adopt a new way of working because of an unnecessarily restrictive computer system. People use systems to achieve their goals because those systems help in achieving their goals. If these systems are difficult to learn or otherwise require a significant effort by the human, they will be bypassed or at best under-utilized.

There are many other personal features which may be pertinent to HCI. A preference for visual rather than verbal interactions, spatial and temporal abilities, the individual's personal constructs or perception of the locus of control over the interaction may all effect the quality of the interaction and be more or less adaptive in particular individuals. Gender and cultural differences have also been suggested[28] in this respect. Other authors argue that cognitive characteristics are not enough and

| Resistance to change | Dimension | Factors |
|---|---|---|
| High | Personality | Introversion/extroversion<br>negative fear of failure<br>Intellectual ability |
| | Cognitive style | Field independence<br>Impulsivity/reflexivity<br>Operation/Comprehension learning |
| | Perception of own competence | Epistemic/heuristic |
| Low | Personal knowledge structure | Learning-style serial/holistic<br>Prior knowledge |

**Fig. 2. Human characteristics and resistance to change (from Van der Veer et al.[8]).**

the conative and affective traits should be considered.[24] Others maintain that lessons from social and occupational psychology are relevant.[25, 26]

An approach to user modelling based on the user's personality and cognitive characteristics has the desirable property that such features are relatively stable and hence can form the basis of a long term, domain independent, user model appropriate for adapting many different systems. At the bottom of Fig. 2 (least resistant to change) is 'prior knowledge'. An individual's experiences with other systems and their knowledge of the current system will affect the interaction. This type of knowledge is domain dependent and is relatively easy to change by educating the user.

The distinction between domain-dependent knowledge and domain-independent knowledge appears useful and suggests that the user model can be seen as having two main components. Aspects such as cultural features, social norms, personality variables, cognitive style, learning strategy, and previous education might be considered domain-independent whereas knowledge of specific packages and generic tasks (such as word-processing, electronic mail, database retrieval and so on) might be considered domain-specific. The representation of knowledge in Intelligent Tutoring Systems is often highly domain specific. However, the level of domain independence of any particular facet of a user model remains a contentious issue in need of further research.

A popular approach to constructing user models is to capture the essential characteristics of one or more 'stereotyped' (generic) users and assign individuals to these stereotypes.[9–12] Individuals will typically belong to more than one stereotype and so the problem of resolving conflicting characteristics arises. A hierarchy of stereotypes could be constructed with general human characteristics at the top, gender and cultural features at the next level, age and occupation at the next level and so on. To our knowledge such a hierarchy has not been completed.

The stereotype approach has the advantage of modelling users at a coarser grain of knowledge. However, it must be stressed that stereotypes are not intended to classify people, they are a convenient user-modelling technique. Individuals assigned to stereotypes inherit the characteristics of the stereotype and only individual deviations from these traits need to be explicitly represented in addition. For example, the UC system[12] is a help system for Unix. It associates Unix commands with stereotype users. A novice user would be expected to know commands X, Y and Z whereas an intermediate user would be expected to know commands U, V, W in addition. Individuals who differ from the stereotype may be moved from one stereotype to another, or their profile is amended accordingly.

### 3.2 Task Model

A significant amount of research has been expended in developing task models (also called application models or domain models). A useful review of eleven task models is presented by Wilson *et al.*[13] They examine most of the popular task analysis techniques in terms of the breadth of user knowledge addressed by a technique, the user centred task dynamics (concerned with users behaviour), cognitive limitations on processing and use of the technique. Not all the methods of task analysis fit into these categoreis, but they do provide a useful basis of comparison. It is worthwhile to note their conclusions here. In particular, 'there is clearly no ideal, general technique for task analysis'.[13]

A variety of models have been proposed which divide the system space into layers. Probably the best known of these is T. P. Moran's Command Language Grammar (CLG)[15] which concentrates on four layers; tasks, semantics, syntax and lexical. The task layer describes the tasks which the system can perform. In a word-processing system these could be edit text, print the document, etc. The semantic layer identifies the concepts and functions of the system – for example the concept of a block of text or a cursor position, and operations such as identify text, move text, etc. The syntax level describes the commands available for accomplishing the conceptual (semantic) operations. (For example, position the cursor at the beginning of a block, position cursor at end of block and mark the block are the syntactic operations necessary to identify a block). The lexical or interaction level describes the keystrokes, mouse movements and so on, necessary to complete the syntactic operations.

Other authors have also suggested a layered approach. For example, Foley and Van Dam[16] identify four layers and Nielsen seven layers.[17] This is not the place to argue how may levels are required for a complete description of a system, but for certain there must be more than one! In fact it can be difficult to distinguish the different layers on occasions and it may be more useful to simplify this part of the description and concentrate on the difference between the logical (or conceptual) part of the system and the physical part.[18] The logical level deals with the functions which the system can perform and the concepts which it deals with. The physical part is concerned with the command language, keystrokes, mouse-button presses and physical displays associated with the system.

Most of the current task models deal with the functioning of the system at these two levels. TAG[21] sets out to predict the learnability of interaction languages. It is firmly at the physical level of task description identified above. Kieras and Polson's production rule model[20] and the GOMS model[22] both deal with logical and physical functions and how one maps to the other. Similarly CLG deals with logical and physical functions.

A significant shortcoming of these models is that they do not focus enough attention on the objects and the relationships between objects which exist in the system. Most of the approaches deal with tasks and sequences of tasks. They do not look at the structure of the objects. And yet if a user misunderstands the definition of an object, s/he is likely to encounter problems.

The outcome of this analysis is that the specific task model must be specified in terms of: the purpose of the system and the goals that it is capable of helping to achieve (what it is), the logical concepts and logical functions (what it does), their relationships (how it works) and the physical concepts, functions and their relationships (how to do it). See Benyon[14] for a more detailed discussion of this.

The form of these models is a matter for debate and personal preference. Grammars are typically favoured in HCI circles, and are probably most suitable at the physical level. In other disciplines such as information system design, graphical models such as dataflow diagrams (for the functions) and entity–relationship models (for the concepts) have proved popular and effective. Transition networks have been used (e.g. Alty,[19] Kieras and Polson[20]) as have production rules. Provided that the models can be consistently mapped into each other in order to ensure that the system is coherent, the form of the model may not be vital. However, it must be remembered that the task model has to be used for a number of purposes, and different models are suitable for different purposes.

### 3.3 Interaction Model

An interaction is the outcome of a particular user's interpretation of a task, which leads to the development of an intention to reach a particular goal. In order to achieve this goal the user may break down the task into sub-tasks and sub-goals. The computer system is the tool which the user employs in order to perform the task necessary to achieve the desired goal. The physical interaction devices, the logical structure of the software being used and the user's own mental model of both these and of the task will influence the ensuing interaction. The resulting user's mental model is developed from inter-relationships constructed from all these sources and will change in order to accommodate and make sense of new information arising from the interaction. The interaction itself takes place within a broader working environment which provides additional physical stimuli such as noise or changes in temperature and light. This environment will also have social influences on the user-stress associated with needing to achieve a particular goal in a limited amount of time, or competition, praise, help or criticism from colleagues. All of these factors will ultimately influence the interaction process and mediate the users goals and task intentions.

Once again, the level of the interaction needs to be considered. If the system is to adapt at the task level, the interaction must be modelled at this level. Some systems can be used for so many tasks that confusion arises for an individual trying to use the system for a small subset of those tasks (for example, authors and designers use the same word-processing or desk-top publishing system for quite different tasks). The same is true at the level of system functions. If the task and functions are understood, there may be problems at the syntactic and lexical levels of the interaction perhaps through the user's familiarity with other systems which use different keystrokes or through a poor understanding of a command language syntax.

### 3.4 Summary

This brief review of the models required by an adaptive system highlights the range of features of HCI that such a system will have to deal with. There are many models which may prove applicable, but exactly which ones are most suitable remains unclear. In many instances it is not even clear what should be modelled, let alone how to model each feature and the grain of the model that is required.

One way of learning more about the efficacy of alternative models is to try them out. The next section describes the Monitor system, the level and type of adaptivity which it provided and the models which it used.

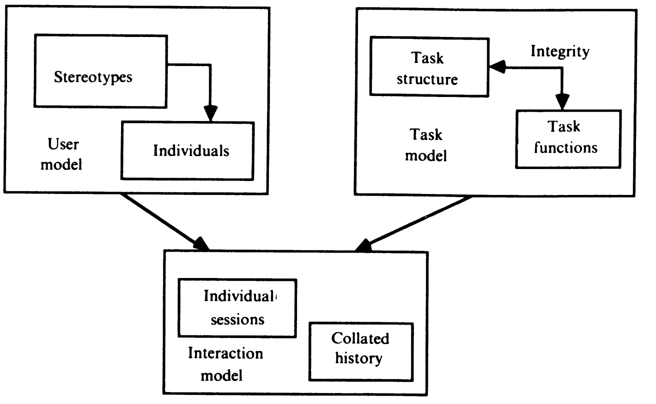


**Fig. 3. Overview of Monitor.**

## 4. THE MONITOR SYSTEM

Monitor is an adaptive interface 'shell', which provides a framework within which different dialogues can be presented to different users.[32] The initial prototype system was written in GCLisp running on an Olivetti M24 and provided alternative dialogues in the domain of CBT.[31] In the second stage of the project, this will be transferred to a more powerful machine with an enhanced environment and the range of applications extended to electronic mail and database query.

In line with the above analysis, Monitor can be seen as consisting of three models; the user, task and interaction models. Their relationship is illustrated in Fig. 3. The user model consists of stereotypes with individuals initially inheriting characteristics from the stereotypes and then adding individual differences as the system detects that they have departed from the stereotype. The task model consists of a task structure model, a function model and an integrity model which verifies that the functions can be supported by the structure. The task structure represents the system concepts and their pre- and post-requisites. For example in a mail system (see Fig. 4*b*) the concept of 'message recipient' needs the concepts of 'network address' and 'user identification' to be comprehended before it can be properly understood. The function model essentially describes the dialogues. The interaction model consists of two parts, one concerned with the interaction during an individual session using the system and the other with summarising these sessions into a user-task history. The session model is used to elicit data which can be used for making inferences about what the user understands and the history model for summarising a user's knowledge of a task.

The form of adaptation is the provision of different dialogues for different users and classes of users to deal with the changing needs of users, thus providing adaptivity at both the semantic and syntactic levels. In the database query application this could take the form of providing a form-filling dialogue for 'visualisers' and a command language for 'verbalisers'. In a command-driven application system, the novice user can be prevented from using certain commands, or varying certain parameters in order to maintain the simplicity of the system. The details can be revealed as the users experience grows or as the need for particular functions arises. In the system controlled dialogue (such as a menu-based system), the network is more closely defined and

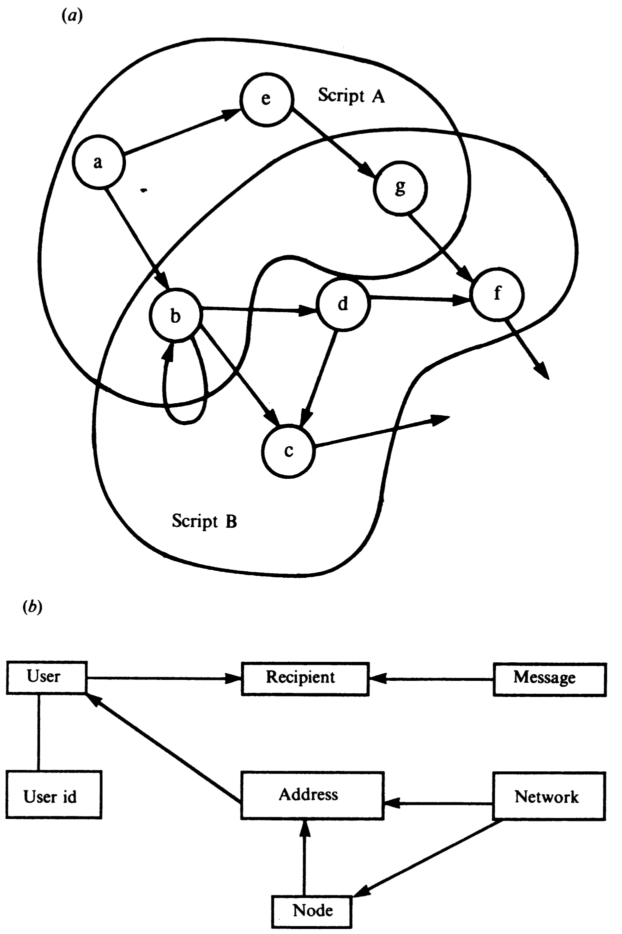


**Fig. 4.** ***(a)*** **Dialogue network showing alternative scripts.** ***(b)*** **Concepts in e-mail system. Interpretation: 'A (e-mail) message is sent to (one or more) recipients. A recipient is a (registered) user. A user has a user identification and an address. The address consists of a network and a node name'. Arrow heads indicate 'many'.**

nodes or sub-networks of nodes can be enabled or disenabled by the system to provide the adaptation. The provision of a co-operative adaptive system is also possible with this approach.

The task model performs two functions. Firstly it forces the domain expert to identify, carefully and explicitly, the elements and functions of the domain. In this role it acts as a design tool. Specifying the concepts forces the designer to consider where inferences about those elements can be made. The integrity part of the task model checks that every function uses only concepts which have been defined. For example in an e-mail system, the designer specifies the function 'Send message', giving it a syntax of

*Send*: *network*: *address*: *user id*

The system checks that the three concepts – network, address and user id – have been defined with their relationships to one another.

Secondly the task model forms the basis of the domain dependent part of the user model. The user's understanding of each concept and each function is represented by a 'competence factor' on a scale of 1–5. This rating is inferred by the system on the basis of the rules specified during the design process. Additionally the system maintains a 'confidence rating' for each competence factor which indicates how confident the system is in its inference of the users competence. Once again, these confidence ratings are specified during the design process, along with rules governing the summation of confidence ratings.

The task function model describes the dialogue. Each node in the dialogue is a production system which controls the presentation of text, the next node displayed and the inferences which are to be made from the user interaction. A more coarse-grained model is maintained by collecting nodes into 'scripts' (pre-defined dialogues) each of which corresponds to a user stereotype. Thus each stereotype user is initially directed by a script through the dialogue network. Figure 4*a* illustrates this.

Although some control is maintained at the script level, the majority is at the node level. A typical production rule in a node might be:

| | |
|---|---|
| If | this is script A |
| and | the user understands concept K |
| and | the user responds to this node with option x |
| and | they have gone through nodes a, b, b and d |
| then | update user model concept L |
| and | direct them to node f |

We will not describe the physical design of the Monitor system in this paper as it has been detailed elsewhere.[18,31] Suffice to say that the physical design is not particularly difficult (at least whilst performance is not an issue) using the current range of AI languages and tools. It is the logical design of the models and the elicitation of appropriate knowledge which has proved to be the bottleneck.

## 5. USING THE PROTOTYPE

The Monitor prototype has been used in a number of experimental situations to test both the operation and theoretical basis of system adaptivity. Our objectives were firstly, to establish the mechanisms and feasibility of adaptation, and secondly, to identify the user characteristics appropriate to adaptation.

Some initial pilot experimentation was conducted in May 1987 with nine undergraduate students using the first computer-based training prototype. This had a stereotype User Model which initially inherited characteristics according to the course which the student was taking. These traits were derived from the experience of the domain expert and the cognitive factors analysis of Van der Veer *et al.* (shown in Figure 2) and were amended for the individual through the task model inference mechanisms.

The six stages of this experiment were pre-test questionnaire, pre-test tutor's report, use of Monitor system, teachback session with tutor, post-test questionnaire and personality test. We were investigating the differences between measures of personality and procedural abilities of the subjects taken by Monitor, assessed by the tutor, and given by the subjects themselves.

The particular cognitive abilities we considered were subject's learning style, heuristic competence, problem solving style, motivation to learn and interest level. The tutor obtained additional information about epistemic competence while cognitive and epistemic variables were recorded by the system.

Initial results indicated that some further refinements of the experimental paradigm were necessary, and that a larger, and more variable, subject pool was required to enable us to draw firmer conclusions from later experimentation. We concluded that there was too little correlation on the factors which we had identified because of fundamental differences in the definition of terms. We also felt that observers' ratings and cross-variable correspondences could be made more rigorous and that a common basis should be identified between the different techniques employed for acquiring subjective data. It was apparent that the representation scales used within the system were too crude to make fine-grained distinctions. However, the feasibility of the mechanisms of adaptation based on a simple stereotype model of the user instantiated in a software prototype[18] was established.

Following this, the system was reviewed and enhanced in a number of ways. An editing facility was written and the first moves to a Shell environment made. The second prototype has a similar structure to the original but refinements were made and full technical documentation was completed. It was decided to conduct a much tighter experimental trial which would eliminate many of the variables and inherent assumptions of the pilot experiment. A study of 20 cognitive and personality factors was commissioned in order to verify to some greater or lesser extent how stable these are. The results were that, although many of the variables identified had some validity in cognitive psychology, they were not as firmly fixed and amenable to careful testing as other HCI studies had indicated. The problems of identifying an individual's style and then providing an individually-suited interface remained a significant barrier.

A new series of detailed experiments was designed which could deal much more carefully, one factor at a time, with cognitive abilities. The level of short-term memory (STM) was the cognitive factor initially chosen. Later experiments dealt with reference memory, with STM once more, and finally, with a combination of the two. The whole series has recently been completed and a full analysis is currently under way, to be reported in a later publication[27]. The preliminary results of this large-scale study seem to indicate that a dialogue adapted to an individual subject's level can enhance the interaction and task goals for that subject. In one of the early experiments, an experimentally slower dialogue significantly improved the knowledge acquisition of poor working memory subjects without significantly influencing the performance of those with good working memory. Although a certain degree of caution should be used when viewing such results in view of the small sample sizes, results do seem encouraging. There is good reason to suggest that dialogues can be made to be adaptive on the basis of individual cognitive characteristics. Moreover, there is strong evidence to suggest that whilst, for some dialogues, the performance is poorer for those with poor STM, this can be compensated for by modifications in the dialogue structure. Furthermore, because these adaptations increase the duration of the dialogue, and only assist the performance of those with poor working memory, it is advantageous for those with good STM to use the faster dialogue. Only an adaptive interface can optimise the performance of both groups.

## 6. DISCUSSION

The construction and use of the Monitor prototype adaptive system has proved to be a useful and valid research tool and has highlighted several things.

Monitor is certainly not a 'level 4' adaptive system in the taxonomy of Section 2, because it does not evaluate the likely outcome of its actions on an internal environmental model. Indeed it is not yet a level 3 system because it does not explicitly evaluate results on its environment. We agree with Totterdell *et al.*[5] that the system will only be as good as its 'notional world' – i.e. the beliefs which it embodies about the users and the system. Monitor's user model needs enhancing before it could be said to represent a human in any but the simplest of ways.

However, what the system does demonstrate is significant. Firstly that the mechanisms of adaptivity are feasible and relatively straightforward at a physical level. It is the conceptual modelling that is difficult. Secondly, that initial results are encouraging in that at least some cognitive factors (in our case the level of working memory) can be captured unobtrusively and accurately.[33] In a similar light, Williges and Vicente[30] report a system which successfully inferred users spatial ability by monitoring their use of commands. The unobtrusive elicitation of appropriate user characteristics must be a high priority for adaptive systems. Thirdly that the dialogue presented to individuals can significantly improve their performance. Putting these three factors together suggests that there is indeed a future for adaptive systems.

There are, of course, many problems. The utility of other cognitive factors in HCI is still a matter of contention, but it is hoped that the continuation of our research will contribute to that debate. The elicitation of peoples cognitive traits is difficult. They cannot really be expected to fill in a questionnaire before using a system. However, with attention to detail, the stereotype can embody many of these factors. There is a major consideration of the privacy of a user model and the problem of 'hunting' in which the system adapts just as the user has understood it! There also remains the problem of the task model. The techniques which we have do not appear suitable for the purpose of adaptive systems. A much more fine grained and wide-ranging model is required.

Our experience suggests that immediate research needs to centre on the elicitation and representation of appropriate user characteristics (user modelling) and the improvement of domain modelling. The current level of understanding of what constitutes the whole interaction (rather than just the interface) also needs more research and the development of appropriate modelling techniques.

We are currently planning a two-year project to produce a prototype development environmental for the creation and testing of User Models. These will be instantiated in a number of target systems and will allow in-depth exploration of the mechanisms and capabilities of adaptive interfaces.

In keeping with experience in the AI field, the bottlenecks we have met with in building adaptive interfaces have proven to be in the area of knowledge elicitation (though from users in general, rather than from specified domain experts) and knowledge representation. What seems now to be necessary has a twofold requirement: a means of enhancing known expertise in user modelling[28, 29] and a tool to assist in the design of interfaces with an adaptive capability.

This, to us, points to the development of a toolkit or 'Shell' type of system, similar in many ways to a classic Expert System Shell. This tool kit will be used for constructing embedded user models (EUMs)[7] (that is the representation of long-term user characteristics as a knowledge-base within an application system) through elicitation of both designer's and user's knowledge of an individual's cognitive and personal characteristics.

The User Modelling Shell is intended to produce program code which can act as an EUM for a specific requirement, or to produce generic models for a range of applications. We see this process as initially producing only prototypes, but the Shell may also be used as a design tool and could form part of a larger Human Interface Development Environment. It will function as a designer's tool, allowing early prototyping and iterative design practices to be applied to the construction of interfaces driven by a model of the user, and will be capable of providing either generic user models suitable for different types of interaction and application or individual dynamic models for customisable and adaptive systems.

The building of EUMs suitable for different types of interaction and application will take place by a process of incremental design and prototyping, based upon practical experimentation. The approach to the software development is that of 'evolutionary prototyping' – the system is developed as the problem domain is better understood and the prototypes are used in order to facilitate a better understanding of the domain. This approach is appropriate because the system cannot be precisely specified until it has been used, and because the building and use of the system is an effective way of fostering understanding of the subtleties of the knowledge representation. Since it is proposed to capture knowledge of some subtle human characteristics which have not been explicitly represented in a computer system before, this is particularly relevant.

Our experiences with building and using the Monitor system have lead us to recognise that adaptive systems have an important part to play in the enhancement of HCI. We believe the mechanisms of adaptivity to be fairly straight forward. The main issues remain in the modelling of users, tasks and their interaction.

User modelling is one of the most controversial and interesting aspects of HCI. Personality and cognitive traits can be effectively captured through personality tests, inferences, co-operation and carefully formulated stereotypes, but it is unclear how these differences effect the interaction of people and computers. Task modelling is well established, but it still lags along way behind the techniques employed in, for example, information systems design. The interaction model is perhaps the most impoverished as there is as yet no agreed model of what constitutes an interaction, let alone rigorous and levelled models which can represent it.

At present HCI guidelines distinguish only crude classes of users such as novices and experts. This needs extending to take account of individual, identifiable and measurable differences between users and how these fit in with HCI guidelines. As with many of the recent advances in information technologies, we believe it will be the provisions of efficient, effective and automated tools which will enable the necessary advances to be made.

**Acknowledgements**

This project was funded by the National Physical Laboratory under extra-mural agreement 82–0436. We would like to acknowledge the participation of all members of the project, particularly Peter Innocent, Cathy Thomas, Jaginder Shergill, Richard O'Kane, Steve Milan and Mathew Turner.

## REFERENCES

1. P. R. Innocent, Towards self-adaptive interface systems. *International Journal of Man–Machine Studies*, **16** (1982), 287–299.
2. D. Benyon, Adaptive computer systems. To appear in *Proceedings of the UK Systems Society Conference*, Hull, UK, (July 1988).
3. E. A. Edmonds, Editorial: adaptation, response and knowledge. *Knowledge-Based Systems*, **1** (1) (1987).
4. I. Clowes, I. Cole, F. Arshad, G. Hopkins and A. Hockley, User modelling techniques for interactive systems. In *People and Computers: Designing the Interface*, edited P. Johnson and S. Cook. Cambridge University Press, London (1985).
5. P. A. Totterdell, M. A. Norman and D. P. Browne, Levels of adaptivity in interface design. In *Proceedings Interact '87, Second IFIP Conference on Human–Computer Interaction*, edited H.-J. Bullinger and B. Shackel. Elsevier–North-Holland, Amsterdam (1987).
6. *User Centered System Design*, edited D. A. Norman and S. W. Draper. Lawrence Erlbaum, Hillsdale, NJ (1986).
7. D. M. Murray, Embedded user models. In *Proceedings Interact '87, Second IFIP Conference on Human–Computer Interaction*, edited H.-J. Bullinger and B. Shackel. Elsevier–North-Holland, Amsterdam (1987).
8. G. C. van der Veer, M. Tauber, Y. Waern and B. van Muylwijk, On the interaction between system and user characteristics. *Behaviour and Information Technology*, **4** (4) (1985).
9. E. Rich, User modelling via stereotypes. *Cognitive Science*, **3** (1979).
10. J. Jerrams-Smith, SUSI – a smart user–system interface. In *People and Computers: Designing the Interface*, edited P. Johnson and S. Cook. Cambridge University Press, London (1985).
11. A. Sutcliffe and A. C. Old, Do users know they have user models? In *Proceedings Interact '87, Second IFIP Conference on Human–Computer Interaction*, edited H.-J. Bullinger and B. Shackel. Elsevier–North-Holland, Amsterdam (1987).
12. D. N. Chin, User modelling in UC, the Unix consultant. In *Human Factors in Computing Systems, Proceedings CHI '86*, edited M. Mantei and P. Orbiton. ACM, New York (1986).
13. M. D. Wilson, P. Barnard and A. Maclean, Knowledge in task analysis for human–computer systems. In *Working with Computers: Theory versus Outcome*, edited G. C. van der Veer, T. R. G. Green, J-M. Hoc and D. M. Murray. Academic Press, London (1988).
14. D. Benyon, User models – what's the purpose? In *Alvey Knowledge-Based Systems Club, Intelligent Interfaces*

*Special Interest Group, Proceedings of the Second Intelligent Interfaces Meeting*, 28–29 May 1987, London, edited M. Cooper and D. Dodson. The Alvey Directorate, London (1987).

15. T. P. Moran, The command language grammar. *International Journal of Man–Machine Studies,* **15** (1981).
16. J. D. Foley and A. van Dam, *Fundamentals of Interactive Computer Graphics.* Addison Wesley, Reading, MA. (1982).
17. J. Nielsen, A virtual protocol model for computer–human interaction. *International Journal of Man–Machine Studies*, **24** (1986).
18. D. R. Benyon, P. Innocent and D. M. Murray, System adaptivity and the modelling of stereotypes. In *Proceedings Interact '87, Second IFIP Conference on Human–Computer Interaction*, edited H.-J. Bullinger and B. Shackel. Elsevier–North-Holland, Amsterdam (1987).
19. J. L. Alty and R. V. Ritchie, A path algebra support facility for interactive dialogue designers. In *People and Computers: Designing the Interface*, edited P. Johnson and S. Cook. Cambridge University Press, London (1985).
20. D. Kieras and P. G. Polson, An approach to the formal analysis of user complexity. *International Journal of Man–Machine Studies*, **22**, 365–394 (1985).
21. T. R. G. Green, S. J. Payne and F. Schiele, Formalizable models of user knowledge. In *Working with Computers: Theory versus Outcome*, edited G. C. van der Veer, T. R. G. Green, J-M. Hoc and D. M. Murray. Academic Press, London (1988).
22. S. K. Card, T. P. Moran and A. Newell. *The Psychology of Human–Computer Interaction.* Lawrence Erlbaum, Hillsdale, NJ (1983).
23. J. Rasmussen, *Information Processing and Human–Machine Interaction*, Elsevier–North-Holland, Amsterdam (1987).
24. G. Kiss, Why bother about user goals? In *Alvey Knowledge-Based Systems Club, Intelligent Interfaces Special Interest Group, Proceedings of the Second Intelligent Interfaces Meeting*, May 28–29th 1987, London, edited M. Cooper and D. Dodson. The Alvey Directorate, London (1987).
25. M. Gardiner, Psychological Issues in adaptive interfaces design. In *Proceedings of IEE C5 Colloquium on Adaptive Interfaces*, November 1986, Digest No. 1986/110. IEE, London (1987).
26. G. N. Gilbert, Cognitive and social models of the user. In *Proceedings Interact '87, Second IFIP Conference on Human–Computer Interaction*, edited H.-J. Bullinger and B. Shackel. Elsevier–North-Holland, Amsterdam (1987).
27. S. Milan, D. M. Murray and D. R. Benyon, NPL Divisional Report (in preparation).
28. C. Fowler and D. M. Murray, Gender and cognitive style differences at the human–computer interface. In *Proceedings Interact '87, Second IFIP Conference on Human–Computer Interaction*, edited H.-J. Bullinger and B. Shackel. Elsevier–North-Holland, Amsterdam (1987).
29. A. Dillon, Knowledge acquisition and conceptual models: a cognitive analysis of the interface. In *Proceedings Interact '87, Second IFIP Conference on Human–Computer Interaction*, edited H.-J. Bullinger and B. Shackel. Elsevier–North-Holland, Amsterdam (1987).
30. R. C. Williges and K. Vicente, Visual momentum as a means of accommodating individual differences among users of a hierarchical file system. In *Proceedings 5th Symposium EFISS*, Risø National Laboratory, Denmark, November 1987, edited J. Rasmussen and P. Zunde. Plenum, New York (in press).
31. D. Benyon, P. Innocent, D. Murray and J. Shergill, Experiments in adaptive interfaces. In *Proceedings of IEE C5 Colloquium on Adaptive Interfaces, November 1986*, Digest No. 1986/110. IEE, London (1987).
32. D. Benyon, MONITOR – a self-adaptive user interface. In *Proceedings Interact '84, First IFIP Conference on Human – Computer Interaction*, edited B. Shackel. Elsevier–North-Holland, Amsterdam (1984).
33. D. Benyon, D. M. Murray and S. Milan, Modelling users' cognitive abilities in an adaptive system. In *Proceedings 5th Symposium EFISS*, Risø National Laboratory, Denmark, November 1987, edited J. Rasmussen and P. Zunde. Plenum Publishing, New York (in press).