User Modelling SIG Workshop

Models of Users in System Design

Draft Report for Comment

Cosner's House

6th and 7th June

Contents

Management Summary

This two day workshop, organised by the HI Club User Modelling SIG took place on the 6th and 7th June 1988 at Cosner's House. The workshop was attended by representatives from academia and industry (see attendance list) with the objective of exploring the issues concerned with incorporating models of users in to the systems design process.

The term *designers' user model* is used to describe all models of the end-user that a designer might use in the design process. These include models of the user's tasks, predictive models of users performance, models of types of errors user's might make and of how users will learn to use the system. All these aspects and others are covered by the term designer's user models.

The aims of this workshop were to identify the requirements for designers' user models and examine models that are currently available to determine if they are usable within the design process. Additional objectives were to produce recommendations to systems designers on what models can currently be used successfully, describe what models are likely to be available in the medium term and identify important research issues for designers' user models.

The main conclusions from the workshop are that:

- Current user models are insufficient for the purpose. They provide input to the design as it is nearing completion. This is just the point when user testing can be used effectively. Therefore new models need to be developed which focus on the early stages of the design process where they can make a significant impact.
- User models can be used to structure the design process but little attention has been paid to this area except for the simplistic structuring provided by the linguistic models of Moran and Foley.
- Building user models should not be a single activity performed as the design is nearing completion. User models should be started during the requirements stage and be refined and verified as the project progresses.
- Currently a large amount of effort needs to be invested in building models and any pay-back comes at the end of this process. The effort/pay-back curve needs to be more linear so that user models provide some benefit for reduced effort. This will allow models to be used on smaller projects.
- Some benefits derive from the process of building a user model by focussing attention on understanding end-users. These benefits can be important even if the model is never used predictively.
- User models need to represent aspects of users that are directly relevant to the design process. Currently they focus on what can be modelled easily ie. aspects that are well understood by psychologists. Further work is required to identify the important user characteristics that need to be considered in the design and how to model them.

- Little validation has been carried out on user models to demonstrate their predictive power. Further validation is required before models become accepted.

The workshop was viewed as a success by those attending it but the process of discussion was just starting at the end of the workshop. A further workshop is being planned which will involve more interface designers and model builders and will focus on the main topic of the workshop.

Introduction

The issue of getting Human Computer Interaction (HCI) issues into the system design process is becoming increasingly important. Most design methods such as JSD, SSADM, Yordon, IEF have very little to say about HCI issues or designing the user interface. Providing user models to designers is one way of improving this situation. By providing designers with tools with which to represent users and think about the problems will allow them to make informed design decisions. This workshop aimed to look at this area to examine the types of models that are available, how they relate together and where they fit into a design process.

The term *designers user models* has been used to describe this type of user model. This is used in a broad sense to include such models as:

Interaction models
Learning models
Conceptual models
Performance models
Error models
Task models

In fact it includes all models of the user that the designer might use in a design process. The workshop aimed to pull together people from different backgrounds with interests in this area to identify what the important issues are and attempt to identify some recommendations for designers on what they should use.

Objectives of the Workshop

The objective of this workshop was to bring together model builders and model users to start process of discussing the requirements for user models and how they can be used in the design process. The workshop aimed to:

- Identify the requirements for designers' user models
- Examine the models that are currently available
- Look at the relationship between different types of models and the rôles they play in the design process
- Produce recommendations to systems designers on what can be used currently
- Identify what is likely to be available for designers in the medium term
- Identify important research issues

Papers

The following is a short description of each of the papers given at the workshop. These descriptions are only brief summaries of what I can remember about the papers and are not meant to be critical reviews of the material.

Steve Payne - *User Modelling in Support of Exploratory Learning*

This paper was concerned with modelling 'learning how to do things from examples' and exploring how this type of model can be used to identify techniques for supporting this style of learning.

Steve described the induction of procedures from examples and gave the following models are exemplars of this approach:

Anderson's ACT*

Kieras & Polson's GOMS

Van Lehn

The basic mechanisms used by these types of model are similarity based generalisation and repair. This is a very simple mechanism which goes some way to explaining learning from examples and can provide support for bug remediation in a tutoring system. However these models suffer from a number of problems which invalidate them:

1. Generalistation needs good examples
2. Bugs don't explain errors
3. Pure procedures aren't the whole story

This can be seen in the problems with algebra mal-rules which attempt to extract procedural descriptions of errors from observations of errors. These descriptions fail because too many rules are required, they are ambiguous, there is a semantic influence on the mistakes and you can't distinguish slips from mistakes.

Steve then went on to describe models which attempt to provide a more detailed analysis of procedures. The exemplars he gave of these types of models are:

EBG (Explanation Based Generalisation) Mitchell et al

PUPS Anderson and Thompson

EXPL Lewis

This models use methods of causal analysis and deduction of generalisations from examples. This provides support for trace based tutoring but ignores the subtleties of generalisation of procedure parameters and what is identified as a parameter. An example of PUPS was given which showed how analogy is used to generate a new procedure to produce a desired outcome.

Steve then went on to examine how we achieve causal analysis and drew on the work of Lewis (1986). This points out that there are a number of heuristics; the identity heuristic, the obligatory previous action heuristic and the loose-ends heuristic.

Four observations were then made about the generalisations of procedures:

1. We use task semantics to learn task-action mappings TAG
2. Device space may contain new conceptual entities eg. string
   Even when mappings are near identity there may be a learning load
3. Grouping of device space entities for TAG mappings may be new
4. Some device space operations may not affect the goal space at all. eg. Copy Buffer, MacDraw Lasso

Steve then described how this relates to the yoked state space hypothesis drawn form the work of Moran (1983) ETIT. This consists of a goal space and a device space and a mapping between the two. The user needs to maintain both these spaces and accomplish transformations in the goal space by applying operators in the device space. He gave an example of cut-and paste editor showing how many of the goal space objects are not represented in the device space and sometimes require complex mapping. He reported work where the notion of "Cut" and "Paste" actually prevented novices from realising that the text is held on the clipboard and that multiple pastes can be made into a document. He felt that "Save" and "Insert" would do better.

Tony Simon - *A Trade-off Approach to Identifying Suitable User-Models*

Tony Simon's paper was concerned with trying to identify the trade-off decisions a designer can make when choosing a user modelling technique. He focussed on user models that are used to predict the behaviour of a user interacting with a proposed interface and was concerned with whether the models are worth using and whether they can be used profitably.

He started by identifying the problems for designers in choosing from the current range of modelling techniques that are available and pointed out that the techniques will only be used if they are easy to use and cost-effective.

The proposal he put forward to this problem was to identify a suitable modelling technique by asking questions about the aspects of user behaviour the designer is interested in predicting such as "Expert" behaviour, common errors or execution time. These can then be used to explicitly select a model and make a trade-off between the different approaches.

He then described a 3D space in which he had mapped the different modelling approaches onto. I won't replicate this space here (see the paper in HCI'88) but he mapped the following techniques onto it:

- Model Human Processor
- Key-stroke Level Model
- Task Action Grammars
- Goals Operators Methods Selection
- Cognitive Complexity Theory
- Integrated Cognitive Sub-systems
- Programmable User Models

Some of the trade-offs are concerned with the effort required to use the model eg. whether it is task instance analysis or task space analysis and the re-usability and predictive power of the model. Another concern is what does the model user (i.e. the designer) have to know to use the model. Do they have to be familiar with the psychological literature, or be able to write production rules, and if so is this an appropriate requirement for designers

The summary of the talk was that the models described, model different aspects of the user and there are trade-offs in selecting between them. The final point was that it is important to assess what's involved with applying a model, what inputs it needs and whether the interface designer supply these inputs.

Andy Whitefield - *The Use of the Blackboard Model in CAD System Development*

Andy Whitefield's talk described how a blackboard model of the user had been used to identify problems in a CAD package. This work had been carried out as part of the Alvey project "A user modelling tool for MMI design" .

The user modelling tool comprises a blackboard model of the user, in this case an engineering designer, a model of the CAD programme and a mapping relationship between them. The blackboard model is derived from Hearsay-II and consists of knowledge sources derived from analysis of verbal protocols. The model describes how an engineering designer uses different knowledge sources to perform a task and how they move between levels of description to produce a solution.

The model of the CAD system in this case BOXER, was split up into a number of levels reflecting the organisation of the program and also the split between the definition phase and the session phase. This model proved to be a long way from the design task and the blackboard model contained details of how the user makes this mapping.

A number of different mapping relationships are possible with the approach such as between direct and indirect operations and global and local behaviour. The general approach is to allow developers to reason about system behaviour in terms of the relationship between program functions and knowledge use.

An approach for how the model should be used was also described which involves developing the user model and the program model, together with criteria for the desired relationship between them and then altering the program model to see if there are improvements. This supports incremental changes to an existing system which was claimed to be non-prescriptive, non-calculational and have a broad scope.

In summary Andy identified that a full application test was required and there were still some questions over the ease of use of this approach. It is still necessary to identify an appropriate rôle for this type of model in the system development process as it is easier to use it to evaluate software or a design that already exists, rather than helping to formulate a new design.

Harold Thimbleby - *Models, Metaphors and Phenomenology*

Harold Thimbleby's talk presented a formal description of user models, examining why they are needed, how they are used and whether they are naive. He was particularly looking at the user's model of the system and then the designers' model of this model.

Harold started by examining distinctions between systems which he split into necessary formal, and chosen distinctions which can be instantiated as physical systems, coherent systems and mental systems respectively. He then made the observation that the user and computer are large systems with a narrow interface which acts as the bottle-neck. During an interaction with a system the user model is used to change state and help keep the interaction in step.

The task of the designer is to abstract a model of the user and understand the computer model so that they can design interactions between the two. He pointed out the differences between users, computers and designers as follows:

- Users create their future through acts of choice constrained by their intentions
- Computer systems unroll their future through acts of necessity determined by previous states
- Designers commit the computer to anticipate the future behaviour of users

Harold then identified a number of procedures and properties that the system should support which have implications for the users'model of the system:

Decision procedure:	Can the user know what the system can do, ie. is this user theory translatable to a computer theory.
Enumeration procedure:	What user theories are translatable to computer theories.
Search procedures:	What is "the nearest" translatable theory? homorphisms
Safety properties:	Can the user prove the system has done what was wanted.
User model sufficiency:	(every) translatable computer theory is user valid.
User model necessity:	(every) translatable user theory is computer valid.
User model closure:	(every) translatable user theory has a computer translatable theory (avoid non-standard models).

A number of problems exist with user models. These are the problems of infinite regress where models of models of models ... are created. They are reflexive and only have validity by convention and user models are hidden variables for 'user model', read 'model of user model'.

In summary user models permit usability issues to be articulated but they have no theoretical basis in general consisting of folk psychology and they are naive. Harold thinks that phenomenology is the right way to go.

Steve Draper - *The Implications for General Models of Users of Three Paradigms: Direct Manipulation, Dialogue and Spreadsheets*

This presentation was concerned with identifying implications for general cognitive theories by looking at three paradigms of interaction. The approach taken was to examine three 'ideal' paradigms of interaction and try and account for them in a theory of action drawing on ideas from AI planning to help explain them. This involved coming up with dreams of ideal forms of interaction (little bits of life that go well that would like to capture and put in a machine) and then comparing them to see how they relate together. The three paradigms chosen were:

Direct Manipulation:	Working on a model world
Dialogue:	Human conversation - what makes it go smoothly?
	eg. Conversational repair, recognition of intentions
Spreadsheets:	more the function building approach

A new characterisation of a theory of direct manipulation was then presented which identified that the model world that is being manipulated is represented at a particular level of abstraction. The semantic and articulatory distance are predicated on having this level established. Direct manipulation systems get into a problem of choosing what is an appropriate level of representation.

The question was then posed as to how a direct manipulation interface affects the thoughts of a user. When using a direct manipulation interface users don't have to plan ahead, they can perform operations one at a time and feel that they are making progress in the obvious direction. There is no feeling of being in a maze. What is critical is the semantic directness of the display. Each operation is a single step which is cheap to evaluate and redo if necessary. However in direct manipulation the goals that can be achieved are fixed at one level. A pure direct manipulation interface offers no support for achieving low level or high level goals. This is in contrast to a conversation where people can slide up and down scales of goals in an explicit way supported by mixed interaction to identify the current goal.

The presentation then moved on to look at *intermediate user needs*. These are things you want to do that don't fit neatly into an interaction paradigm.

Diagnostic feedback: In direct manipulation you get feed-back at every instant. In conversational dialogue the detection and repair of the dialogue is the responsibility of the other conversant.

Reference: Direct manipulation systems have problems where the universe of objects you are trying to reference is large. In a small universe pointing works but direct manipulation systems turns referencing into a navigation problem in a large universe which isn't very satisfactory. Language on the other hand is very good at handling reference and can reference one object in many ways.

The presentation now moved on to look at spreadsheets which have the property of tying together the specification and the result which has similarities with functional programming. It would be nice to just provide the result and the system to work out possible missing values. Spreadsheets have operational and figurative aspects where a procedure is regarded as a black box and you have to look inside at the components and do a causal analysis to work out what is happening.

However there are many problems with spreadsheets, they only have one way pointers, they have a high viscosity (ie it is difficult to make small local changes without affecting the whole spreadsheet) and they should not be constrained by rectangular grids and should support loose cells and arcs.

To understand how to use different paradigms effectively we need to understand how they can be combined together and where each works well. For example in most direct manipulation systems they fall back into using language at various points with complex dialogue boxes. Another area that can be needs to be explored is programming by example where the system has limited inductive power, you can either "do what I do" or "do what I said" and there is currently little support for "do what I said".

There then followed an interesting discussion about encapsulation and and saving units of action. It was argued by Steve that in a pure direct manipulation system this was not possible because you would need to have actions as object which would require a shift of level in the direct manipulation world.

In summary Steve was pushing for the understanding of "pure" paradigms with describeable properties that can be described to designers. However real systems are not pure and require a mixture of paradigms. Therefore the way forward is to understand how to combine pure paradigms together in useful ways to meet the interaction needs of the application and the user.

Discussion Groups

There were two sessions of discussion during the workshop. The discussions on the first day were concerned with clarifying issues, defining important relationships and specifying requirements on user models.

The discussions on the second day were more focussed and involved interface designers describing design problems they had to model builders in an attempt to identify models that could help support them.

Day 1

The purpose of the discussion groups in the first day of the workshop was to set the scene for future discussions and clarify some of the confusing issues in this area. Three workgroups were formed which looked at the following topics:

Group 1 : Relationship of Models of Users to Task Analysis (and other models)

Group 2 : Models and their Relationship with System Design

Group 3 : Designers' Requirements for Models of Users

Group 1 : Relationship of Models of Users to Task Analysis (and other models)

Present: Lisanne Bainbridge, Claire O'Malley, John Long, Steve Payne, Tony Simon, Ian Clowes

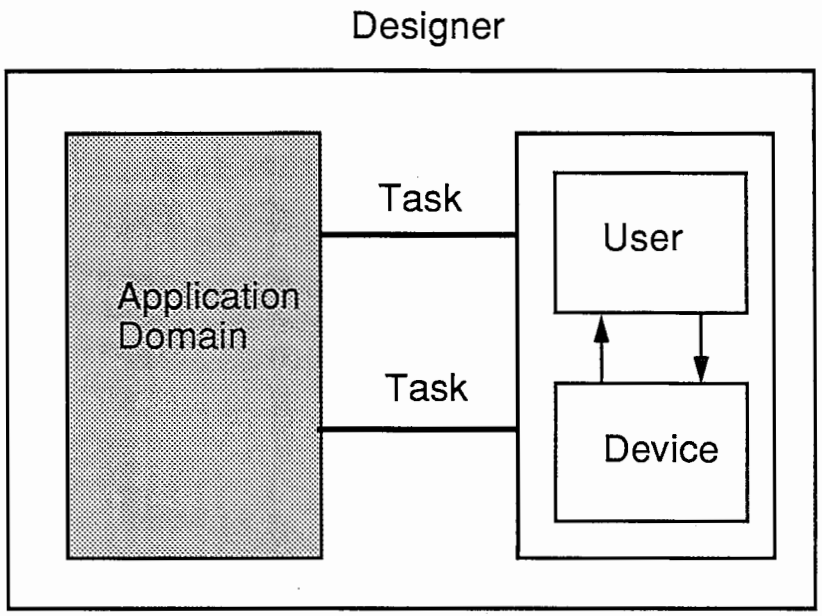
This group was considering the relationship of different types of model to one another. It started by going round and identifying the areas of interest of each of the participants. This identified people as model builders or model users and what they saw as the purpose of the models they wanted to build or use.

John Long defined task as the realisation of potential, not just what you do to a device but the changes you bring about. In looking at task analysis you need to differentiate between the process and the product.

Task Models describe the changes in the world that have to be made (the application domain).

User models describe how those changes are carried out, either as general tasks or as specific instances.

A diagram was produced showing the relationship between task models and user models:



Some attempt was made at categorising the different stages in the design process and the different rôles that models take at these different stages. It was felt that a useful exercise would be to compare the properties of models with the properties of design.

Group 2 : Models and their Relationship with System Design

Present: Nigel Bevan, Andy Whitefield, Alan Dix, Kee-Yong Lim, Vic Maller, Angela Sasse

This session was concerned with the rôles that different types of user models might be expected to play in the design lifecycle. This of course meant it was necessary both to identify the different types of model available and to agree some view of the design lifecycle. The first of this was done crudely but acceptably by adopting the now standard three-way division: the user's model of the computer, the computer's model of the user and the designer's model of the user.

The content and focus of these models was agreed as important for system development, but any refinement of the classification was deferred until after consideration of the design lifecycle.

Developing and agreeing some view of the design lifecycle occupied most of the remaining time. A "Waterfall" view was accepted as a starting point, despite its acknowledged deficiencies. After discussion of the details, the following (or something very like it) was accepted:

1. Statement of user needs
2. Requirements specification/definition
3. Systems specification (including a possible separate user interface specification)
4. Design
5. Implementation
6. Laboratory and field tests

Although a typical description would include a feedback loop from one stage to its predecessor, we recognised the necessity for rather larger feedback loops which would jump several stages, and especially to the requirements and specification. And of course there was general agreement with the necessity for iteration.

From this point, any proposals about the relationship between models and design tended to be made with several qualifications and constraints attached. At its simplest, the general suggestion was: that a user's model of the computer would be of benefit only at (or after) the implementation stage, when the user had a computer to use and to model; that different sorts of designer's models of the user were appropriate to specification and design; that the computer's model of the user occupied a rôle that was largely separate from system development lifecycles, since its is part of the system; and that task analysis was appropriate to developing the statement of user needs and requirements definition.

The different sorts of designers' models of the user were fairly ill-defined, but were suggested as being procedurally-oriented analytical models.

In the time available, and given the need to consider both types of models and the system lifecycle, this was as far as the discussion went. Any further meeting ought to be fairly clear about these as a start, and concentrate on considering the roles of particular example models.

Group 3 : Designers' Requirements for Models of Users

Present Steve Draper, Angela Lucas, Harold Thimbleby, Keith Dickerson, Mike Wilson

The scope of this session was taken to be the system design requirements from the designer's point of view, the starting point for our discussion was to list the requirements for achieving a desirable end-point in the design cycle. This list should not be considered to be exhaustive, but rather it served to provide the group with a number of points for discussion.

- Gross style of user interface and the impact of the interface on the user.
- Task analysis.
- Minimise errors : The system should both be easy to use and easy to learn. Learning effort should be treated as a function of user type, we must therefore be able to define the user population. The application of good human factors principles will enable the designer to recommend levels of user modelling appropriate to the task.
- The system should be able to produce output quickly at the beginning of the design cycle. If appropriate information for user modelling is fed in at the top level, then the design of the system should be pointed in the right direction immediately. The amount of information which is required for this implies that even at this early stage, the designer has already made commitments which may not be changed lightly. The moral of this is, 'get it right before you start !'

Harold Thimbleby described this aspect of design with the *'cupboard analogy'*, he told the story of how he had tried at one time to make a cupboard, he carefully took all the appropriate measurements of the gap in the wall which he wanted to fill, he bought all his materials according to these specifications and then went ahead and built his cupboard, after a lot of time and effort, he then tried to fit the cupboard into the gap, he was most surprised that it didn't quite fit. With hindsight, he was interested to watch a professional to do the same job, the carpenter built the cupboard to fit the gap, he did this by only making measurements that were appropriate to start the task, once the base was in place, he then took the measurements for the sides and then fitted them in. In this way he built up the cupboard in little stages until it was complete and more to the point a better fit than Harold's had been. At many stages in the building of the cupboard, the carpenter has to make decisions which are appropriate to a particular stage in the design. At each of these stages, the designer has to make a choice and these choices have repercussions which permeate through the rest of the design phase. The moral to this story is that the carpenter built the cupboard in stages, he could recover from errors easily and compensate for them at the next stage of development. This means that he wasn't left at the end of the process with an ill fitting cupboard, by the time Harold noticed that his cupboard didn't fit, it was too late to do anything about it. After all what's the good in having a perfect cupboard at the end if it doesn't the space it was intended to ?

Some conclusions to draw from this are:

- Evaluation should not only be an on-going process, it should be appropriate to the task to which the model is being used. Peter Windsor suggested that we should model the task, the system and the user and then ask questions like, is the system being used in the way it was intended, which means effectively, that the users are the tools which are used to evaluate the system. A user model makes accurate predictions about what the goal inductions are that might be able to do this.

How would this 'cupboard' analogy be evaluated? Peter Windsor reckons that this form of on going evaluation is better than getting loads of advice at the outset. Mike Wilson asked, how is this on going evaluation performed? What is the minimal unit of measurement for the real world, for the carpenter, it would be a bit of wood! Harold,wondered if it was the case that we can't consider anything less than the whole system eg, what's the good of testing the steering wheel of a car, we would find this difficult in isolation, it makes more sense to test the performance in relation to the system as a system. Steve Draper disagreed, if we build our interfaces incrementally by debugging techniques, this is analogous to building a warped cupboard up from the ground, this cupboard may not be 'ideal'; one end might be longer than the other, but if it fits the gap and it contains secure shelves and the door fits, then it is considered to be an adequate cupboard. However, if we were to break the cupboard down into its meaningful constituents, the bits of wood, these are not meaningful constituents. He summed up this argument by saying that a builder lives in a twisted world, where absolute measurements are not a good basis for design, this implies that we need different tools for a practical interface design.

The fact that user models have therefore to deliver something which is useful against a sometimes warped specification, should make us question some current design features eg. pull down menus, do they tend to be too big ? Do they work ? etc

- The designer wants to be sure that the user population find the system to be indispensable, with the effect that users buy upgrades of the system rather than just going and buying some a similar system with other features. The system should therefore be addictive, ie. use of the system becomes an end in itself.

It is difficult to model this kind of usage, it usually takes many exposures to the system before the user becomes addicted and often, after a long bout in front of a terminal, a user may lose a former compulsion.

We must consider this 'addictiveness' in terms of goal formation and the conditions under which we internalise new goals which we will pursue for their own sake and not just as a means to achieving other goals.

Should the designer of a system which controls both the user's purchasing behaviour and enthusiasm ever wonder whether the user feels that the time spent learning the system has been worthwhile ? It was noted by the participants in the discussion that the power of the system is generally measured by its ease of use, a powerful system requires a longer time spent in training. As an example, Peter Windsor, noted that any good CAD system is not easy for novices to use, but after six months of training the return in terms of expressive power is well worth the effort. This is because of the complex set of functions required to support the task involved in CAD.

To produce a system that is both easy to use and at the same time powerful is difficult, the group felt that the designer ought to consider that there to be two types of users in this case ; the novice and the expert, and so the system must also take on two different forms for each of the user groups.

A solution to this would be to build a powerful system which is easy to use, by incorporating such facilities as help features and undo buttons etc. An example of this is the Macintosh, where the direct manipulation proves easy for a novice to use, they don't immediately have to handle the full functionality of the system. A good interface reduces the initial problems for a novice. The utility of the system was considered to be more important than ease of use. Functionality should not be exclusive with ease of use, and neither should conflict with naive principles.

This discussion brought the group round to consider the inductive techniques which a user is likely to use. A basic result of user modelling research is that different instructions are required for different users ie, the word 'storage buffer' means different things to more or less experienced users. All of us humans have the facility to induce from a situation, but designers must think carefully about what is a sufficient prompt.

In discusing the induction of models by users, the group considered that it was necessary to consider,

- An information flow requirements analysis: what will users need to know about a given interface, and how it can be delivered effectively ie, explicit lecturing.
- What mental model will users have of a system ie as a prediction of induction by users.

- Goal formation, user models predict what goals the users will form and how to manipulate these goals, this would include how to make an interface addictive; how to get users to know about and remember when relevant features of the interface; how to support the creative use of the system for functions that the designer didn't anticipate and predicting whether the interface will be used as the designer intended.

In learning to use a new system, novice users tend to rely heavily on the core functionality. Angela Lucas pointed out that there are often at least two ways of tackling a particular task; the easy long way and the more difficult short cut eg Unix macros. The tricky bits are not often used and depend on the user's awareness of the capability and the functionality of the system. Bearing this in mind, designers might do well to ask if the system is being used as they intended or in some new way which they may either want to discourage (advertising in e-mail) or redesign the system in order to support it better (eg spread sheets for financial planning on top of retrospective accounting).

The discussion moved on to the topic of rapid prototyping,

Support for the rapid prototyping design cycle

- Prototyping should be performed immediately after the specification. Thus making the first leap as productive as possible. Rapid Prototyping should allow us to predict user errors, predictive measurement.
- Information from the user model must be delivered early in the design cycle.
- User modelling must specify how to change the interface (not just measurements or statements that there is a problem).
- To support rapid prototyping the user models must be easily, quickly and incrementally modifiable like the prototypes themselves. The user model has to support small changes at small costs and support an incomplete prototype. This is different from a complete model of a complete interface.
- Rapid prototyping should support middle out design ie, beginning with a skeletal prototype, which is far from complete in any respect except that it runs. User models must give useful information about very incomplete designs.

Steve Draper considered that rapid prototyping, if it is performed on existing systems and is interpreted in terms of the development of the system may be divided into three main areas,

1. Analysis
2. Empirical
3. Remediation

The group discussed whether they believe in the usefulness of rapid prototyping as a constraint for what designers require of a user model. The design of a good system might be said to be dependent on whether the designer is aware of the orthogonal components of the system, it is also recommended that designers work in design teams rather than independently. We should always be aware of subdividing the problem into smaller subproblems.

Rapid prototyping, is considered useful to demonstrate to the customer that something can be built and is therefore an important marketing device, they reduce ambiguity for the customer and can provide a means of walking through the system, but in general the prototypes have no functionality. Harold Thimbleby thought that the view that there is no underlying functionality behind the prototype is false, there is after all the prototype car which can be driven around as a means of demonstration. There is no manufacturing advantage with the model, once you've built it you can do the real thing. However it is a different proposition if we are talking about an abstract prototype.

Mike Wilson, said that a prototype system which has screens and nominal processing capacity, requires that the designer provides some level of specification although accuracy or speed may not be a requisite. The interface of a model is fine as far as it goes, if you build a prototype you need not implement it to full functionality, but in building an interface, design is of paramount importance and any valid requirements have to be considered. In HCI terms, a prototype has got to look like the finished product.

Rapid prototyping is valid, but we should recognise its limitations, Peter Windsor, reckoned that rapid prototyping tools are good in theory, but in practice present problems. Harold Thimbleby, acknowledged that rapid prototyping is expensive, real design requires a rapid prototyping tool which might end up being more expensive than the system. Keith Dickerson, thought that it would be more useful to get hold of better feedback.

On the plus side, prototypes are quicker and cheaper to build than the real thing. They can be built without any loss of information, but should be considered to be more essential in terms of their power for testing. Although, testing is not always viable as it relies on theory to interpret the test results. User models must be used to interpret the performance of the users against the theory. We need to be able to deliver user models which integrate well with our user measurements.

The question was posed, how do we test user model specifications against the prototype and why would we do it ? To ensure that user models are consistent with the prototype, we need to be able to make changes at low cost through out the cycle eg the cupboard example. There are many intermediate stages which should be seen as a skeletal working version which is distinct from the real thing, but allows us to make the relevant additions and changes to ensure consistency. This method amounts to a 'middle-out' technique rather than bottom-up or top-down.

A suitable test is a task analysis, we must consider whether the system does the task in the same way that the user would perform it? In order to do this, we must make the assumption that both the user and the system have a particular way of performing a task. If the mapping between these techniques is good then the model of the user is appropriate (and vice versa). User models should make the designer think in terms of the user and the language which is required to open up the design procedure.

One problem that such a session as this should address, (and this might seem a bit obvious) is that system designers ought to understand the value of HCI techniques, many designers design for themselves and so feel that they don't need any models,they are their own models. Designers *must* be made aware of the value of user modelling especially in terms of testing procedures. This brings us back to the earlier point, the importance that designers be made aware of their target population

Day 2

On the second day of the workshop it was decided to focus in on the main objective of the workshop and discuss what user models have to offer to system designers. The success criteria for user models in the design process are that they help come up with "better" designs in some way, where better includes; higher quality user interfaces, faster development time, less prototyping and systems that are easier to enhance. The participants were split into two groups and given the task of seeing how user modelling techniques could be applied to design problems provided by designers in the groups.

Group 1 Applying User Modelling Techniques

Present: Nigel Bevan, Peter Windsor, Harold Thimbleby, Vic Maller, Mike Wilson, Steve Draper, Lisanne Bainbridge, Steve Payne

This work group looking at applying User Modelling techniques to user interface design problems. Two broad topics were considered: the environment in which the decisions were made and the decision making process itself.

The first part of the discussion covered the familiar, but important ground of the effect of project organisation on user interface design. The following points were raised:

- The 'battle' to get human factors specialists into design teams is being fought today. It has not been won.
- Who in a design team makes decisions about the user interface? In a large, hierarchically organised project, it is possible for the core design team to leave important decisions to the implementors. A typical example is the choice of terms for menu items. This is not necessarily deliberate, but may be due to the designers not knowing the impact that those aspects of the design will have on the system's usability.
- The quality of the task analysis will have a large impact on the ability of designers to make good use of user models.
- The user population have to be prepared for change or they may not accept a new system despite a high quality user interface.
- Designers, like all human beings, act at various levels of intellectual sophistication. If designers are to produce high quality user interfaces, it is necessary to engage their enthusiasm for human factors so that they make conscious, informed decisions that they are prepared to change.

Two conclusions were reached:

1. The Unix-derived 'toolkit' approach to systems development allows all important decisions to be concentrated with a core design team. The approach suggests that 90% of the effort should go on building tools that will are then used to construct the required system. The difficulty with this is that it can be hard to identify tools that are sufficiently generic for the purpose, without them becoming so expensive to develop that the wrong choice of tools causes a major disaster.

2. There are three useful models that can improve how human factors are employed in the design process. Firstly a model of project management can determine the place and importance of HCI in design. Secondly, a model of how people adapt to change can assist in introducing a system to its final users. Finally, a model of the design decision making process can improve the flexibility of design particularly by helping with the selection of commitment points.

The second part of the work group's discussion attempted to identify how user models could be used to help designers make decisions. In the absence of a satisfactory model of the design process, specific examples of design issues were considered. A list of eight issues was suggested as being suitable although there was some concern over whether or not they were representative. These were:

1. Choice of interfaces styles for a help system
2. Grouping system functions together, especially for deciding which functions should appear on which screens.
3. When and how to update representations of state.
4. Details of dialogue design
5. Choosing appropriate action sequences for tasks.
6. Choice of terms to be used in the system.
7. Organisation of elements in a menu.
8. Which of two or more precedents to follow when a new system is expected to be compatible with existing ones.

There was only sufficient time to consider the first issue in detail and to briefly examine the second.

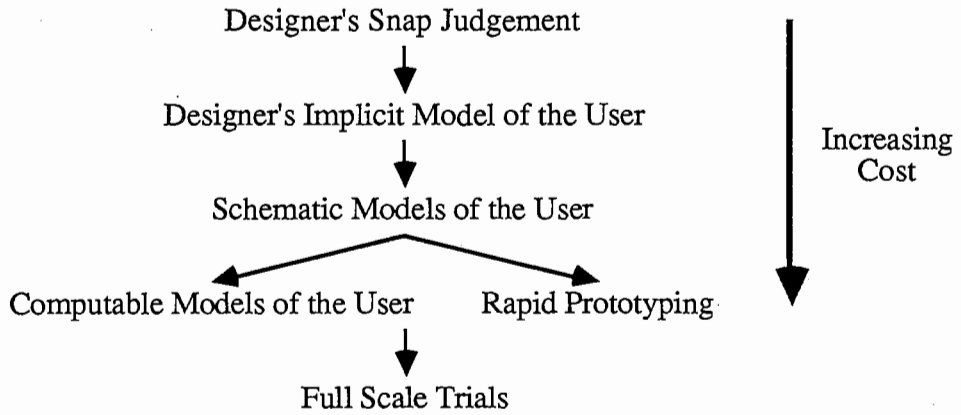
The problem tackled was part of the design of the user interface to a small help system. Some description of the problem is necessary:

> Logica Cambridge is building a knowledge-based help system for the Sagesoft Accountant book-keeping package. The abstract interface is a dialogue with the user asking questions and the system answering until the user has solved his/her problem.[1] The design problem is how to allow the user to ask questions given that the system has limited expressive power. In particular, how to decide between the following:
>
> 1. Allow the user to select a question type and verb and then choose a question object from a restricted list.
> 2. Allow the user to select a topic of interest and then choose from a list of questions that the system can answer about that object.
>
> > The objects in the domain of interest are such things as ledgers, bank accounts and balance sheets, and typical questions are How do I calculate...?, What happens...? and Why is it necessary...?

1 A better interface might involve mixed-initiative dialogue. This was considered to require more resources than were available in the project. Other possibilities based on database retrieval interfaces were suggested by the work group.

The work group made the following points:

- An important aspect of the decision making process is the cost trade-off – how much effort can be put into each decision. A range of possibilities exist as shown in the diagram:



- Flexibility is important. In this case, it might be sensible to have both possibilities. The Logica team had rejected this for on the grounds that it would be confusing to the user population (shop-keepers and other small business people) and because it would be more expensive.
- The best advice to the designer in this case is to use cues relevant to what the user is thinking about. This suggests that the second option is preferable because the user is likely to be thinking about the objects involved in the problem. (This is in fact what the Logica team thought.)
- More general advice to designers is to support 'good modes of action'. For example, in this case the dialogue should allow repair, that is, if the user asks the wrong question, he/she should be able to ask the right one with the minimum of effort.

The overall conclusion of the work group was that designers need models of users that allow them to determine which styles of action are suitable for a given task. Three classes of model are required:

1. A model of cognitive processes.
2. A statistical model of users' cognitive abilities
3. A model that describing 'good things' to support in an interface in terms of cognition. These include support for rescue, repair, hill climbing (ie solving problems incrementally without long distance planning) and GUEPS. This model should enable a designer to judge when such a facility is important.

A final comment was made during the reporting back from the work groups. All the design problems raised are quite 'late' in the development process. However, the help system problem was the first conscious, explicit decision the design team was aware of.

Group 2 - Designers Problems

Present: John Long, Keith Dickerson, Alan Dix, Ian Clowes, Kee-Yong Lim, Claire O'Malley, Andy Whitefield, Tony Simon, Angela Sasse.

This group split up into three smaller groups with one designer describing design problems and posing questions to the model builders.

The first problem was how to design icons so that they would be recognizable and learnable. It was felt that there were currently no models that were applicable to this problem only guidelines and some findings. However it was felt that TAG may help in the design of the command set.

The second problem was concerned with how to structure menus. The designer wanted help with the specification and structure of menus and the naming of menu choices. It was felt that not much help could be provided in this task but some attempt was made to come up with the requirements for a model that could be of use. It was felt that what was required was a structure mapping engine that would take as inputs the design constraints and what the user already knows and come up with a mapping across to the new domain.This would then help in deciding what items to have on the menu and how to structure them.

The third problem was concerned with a problem of editing and how to differentiate between a screen line and a program line. This is seen as a problem of overlapping explanations where it is possible for the user to generate two explanations for a phenomena. The system need to signal where one paradigm ends and another begins and avoid problems of overlapping conceptual models.

Conclusions

The following conclusions can be drawn from the workshop:

- Current user models are insufficient for the purpose. They provide input to the design as it is nearing completion. This is just the point when user testing can be used effectively. Therefore new models need to be developed which focus on the early stages of the design process where they can make a significant impact.
- User models can be used to structure the design process but little attention has been paid to this area except for the simplistic structuring provided by the linguistic models of Moran and Foley.
- Building user models should not be a single activity performed as the design is nearing completion. User models should be started during the requirements stage and be refined and verified as the project progresses.
- Currently a large amount of effort needs to be invested in building models and the main pay-back comes at the end of this process and is difficult to quantify. The effort/pay-back curve needs to be more linear so that user models provide some benefit for reduced effort. This will allow models to be used on smaller projects.
- Some benefits derive from the process of building a user model by focussing attention on understanding end-users. These benefits can be important even if the model is never used predictively.
- User models need to represent aspects of users that are directly relevant to the design process. Currently they focus on what can be modelled easily ie. aspects that are well understood by psychologists. Further work is required to identify the important user characteristics that need to be considered in the design and how to model them.
- Little validation has been carried out on user models to demonstrate their predictive power. Further validation is required before models become accepted.

In summary the workshop provided a useful forum for discussion and some of the issues of concern were aired. However it was felt that this is just the start of the process and further work is required. Some of the papers were not directly relevant to the aims of the workshop and in future we need more people talking about the main topic and the question that is being addressed. In particular we need to examine the construction and use of models and how the model building process fits into the design process. We also need more designers to attend and model builders who are interested in the designers requirements.

It is currently planned that a further workshop will be held on this topic which will take the exploration of these issues further. This will involve more interface designers and more model builders to further develop the requirements and increase designers understanding of what modelling techniques are already available.

Recommendations

The following recommendations come from the workshop:

- There are currently no explicit user modelling techniques that can be recommended for use in a commercial project.
- There are a number of modelling techniques that should be explored further in research projects to validate them and assess their benefits. Case studies should be prepared which explore the use of these techniques.
- Further research work is required to develop user models that can be used by designers which meet the following requirements:
    - Can be used throughout the design process.
    - Provide results that are relevant to the different stages of design and are useful to the designer.
    - The designers can provide the inputs needed by the modelling technique.
- Tools are required to support the development of user models in a design environment and to feed these through as constraints and requirements on the design and also to be used as executable specifications that can be used to test early versions of the design.

Workshop Participants

The following people attended the workshop:

Lisanne Bainbridge	University College London
Nigel Bevan	National Physical Laboratory
Ian Clowes	Logica Cambridge Limited
Keith Dickerson	British Telecom Research Laboratories
Alan Dix	York University
Stephen Draper	University of Glasgow
Kee-Yong Lim	University College London
John Long	University College London
Angela Lucas	Logica Cambridge Limited
Vic Maller	V M Associates
Claire O'Malley	IET, Open University
Steve Payne	Lancaster University
Angela Sasse	Birmingham University
Tony Simon	MRC Applied Psychology Unit
Alistair Sutcliffe	City University
Harold Thimbleby	York University
Andy Whitefield	University College London
Mike Wilson	Rutherford Appleton Laboratories
Peter Windsor	Logica Cambridge Limited