Vive la difference! Individualised interaction with users

Judy Kay
Basser Department of Computer Science
University of Sydney
Australia 2006
judy@cs.su.oz.au

Abstract

To build systems that can deal with the individual differences of users, we need to represent a model of the user. Such a user model can support interpretation of user actions and drive the system's actions.

This paper takes considerable care in defining the various notions of 'user model'. From this, we characterise the promise of user modelling in terms of the individualisation it offers. Then we describe the challenges and benefits user modelling offers as a domain for research in many fields of Artificial Intelligence. Firstly we do this in terms of the major subtasks that are critical to user modelling and then in the context of an example system that filters and customises multi-media objects.

1. Introduction - what are user models?

User models are important because they represent information about the user so that a system can operate more effectively. User modelling has been studied in many different communities such as: knowledge representation; planning; natural language understanding and generation; intelligent teaching systems; intelligent information retrieval. This paper explains the main ideas in this interdisciplinary research field and argues that AI researchers can find an exciting source of both problems and solutions in building user models.

Because individuals have different knowledge, preferences and goals, there are many situations where customised treatment of the user, based on information in a user model, should offer advantages. There are also many challenging problems in achieving such customisation.

Before discussing the various elements of user modelling, we define the two main types of user models. We will use the model of interaction shown in Figure 1, derived from that of Suchman (1987). The leftmost part of the figure shows the user's long term knowledge, beliefs, preferences as well as current goals, plans and tasks and whatever else may be relevant to the current interaction. They bring these to their interaction with a machine. The *user model* is a set of beliefs about the user. In the case of human-machine interaction, this model is stored in the machine and represents a modest attempt to model salient aspects of the user (Self, 1990). In association with this set of beliefs there must be a set of processes that manage them: adding new information as it becomes available, dealing with inconsistencies and noise and inferring new information where this is possible.

To refine this notion, we return to Figure 1. Left of the dotted line is the user. This part of the figure is divided into two areas: $user_{private}$ and $user_{shared}$. The critical distinction is that private aspects are not accessible to the machine, where shared ones are. So, for example, $user_{shared}$ includes the commands that the user has typed and other user actions afforded by the interface. The amount of information in $user_{private}$ is huge compared to this and typically includes aspects like intentions which are difficult for the machine to determine.

The middle part of the figure represents the machine's side of the interaction. We show the distinction between those aspects accessible to the user, $machine_{shared}$ and those that are not, $machine_{private}$. Once again, $machine_{shared}$ also called the *system image*, consists of just those aspects of the system that have been presented at the interface, the actions of the system. Here too, $machine_{private}$ is very large compared to what is available to the user. The bulk of the system's reasoning and knowledge is typically limited to $machine_{private}$.

As described so far, the figure is faithful to Suchman's original model. At the right we show a new element, the 'programmer' that created the machine's various models of knowledge. It may be that the 'programmer' is, indeed, a person who has encoded the information for the machine. Alternately, it may be a program, possibly on a remote machine, that has been responsible for creating the machine's set of beliefs about the domain and user. We will discuss the processes involved in this in Section 3.

To refine the notion of a user model, we now focus on $machine_{private}$. We introduce three different classes of models as shown in the figure. Many user modelling systems actually intertwine these classes of models. We maintain the distinction as the problems and demands of each are somewhat different. It also helps in highlighting the common elements in different user modelling work.

We now describe the three classes of models in terms of an example: a system that gives advice about Unix like UC (Wilensky at al, 1985).

At the bottom is M_{domain}, the expertise model in the domain. For our example, this is expertise about Unix.

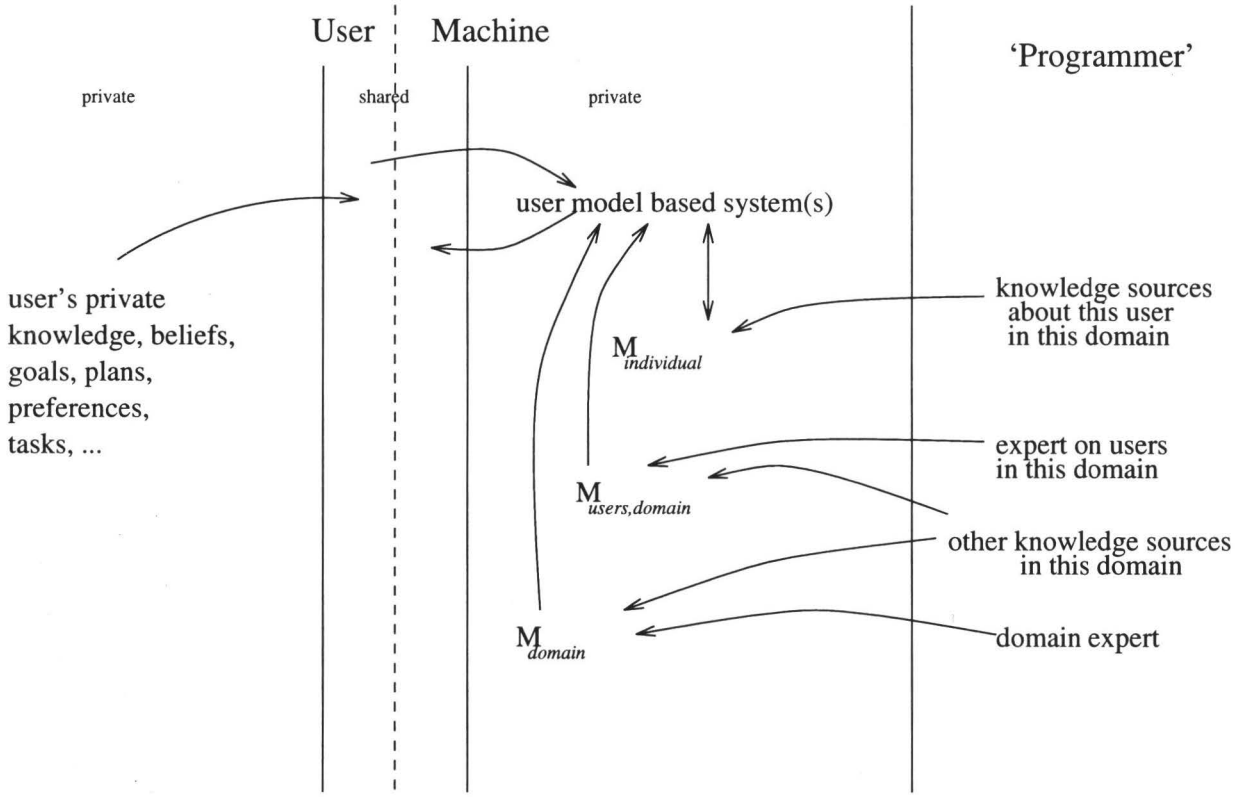

Figure 1: Interaction defining user models and user modelling

This codifies the essential concepts in the domain and the relationships between them. In addition, it models goals and the Unix actions that are the primitives of the plans for achieving these goals. This capacity to create and evaluate various plans for achieving a goal is also part of the domain expertise. The next class of model is $M_{users,domain}$, expertise about typical users in this domain. Commonly this categorises users into expertise groups like beginner, intermediate and guru. It can also classify the domain knowledge correspondingly so that each aspect is categorised according to whether it is likely to be known by beginners, by intermediate users or only by gurus. Note that it is natural for this knowledge of users to be tightly coupled to M_{domain}, the domain model. For example, the actions represented in M_{domain} can be associated with particular classes of users in $M_{users,domain}$.

The third aspect of the machine's knowledge shown in the figure is $M_{individual}$, the model of a particular user. In our example domain, this would classify the user, perhaps as a novice. It would also represent additional information about this user: for example, they might seem to know some things that the 'typical novice' is represented as not knowing. In addition to such long term aspects as the user's knowledge, the machine may model the user's current goals and their progress toward them. Also, it is typically at this level that we need to cope well with conflicting and noisy information about the user as well as temporal change.

The term 'user model' is widely (and confusingly) used for both the expertise about general users in this domain, $M_{users,domain}$ and the model of the individual, $M_{individual}$. The research into $M_{users,domain}$ has been characterised by two main schools: one aims for cognitive validity and the other is ad-hoc and pragmatic. In modelling Unix users, the former would model user's understanding of Unix in ways that match the way those people actually think when they reason about tasks in that domain. So the guru model might be modelled in terms of quite different processes as well as concepts from the novice's. Such cognitive modelling has been used by Doane et al (1992) with three quite different forms of model for each of the three $M_{users,domain}$: novice, intermediate, expert. These reflect the different ways that users of varying expertise reason about the domain as well as their differing levels of knowledge of individual commands.

In terms of Figure 1, the cognitive models for $M_{users,domain}$ claim some correspondence with the actual way that $user_{private}$ operates where pragmatic models are justified in terms of the empirically assessed effectiveness of the systems that use them.

One of the widespread approaches for pragmatic

representations of $M_{users,domain}$ is the *stereotype*. This was introduced by Rich (1983, 1989) to capture the type of default reasoning that people commonly do when classifying people into groups with much in common. When we discussed the three classes of Unix users, they constitute stereotypes which allow statistically useful initial assumptions about a user. It is to be expected that some (perhaps many) of these assumptions will prove to be invalid for any particular person in the class. The power of stereotypes is that they enable the system to take small amounts of information about the user and deduce much more from it quickly and cheaply. We will revisit stereotypes in sections 3 and 4.

The last aspect of the $machine_{private}$ is the user model based system that is driving the interaction. It has to manage the various models beneath it to achieve customised interaction as described in the next section.

To this point, the examples are from a domain dominated by representation of user's knowledge and goals since these are the usual concerns for a system like a Unix advisor. Consider another, very different example: a system that produces a customised hypertext 'newspaper'. In this case, M_{domain} could well represent many different aspects of the domain. For example, in a system that takes news items, selecting those of interest to the user, each news item may be modelled in terms of its content. The $M_{users,domain}$ can model the news preferences typical of various groups of users: for example, one for the sport addict, the financial investor, the wine buff and the like. In existing newspapers, we can view the various subsections (business, sport, international news, entertainment ...) as reflecting the stereotype for the readership of that section. The model of an individual, $M_{individual}$, might be in terms of a collection of these in addition to particular interests outside this set. It may also hold short term goals that affect interests. So, $M_{individual}$ is reflected in the sections of a newspaper that interest the reader on any one day. For example, the person who is about to buy a house may develop strong but short term concerns about interest rates, taxation policy changes and property price trends.

2. Roles for user modelling in intelligent systems

From the machine's point of view, interaction with users involves two main tasks: $user_{shared}$ -> $machine_{private}$, the machine's interpretation of the user's actions at the interface and $machine_{private}$ -> $machine_{shared}$, the generation of machine actions. Note that the latter incorporates the actions the system undertakes on behalf of the user, as well as the interface manifestations of them.

User modelling has the potential to assist in both of these. For example, in a natural language based interface, both these tasks can be quite difficult. It is not surprising that there is a considerable body of work where user models support both these transformations. See, for example, much of the work reported in Kobsa and Wahlster (1989) and more recently in Sarner and Carberry (1992). In interpreting the user's actions at the interface, the user model can reduce ambiguities and constrain the problems of making sense of the user's utterances.

For natural language generation, in a system like Zukerman and McConachy (1993) the user model enables the system to generate more concise discourse by omitting information the user already knows. In addition, the model enables the discourse planner to deal with incorrect inferences the user is likely to make. In Paris (1988), it drives the tailoring of descriptions so that they are comprehensible and helpful for users at various levels of expertise.

Another impressive demonstration of the potential of user modelling for interfaces is in the work of Goodman and Litman (1992). They explored the way that domain constraints affect plan recognition and at the same time demonstrated how a user model containing plans can support advice generation, task completion, context-sensitive responses, error detection and recovery in a CAD interface. We may expect that the leverage gained from user modelling can be significantly attributed to the complexity of the task of design and hence, the complexity of the total interaction with the CAD interface. With increasing machine power, we can expect more systems to support such complex processes as design and we can correspondingly expect that user modelling will be invaluable in making such systems effective.

Also, there has been very interesting work in managing both the structure and content of multi-modal interfaces by Wahlster (Wahlster 1991; Wahlster et al 1993).

Intelligent information retrieval is natural area for exploiting the power of user modelling. For example, Brajnik et al (1990) support the user in the normally difficult task of formulating queries.

3. Roles for AI in user modelling

There are many roles for AI techniques in support of user modelling. We use Figure 1 to illustrate these. Firstly, the various models in $machine_{private}$ must be represented. The special needs of user modelling make this challenging and interesting.

Knowledge representation

One of the most obvious demands of the knowledge representation for most user modelling tasks is that it cope well with noisy, changing, varyingly reliable sources of modelling information. This makes user modelling an excellent testbed for the many approaches for dealing with this type of situation: for example, plausible reasoning, non-monotonic reasoning and belief revision.

It also poses some special problems that are generally

neglected in knowledge representations. Since the user model represents information about a person, it should ensure user control, access and privacy. It is also reasonable to demand accountability from the 'programmer' for the content of the user model.

The particular needs of user modelling are also set by the nature of the human reasoning that is to be modelled. So, for example, Appelt and Pollack (1992) have used an interesting approach to assessing competing explanations in a manner that seems to be particularly suited to reasoning about mental states.

Beyond this, there are many situations where the model of the user needs to represent misconceptions or alternate frameworks. This is common in modelling learners (Greer and McCalla, 1994) where the recognition and correction of misconceptions may often be as important as modelling the preferred or correct 'knowledge' in the domain.

User modelling is a demanding area for knowledge representation. For example, as London (1992) observes, it is common for planning research to be based upon assumptions like: a closed world; the user having correct, well formed plans without critical misconceptions; having a consistent goal and plan. These and many other simplifying assumptions are not appropriate where we model the knowledge of a user, especially one who is inexpert and learning. London was able to construct a representation for knowledge, intent and strategic activity and, from this create advice that was very consistent with that of a human expert.

Building $M_{users,domain}$

Now we move to the many processes involved in user modelling. Essentially, these need to construct the user models and update them. Clearly these must be strongly linked to the choice of knowledge representation. Consider the processes depicted as arrows onto $M_{users,domain}$ in Figure 1.

There are two main types of such process. First, we can build from the knowledge of an expert on users in the domain. For example, to build a model of people who are learning a programming language, like Pascal, we can draw upon the considerable psychology and education literature. So, we might use the bug catalogues developed in the work on PROUST (Johnson 1990). More simply, we might just capture the expertise of someone who has had considerable experience in dealing with various users in the domain of the system. This approach requires knowledge elicitation from a domain expert as in construction of a typical knowledge based system.

The second means of constructing the collective models takes advantage of the other methods for knowledge acquisition. One might apply machine learning to a large collection of data about users in the domain to identify stereotypes. This approach seems very natural and promising but has been little explored. One case where it has been used is by Orwant (to appear) where *communities* of users are dynamically identified so that any one user can be associated in varying degrees with many different 'stereotypical' communities of users.

Building $M_{individual}$

The transition from the 'programmer' to $M_{individual}$ is for the various sources of modelling knowledge that reside outside the current system and can contribute information about this user. These can potentially be extremely varied: they may be conventional databases of information about the user, observations of this user, possibly interpreted by various different programs, or they may even be individuals who know about the user, perhaps the user themself.

An important process could operate via some form of machine learning: there are many and varied sources of information to be exploited, as for example in the work of Orwant (to appear).

The history of the user's dialogue with this system and related ones has the potential to provide much, invaluable input to $M_{individual}$. For a Unix advisor, this would include the dialogue history from interaction with the advisor as well as the user's Unix command use history. Mining this type of data is interesting but difficult. It was used by Cook and Kay (1994) to build detailed instances of $M_{individual}$.

The last source of information for $M_{individual}$ is within $machine_{private}$. This is via the system that employs the user model and can contribute to it. In particular, this is the source of much status information about the user's current tasks, goals and actions. It is also the point at which the system performs diagnostic tasks. In stereotype based systems especially, the goal of this diagnosis is to assess the right stereotype(s) to apply for this user as well as to record the ways that the user differs from these.

Finally, we note that the user model based system needs to combine the three classes of models: domain, collective and individual user models. This process typically takes the categorisation of the user, for example, as a beginner from $M_{individual}$ and use the appropriate stereotypes from $M_{users,domain}$ along with the other relevant elements to build $M_{individual}$.

4. An example system

The above discussion has been largely couched in terms of the general application of user modelling. We now describe a current project that employs user modelling extensively at several levels to support customisation and filtering (Kay and Kummerfeld, 1994). The current focus of the work is to produce a customised hypermedia 'newspaper'. The prototype was based on audio news and it

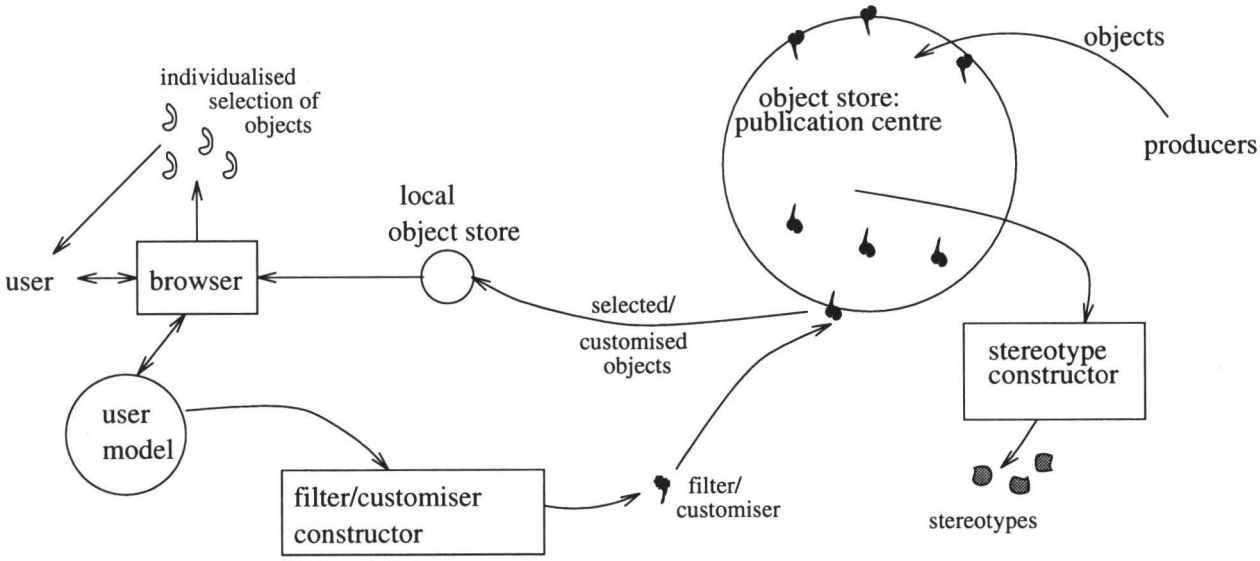

Figure 2: Architecture for a filtering/customisation system.

filtered news items so that those most likely to interest the user were offered in the 'newspaper' for the user to listen to. We have also been working with hypertext with customisation at various levels: the form and content of the actual hypertext pages and the links between them. To achieve this, the hypertext is actually represented as a 'meta-hypertext', a form ready to be customised. Once the user model is applied to this, the actual hypertext is generated for the user.

The architecture supporting this is shown in Figure 2. Circles are data stores, boxes show transformations and the small variously-shaped icons are entities created by the transformation programs. We now describe the elements of this system that are critically determined by user modelling concerns.

First is the user model itself. This is represented as a um (Cook and Kay, 1994;Kay, to appear) based model. A critical aspect of this representation is its support for users to access their model and the values it holds. It can give customised explanations of the aspects modelled M_{domain} as well as justifications for the values from $M_{individual}$.

The other, larger store is the publication centre which holds the objects that are to be selected or customised for many users. It has a collection of news items, as text, audio or video. As shown in the figure, these arrive at the publication centre as they are created by *producers*. Some producers will lodge actual objects while others may send meta-objects. The simplest of these are descriptors that add value to the store. For example, they may be a very knowledgeable individual's ratings of the importance or merit of news items or, as we discuss below, from analyses of information about many users. Equally, they may be machine generated from analyses of the news items.

The prime actors in this system are programs that perform the filtering or customisation on behalf of the user. These are shown as virus-like attachments to the object store. They travel to the store and perform actions on behalf of the user they represent. They filter some classes of objects, sending these on to the user's local object store. From here, the user interacts with a browsing interface.

The user model is critical at all stages. The filter or customiser has two main parts: the method for filtering (or customising) and parts of $M_{individual}$ that will control this. Since the user's interests and knowledge change over time, the filter or customiser needs to be generated repeatedly. The methods of the filter/customiser will change over time, too.

The other element of the system that makes heavy use of $M_{individual}$ is the browser interface. This facilitates the user's exploration of the local object store. It also participates in additional filtering and customisation with parts of the user model not in the filtering or customising programs that were exported to the non-local object store. (These would be aspects the user considered to be sensitive, to be restricted to the local system only.)

The last user modelling element of the system is depicted at the right. This is the stereotype constructor which analyse the user model data of the large numbers of filtering/customising programs at the publication centre. The aim here is that groups of similar users be identified. It also aims to identify opinion leaders, individuals whose preferences and interests are valuable for predicting many other users' preferences. This is an aspect that will become of considerable importance in the exploding field of support tools for filtering, retrieving and customising objects for individuals.

5. Summary

The current reductionist model of computing research has meant that user modelling work has been highly fragmented and results that should be relevant for and from user modelling are not being transferred.

As the reader should have realised, user modelling needs to build on foundations from many of the major areas of AI: knowledge representations; automated reasoning; planning, including multi-agent planning; machine learning from its various schools; the various means for increasing the bandwidth between the user and the machine, especially natural language but also various aspects of perception and, of course, cognitive modelling. Nor should the benefits flow one way. The particular needs of user modelling can offer constraints that define a special flavour for the tasks.

Moreover, user modelling will demand greater cooperation between these subcommunities if we are to exploit individualisation of interaction with users. At present the literatures of the various subgroups of AI are remarkably disjoint.

One important exception to this is in the intelligent teaching systems community where there is a long tradition of user modelling (though typically under the title, student modelling, but continuing the use of this term for both $M_{users,domain}$ and $M_{individual}$). This community has actively explored and *applied* both cognitive and pragmatic modelling for building $M_{users,domain}$. For example, Anderson et al (1990) use a cognitive modelling framework as the basis for a series of teaching systems that have been demonstrated to give impressive improvements in learning.

The student modellers have also used machine learning for building both $M_{users,domain}$ and $M_{individual}$. They have explored many, many approaches to representation of domains and user models. They have built systems embodying their ideas and commonly evaluated them.

It seems intuitively obvious that we deal well with another person when we treat them as an individual, customising our interpretation of their actions as well as generation of our own. It is easy to make assumptions about human-machine interaction based upon such intuitions. But they may not be sound. For example, Brennan (1991) performed an elegant experiment which suggests that when people think they are interacting with machines they use natural language differently from when they believe they are interacting with a person. Moreover, significant parts of the HCI community would contend that if we build our systems well - founded on a single, solid $M_{users,domain}$ - we would not need M_{user} or associated customisation. Certainly, human-machine interaction is not symmetric: the two partners play different roles and operate differently. Our interaction model of Figure 1 highlights some aspects of this. In spite of these reservation which we should keep in mind, we consider that there are many systems that should perform more intelligently and effectively with user modelling.

In particular, we are poised at the beginning of an era when vast resources of information and knowledge will be available to users mediated by machines. The magnitude and variety of those resources pose problems with users being overwhelmed by the diversity and choice available. This is the type of domain where there will be benefits from user modelling in various roles such as those we are exploring in the approach of Figure 2.

Users are different. User modelling is the basis for a system to respond to those differences. We have offered a view of user modelling as a domain where many AI techniques are needed and where the particular demands and constraints of user modelling can be useful for many of the tools sought in AI.

Acknowledgements

My thanks to Alan Fekete, Bob Kummerfeld and Richard Thomas for kindly reading drafts of this paper and making many helpful suggestions that have undoubtably improved the lot of readers who have made it through to this point.

References

[Anderson *et al.*, 1990] J R Anderson, C F Boyle, A T Corbett and M W Lewis, Cognitive modeling and intelligent tutoring. *Artificial Intelligence*, 42(1):7-49, 1990.

[Appelt and Pollack, 1992] D E Appelt and M E Pollack, Weighted abduction for plan ascription. *User Modeling and User-Adapted Interaction*, Kluwer. 2(1):1-25, 1992.

[Brajnik *et al.*, 1990] G Brajnik, G Guida and C Tasso, User modeling in expert man-machine interfaces: a case study in intelligent information retrieval. *IEEE Trans on Systems, Man and Cybernetics*, 20(1):166-185, 1990.

[Brennan, 1991] S E Brennan, Conversation with and through computers. *User Modeling and User-Adapted Interaction*, Kluwer. 1(1):67-86, 1991.

[Cook and Kay, 1994] R Cook and J Kay, The justified user model: a viewable, explained user model. *UM94: User Modeling Conference*, Hyannis, Cape Cod, USA, pages 145-150, 1994.

[Doane *et al.*, 1992] S M Doane, S M Mannes, W Kintsch and P G Polson, Modeling user action planning: a comprehension based approach. *User Modeling and User-Adapted Interaction*, Kluwer. 2(3):249-285, 1992.

[Goodman and Litman, 1992] B A Goodman and D J Litman, On the interaction between plan recognition and intelligent interfaces. *User Modeling and User-Adapted Interaction*, Kluwer. 2:83-116, 1992.

[Greer and McCalla, 1994] J E Greer and G I McCalla (eds), *Student Modelling: the Key to Individualized Knowledge-Based instruction*, Springer-Verlag, Berlin, 1994.

[Johnson, 1990] W L Johnson, Understanding and debugging novice programs, *Artificial intelligence*. 42(1):51-97, 1990.

[Kay and Kummerfeld, 1994] J Kay and R J Kummerfeld, Customization and Delivery of Multimedia Information. *Proc Multicomm 94*, Vancouver, pages 141-150, 1994. also http://www.cs.su.oz.au/~bob/CandD.html.

[Kay, to appear] J Kay, The um toolkit for cooperative user modelling. *User Modeling and User-Adapted Interaction*, Kluwer. to appear.

[Kobsa, 1993] A Kobsa, User modeling: recent work, prospects and hazards. in M Schneider-Hufschmidt, T Kuhme, U Malinowski (eds), *Adaptive user interfaces - principles and practice*, North Holland. pages 111-128, 1993.

[Kobsa and Wahlster, 1989] A Kobsa and W Wahlster (eds), *User models in dialog systems*. Springer-Verlag, Berlin, 1989.

[London, 1992] R V London, Student modeling to support multiple instructional approaches. *User Modeling and User-Adapted Interaction*, Kluwer. 2:117-154, 1992.

[Orwant, to appear] J Orwant, Heterogenous learning in the Doppelganger user modeling system. *User Modeling and User-Adapted Interaction*, Kluwer. to appear.

[Paris, 1988] C L Paris, Tailoring Object Descriptions to a User's Level of Expertise, *Computational Linguistics*, Assoc for Computational Linguistics. 14(3):64-78, 1988.

[Rich, 1983] E Rich, Users are individuals: individualizing user models *Intl J of Man-Machine Studies*, 18:199-214, 1983.

[Rich, 1989] E Rich, Stereotypes and user modeling. in A Kobsa and W Wahlster (eds), *User models in dialog systems*, Springer-Verlag, Berlin, 35-51, 1989.

[Sarner and Carberry 1992] M Sarner and S Carberry, Generating tailored definitions using a multifaceted user model. 2(3):181-210, *User Modeling and User-Adapted Interaction*, Kluwer, 1992.

[Self, 1990] J Self, Bypassing the intractible problem of student modelling. in C Frasson, G Gauthier, *Intelligent Tutoring Systems: at the crossroads of artificial intelligence and education*, Ablex, Norwood, NJ. pages 107-123, 1990.

[Suchman, 1987] L A Suchman. *Plans and situated actions: the problem of human-machine communication*, Cambridge University Press, Cambridge, 1987.

[Wahlster, 1991] W Wahlster, User and discourse models for multimodal communication, in J W Sullivan and S W Tyler (eds), *Intelligent user Interfaces*, Addison Wesley, Reading. pages 45-67, 1991.

[Wahlster, *et al.*, 1993] W Wahlster, E Andre, S Bandyopadhyay, W Graf, and Th Rist, WIP: the ccordinated generation of multimodal presentations from a common representation. in A Ortony, J Slack, O Stocks (eds), *Communication from an artifical intelligence perspective: theoretical and applied issues*, Springer, Heidelberg, pages 121-133, 1993.

[Wilensky *et al.*, 1985] R Wilensky, D N Chin, M Lurai, J Martin, J Mayfield and D Wu. The Berkeley UNIX consultant project. *Computational Linguistics*, 14(4):35-84, 1988.

[Zukerman and McConachy, 1993] I Zukerman and R McConachy. Generating concise discourse that addresses a user's inferences. In Proc 13th Intl Joint Conf on Artificial Intelligence, pages 1202-1207, Chambery, France, 1993. Intl Joint Ctee on Artificial Intelligence.