A General User Modelling Facility

*Robert Kass**

Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104-6389
(kass@linc.cis.upenn.edu)

Tim Finin

Paoli Research Center
Unisys Corporation
P. O. Box 517, Paoli, PA 19301
(finin@prc.unisys.com)

Abstract

An important component of adaptable interactive systems is the ability to model the system's users. Previous systems have relied on user models tailored to the particular needs of that system alone. This paper presents the notion of a *general user model*, and describes some of our research on building a general user modelling facility that could be used by a variety of applications. This work focuses on the representation, maintenance, and acquisition issues of modelling long-term beliefs of the user, and describes a general facility for accomplishing these tasks.

Keywords: User modelling, model acquisition, default reasoning, stereotype, cooperative behavior.

1 Introduction

User modelling is important to many systems that attempt to adapt their behavior to users in order to interact more intelligently. This modelling may be static, involving the design criteria for a good interface, or it may involve using a model dynamically to direct system behavior in an interaction. This paper discusses only the second aspect of user modelling, describing our on-going research on the feasibility and effectiveness of *general user models*: models that have a well-defined set of capabilities that can be used in diverse situations and systems. To this end, the remainder of this introduction discusses when user models are needed and describes the characteristics of an "ideal" general user modelling facility. Sections 2 and 3 present work we have done on the issues of user model maintenance and acquisition—focusing on models of the user's beliefs. Section 4 discusses how stereotypes, and the classification techniques of section 2, can be integrated with the implicit model acquisition techniques of section 3, describing a particular problem of arbitrating between conflicting beliefs about the user in such a system.

*This work was supported by DARPA grant N00014-85-K-0018, and grants from the Army Research Office and the Digital Equipment Corporation

The Importance of User Modelling

User models are not needed for all man-machine interactions, or even all intelligent interfaces. User models are likely to be beneficial if a system has one or more of the following characteristics [15]:

- The system seeks to *adapt* its behavior to individual users;
- The system assumes *responsibility* (or shares responsibility with the user) for ensuring the success of user-system communication;
- The class of potential system users, or the potential system uses, is *diverse*.

In each case, an interactive system must reason about the user's beliefs, goals and plans, preferences and attitudes, or capabilities to understand the actions taken by the user, and to control the system's own behavior.

Many types of systems can benefit from a user modelling capability—in fact, this capability has been exercised in a range of systems. For example, the importance of modelling student knowledge and plans has been recognized as an important aspect of intelligent computer-aided instruction (ICAI) [11]. Student models have been incorporated into many ICAI systems, such as DEBUGGY [4], LMS [23], and tutors for high school geometry and Lisp programming [2]. User models have also been used in intelligent help systems [6, 25, 26], and in the explanation facilities of expert systems [24, 22, 9].

Systems that can benefit from a user modelling facility need not be classified by a particular form of interaction. Any system that strives to be *cooperative* can benefit from a user model. In this area, user models have been useful to identify potential obstacles in a user's plan [1], recognizing when a user's query does not reflect the user's underlying goals [20], tailoring responses according to the user's perspective [18] or knowledge [19], or correcting user misconceptions [17].

The Ideal General User Model

Although user models have been employed in many types of interactive systems, the models have been specifically crafted for each application. This hand-crafting is costly, since building a user modelling facility requires a substantial amount of effort. We have been exploring some of the characteristics desirable in an *ideal* general user modelling facility, but here will only touch on two aspects: dimensions for measuring the generality of a user modelling system, and the facilities that any general user modelling system should have. A longer discussion can be found in [12].

Is it possible to produce a general user modelling facility so multiple systems can benefit from a single design effort? Our ultimate answer is yes, general user modelling is practical. Although the ideal system described here may not be realized, significant features of the ideal model can be used effectively.

Dimensions of Generality. User models may be general with respect to three dimensions: the range of users, the forms of interaction, and the underlying system domain. *User generality* is usually a requirement of any user modelling facility, since user models are normally used (and most beneficial) when a range of users deal with the system, or when the system strives to adapt its behavior to individual users. Thus, interaction and domain generality are the unique features of an ideal general user modelling facility. A user model has *interaction generality* if it can be used with a variety of interaction modes, such as structured interactions or mixed initiative dialogue, and can be used with various modes of communication, such as natural language, menus, speech, and graphics. A *domain general* user modelling facility can be used with applications having a range of knowledge bases, such as diagnostic systems for medicine, mechanical devices, and electronic components.

User Modelling Facilities. A general user modelling system must provide three essential facilities: representation and maintenance facilities for the contents of the model, access facilities for other components of the system or interface, and acquisition facilities for building the model [15]. Our work has focused on the representation and maintenance, and acquisition facilities for general user modelling. Representation and maintenance of information about the user is central to any user modelling activity, while the acquisition of such information has been a major bottleneck to effective user modelling. We have also focused on modelling *long term* user information, such as the beliefs a user holds about the world or about the system domain. Such beliefs tend to persist over time, so the user model formed for an individual can be useful in many separate user-system interactions. The next two sections describe some of our efforts towards building a general user modelling facility.

2 A General User Modelling Shell

This section describes GUMS, a general user modelling shell, designed to experiment with the representation and maintenance issues of general user modelling [7, 5]. GUMS is intended to serve as a utility for a set of application programs (see figure 1). For each application, GUMS keeps a knowledge base of user models relevant to that application. Applications are responsible for acquiring information about the user and supplying it to GUMS to update the user model. Likewise, the application queries GUMS to obtain information about the user, although demons could be used to accomplish a task (such as informing an application) when specific changes occur in the user model.

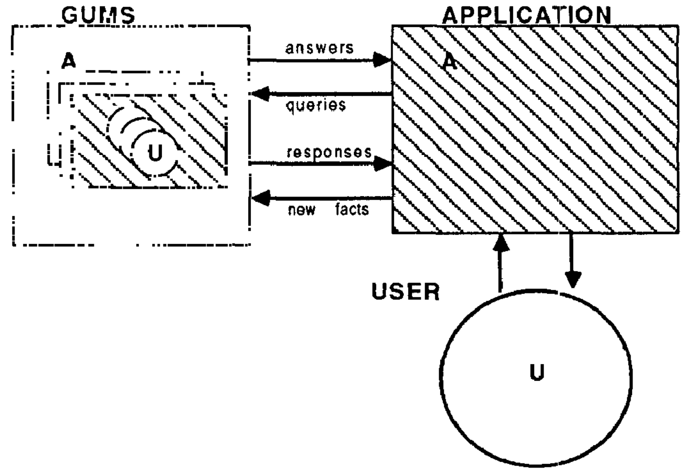

Figure 1: A General Architecture for a User Modelling Utility

Representation User modelling is most useful in situations where a system must draw many plausible conclusions about the user on the basis of a small amount of definite knowledge about him. Thus, *default reasoning* [21] is an appropriate method for representing user model information. GUMS uses three default reasoning techniques to represent its beliefs about user knowledge: stereotypes, explicit default rules, and failure as negation. These three techniques are used to capture generalizations of different grain size, forming a hierarchy with respect to the strength of their conclusions. Stereotypes are used to capture generalizations about large classes of users. Within a stereotype, explicit default rules may be used to express stereotypic norms which might vary for individuals of that class. Failure as negation is the weakest form of default reasoning, used to gather weak evidence for beliefs about the user when stronger methods do not exist.

Stereotypes consist of a set of facts and rules believed to apply to a class of users. These facts and rules may be *definite*, meaning they necessarily apply to all users of that class, or *default*, specifying initial beliefs about users of that class that can be overridden. The definite information in a stereotype forms a sort of definition for the stereotype, by specifying information that must be believed about the user for him to be a member of that class.

Stereotypes can be organized in hierarchies, where one stereotype, S1, subsumes another S2, if everything true in S1 is necessarily true in S2. Thus, a stereotype can inherit information from more general stereotypes in the hierarchy. A model of an individual user is represented as a leaf-node in the hierarchy. Individual user models can have specific information associated with them, in addition to inheriting the facts and rules from subsuming stereotypes in the hierarchy. This information, however, is constrained to be *definite* and *unitary*, i.e. it must consist of definite, fully instantiated facts. Ideally, user models and stereotypes should be able to inherit information from several immediate subsumers, as in a lattice, but this initial implementation limits the hierarchy to a tree.

Maintenance As new information about a user is supplied to GUMS, the individual user model must be updated, potentially creating inconsistencies in the user model. The task of the maintenance facility is to update the individual user model, and restore the consistency if necessary. Some inconsistencies are easy to resolve. If a new, definite fact about the user is asserted, contradicting a default belief, the definite information is believed. Thus, beliefs which depend on default facts inherited from stereotypes, conclusions of default rules, or failure as negation will be overridden by definite facts about the user.

A more difficult conflict to resolve is one between a new, definite fact and a conclusion reached from definite facts and rules inherited from a stereotype. In this case, again, the definite fact asserted about the user will be believed, but steps must be taken to resolve the conflict with the stereotype. Since a definite fact is a defining characteristic of a stereotype, a conflict of this form means the user has been classified incorrectly in the stereotype hierarchy. Thus, this form of conflict requires a *reclassification* of the individual user model.

Reclassification can be either domain dependent or domain independent. Domain dependent methods are useful when reasons for a misclassification are understood. For example, as a user learns about the domain, the appropriate stereotype for representing his knowledge will change. Domain dependent reclassification could use knowledge of the user's expected "growth path" to select a new stereotype. A powerful domain independent reclassification method would be a technique similar to classification of concepts in the KL-ONE family of representation languages [3]. Such a strategy would consider all possible stereotypes and find the set of most specific subsuming stereotypes for the individual user model. GUMS implements a simpler scheme. When a conflict is encountered, the ancestors of the current stereotype are searched in order of specificity (by moving up the tree) until one is found that does not conflict with the individual user model; reverting to the null stereotype—one that makes no assumptions about the user—if necessary.

GUMS enables general user modelling by providing applications with an environment containing a set of user modelling facilities. Applications using this environment take advantage of these facilities, instead of re-creating them. Thus GUMS centralizes control of the access and maintenance of information about the user, similar to the way knowledge base systems and data base systems centralize the control of these functions for knowledge and data.

3 Implicit User Model Acquisition

Our work has also focused on the problem of *acquiring* a model of the user's beliefs [9, 14]. In GUMS, beliefs about the user are acquired in two ways: by explicitly encoding beliefs in stereotype rules and facts, and by assertions about an individual made by the application. Although GUMS can support the user modelling requirements for many applications, a great deal of effort is required to make use of it. Not only must the application designer discover and implement a system of domain related stereotypes, but (since the application is responsible for populating the individual models of users with facts representing their beliefs) he must also design and implement some kind of knowledge acquisition strategy.

The acquisition problem in GUMS, as in most user modelling systems, is the need to explicitly encode large amounts of information about the potential system users. To model user's beliefs about the system's domain, the task of building stereotypes may be more time consuming than building the domain knowledge base itself, because of the large number of stereotypes necessary. Furthermore, explicit model acquisition is error-prone due to the difficulty of classifying users and their likely beliefs.

An alternative to explicit user model acquisition is to build the model implicitly, as the user interacts with the system. Implicit acquisition avoids the explicit encoding bottleneck, and can reduce the burden on the application as well. If the user modelling facility takes responsibility for acquiring the user model, applications do not need to reason about what information should be asserted about the user. Furthermore, if the user modelling facility's acquisition capability is application independent, then general user modelling is a practical method for providing user modelling capabilities to a variety of applications.

Implicit user model acquisition has not been pursued extensively because it has generally been considered to be too slow to build a useful, robust model, and too uncertain in the conclusions it makes. Our research suggests that this need not be the case. In particular, for specific forms of user-system interaction, many basic assumptions about the user's behavior can be made, providing a foundation for drawing many conclusions about the user's beliefs.

User Model Acquisition Rules By focusing on *cooperative advisory systems* (systems that advise the user, and seek to be as helpful as possible) that allow the user to volunteer information, and that communicate in natural language, we have been able to develop a set of *user model acquisition rules*. These rules are domain independent, supporting the feasibility of general user modelling. The rules were inspired by transcripts of over 100 conversations between a human expert and people seeking advice concerning their personal financial investments.[1] The rules capture reasonable methods that an expert might use to draw conclusions about the beliefs of the user. In fact, from a short dialogue between the system and a user, the rules are capable of building a model sufficiently robust to enable it to tailor its explanations to that user.[2]

The implicit acquisition rules rely on basic assumptions about the user and his behavior. For example, one set of rules,[3] the *implicature* rules, assume the user is cooperating with the system, and thus is observing Grice's Maxims for cooperative communication [8]. One of these maxims, the maxim of relation, admonishes a speaker to "be relevant" in what he says. Assuming the user observes this maxim, the *relevancy* rule states:

[1] The transcripts were made by Martha Pollack and Julia Hirschberg from the radio talk show "Harry Gross: Speaking about Your Money" broadcast on station WCAU in Philadelphia, February 1-5, 1982.

[2] Although this capability has not been implemented yet, it has been well studied, and the implementation is now in progress. See [13] or [10] for an extended example of a dialogue where a user model is built, then used to tailor the system's explanation of a recommendation.

[3] See [9] for a complete presentation of the rules.

> *If the user says P, the user modelling module can assume that the user believes that P, in its entirety, is used in reasoning about the current goal or goals of the interaction.*

Thus, in the following dialogue taken from the transcripts, the caller believes that all the information she provides is relevant to determining how to take her supplemental annuity.

C. I just retired December first, and in addition to my pension and social security I have a supplemental annuity which I contributed to while I was employed from the state of New Jersey mutual fund. I'm entitled to a lump sum settlement which would be between $16,800 and $17,800, or a lesser life annuity and the choices of the annuity would be $125.45 per month. That would be the maximum with no beneficiaries.

E. You can stop right there, take your money.

If this dialogue took place between a user and an investment advisory system, the user modelling module would assert that the user believes that each of these items is relevant to the goal of deciding how to take the annuity, even though the system knows some of that information is, in fact, not relevant.

Other acquisition rules make assumptions about reasoning the user is likely to do. In general, one cannot assume that the user will believe all of the logical consequences of his current beliefs (i.e. assume *consequential closure*). Instead, an approximate model of the inferences any user would be likely to draw is needed. Our current model includes, for example, rules to cover the transitivity of subsumption relations *(if the user knows A is a kind of B, and B is a kind of C, then he will infer that A is a kind of C)* and inheritance of properties of concepts.

Another group of rules focuses on assumptions about the user's human behavior. For example, the *agent rule* assumes the user actively performs actions and remembers them. The rule states:

> *If the user is the agent of an action, then the user modelling module can attribute to the user knowledge about the action, the substeps of the action, and the factual information related to the action.*

Using the agent rule, if the user says "I just rolled over two CD's," the user modelling module can recognize that the user is the agent of the "roll over" action, and conclude that the user knows about the steps involved in rolling over a CD. Furthermore, the agent rule will also assert that the user knows about related facts, such as: that CD's have a due date, that money from a CD can be reinvested, that CD's are obtained from banks, and so on. Thus the agent rule can be particularly powerful in contributing information about the user.

In summary, a significant problem in acquiring user models can be overcome through the use of implicit acquisition techniques. The acquisition rules developed in our work are domain independent, thus they enable more practical general user models to be built.

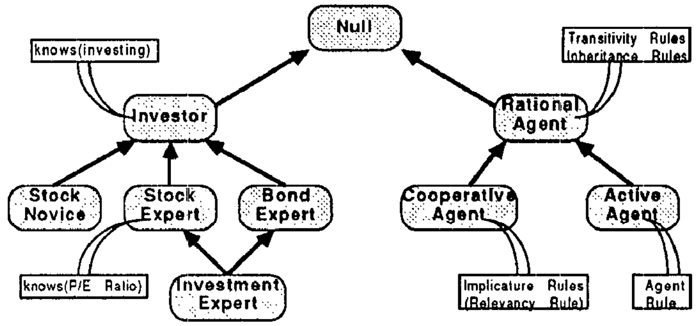

Figure 2: A Hierarchy Containing General and Domain Independent Stereotypes

4 Integrating Stereotypes and Implicit Acquisition

Although the contrasts between explicit and implicit user model acquisition techniques were emphasized in section 3, these methods can complement each other. Despite the problems with explicit acquisition, and the advantages of implicit acquisition methods, it is still desireable to encode domain-specific knowledge about users in many situations. This section explores how the implicit acquisition rules can be integrated with the GUMS framework, resulting in a powerful, extensible general user modelling facility that benefits from both acquisition approachs.

The key to integrating the implicit acquisition rules with GUMS is recognizing that the rules are *default* rules, sanctioned by specific assumptions about the user. Thus, the acquisition rules can be viewed as elements of very general stereotypes. For example, a stereotype for the class of "cooperative agents" would contain the implicature rules as default rules, including the relevancy rule. Other general stereotypes would be "rational agent," containing rules modelling the user's reasoning capabilities, or "active agent," containing the agent rule. These stereotypes form an independent hierarchy in a stereotype lattice, distinct from the domain specific stereotypes. Thus, for an investment advisor system, a hierarchy of stereotypes such as in figure 2 might be used.

With this notion of general stereotypes, it is useful to distinguish between two types of rules a stereotype may contain: rules believed to be used *by* the user in his own reasoning (user inference rules), and rules *about* the user used by the system to make conclusions about him (model acquisition rules). For example, the relevancy rule reasons *about* the user's beliefs, but a transitivity rule, although it can be used to draw conclusions about the user's beliefs, is assumed to be used *by* the user in his own reasoning.

No representational distinction is made between domain independent and domain dependent stereotypes, enabling two additional, potentially powerful user modelling capabilities. First, model acquisition and user inference rules that are domain-specific can be included in stereotypes. One of the problems with building stereotypes is the number of facts that must be included in the stereotype. Domain specific model acquisition rules enable the user modelling facility to infer these facts implicitly, providing a more principled method for attributing beliefs to the user than simple assertions in a stereotype.

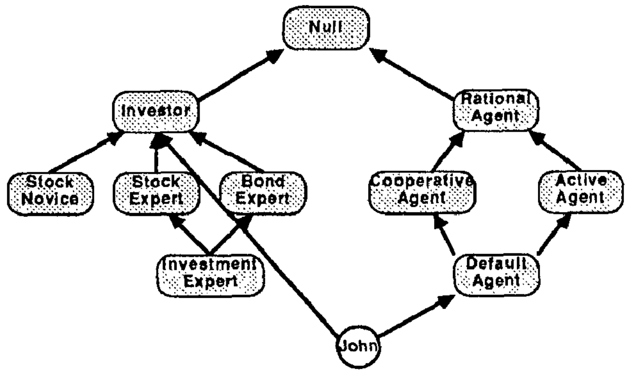

Figure 3: Initial Classification for an Investment Advisory System User

The second capability involves the encoding of explicit facts in general stereotypes. A problem with many interactive systems is their limited knowledge of the world. If a body of common sense knowledge that all users are assumed to know is available in a very general stereotype, it could relieve the brittleness problem of such systems. Such a stereotype of common sense knowledge might be called the *any fool stereotype*, after McCarthy's notion that this is the knowledge "any fool" would know [16].

Integration Problems

Some problems arise when integrating implicit acquisition methods with the stereotype hierarchy of GUMS, however. Two that we have encountered are discussed here, with our thoughts on potential solutions for the problems.

One problem with very general stereotypes, such as "cooperative agent," is the lack of defining facts about members of that class. In GUMS, the hierarchy of stereotypes was determined by the definite facts contained in the stereotypes, but for a stereotype such as "cooperative agent," useful defining facts are rare. Thus, it is not feasible to wait to discover the fact "cooperative(U)" before classifying U as a cooperative agent.

This difficulty may be avoided by initially classifying the user beneath all of the domain independent stereotypes. Thus, the system would initially assume the user is rational, cooperative, and an active agent, so the defining facts of the stereotype would not be needed in order to use the stereotype. For example, in an investment advisory system, a user might originally be assumed to be a "default agent" and an "investor," as in figure 3. It may be necessary, however, to retract the assumption that a general stereotype applies to a user, such as when the implicature rules consistently make conclusions that are contradicted by other, more certain, beliefs about the user. We do not know of a general method for distinguishing when one or more default rules in a stereotype should be retracted, and when belief in the stereotype as a whole should be retracted.

A second problem involves conflicting beliefs about the user. In GUMS, new information about the user is supplied by the application, and assumed to be definite. Thus, most of the belief conflicts encountered by GUMS are between definite knowledge from the application, and definite or default knowledge assumed by stereotype subsumption. Conflicts between definite and default knowledge are easy to resolve: believe the definite knowledge. In GUMS, a conflict with definite knowledge from a stereotype is also straightforward: belief in the stereotype is dropped, and the hierarchy is traversed upward until a stereotype consistent with the definite knowledge about the user is found.

The simple resolution techniques in GUMS do not extend well to the integrated version of the system. First, the assumption that assertions from the application are definite is a simplification: most assertions made by the implicit acquisition rules will be default conclusions. Second, the stereotype hierarchy in GUMS is a tree, while the integrated system will require a lattice to reflect the inheritance of several, independent sets of assumptions about the user. Thus, the integrated system will have conflicts between stereotypes that are incomparable.

Some of these problems may be handled by applying a heuristic that attempts to maximize the number of assumptions about the user after a conflict is resolved—this heuristic is used implicitly in GUMS. The stereotype hierarchy in GUMS represents an ordering based on the amount of information assumed about the user: stereotypes at the top of the tree contain few assumptions, while those at the bottom inherit the assumptions from higher stereotypes, so they contain more assumptions. When a conflict is encountered that requires a stereotype to be dropped, GUMS traverses up the tree, trying to find the stereotype with the most information that is consistent with the definite beliefs about the user.

In the integrated system, a similar method may be used. When a conflict between stereotypes arises, the sytem should retract the stereotype that leaves the maximum number of assumptions about the user intact. Since several stereotypes may concurrently draw conclusions about some facts, this means the stereotype that makes the least number of unique conclusions about the user should be retracted.

5 Conclusion

General user modelling is an attractive approach to providing interactive systems with information about their users. Our work in this area indicates that general user modelling is not only feasible, but practical. The GUMS system demonstrates how a set of facilities for user modelling can be provided in the scope of an environment for building interactive applications. Furthermore, work on implicit acquisition indicates that the acquisition bottleneck can be overcome in a domain independent manner. Thus, a general user modelling facility for supporting cooperative advisory systems is a practical possibility.

References

[1] J. F. Allen and C. R. Perrault. Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178, 1980.

[2] J. R. Anderson, C. F. Boyle, and B. J. Reiser. Intelligent tutoring systems. *Science*, 228:456–462, 1985.

[3] R. J. Brachman and J. G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9:171–216, 1985.

[4] J. S. Brown and R. R. Burton. Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, 2:155–192, 1978.

[5] T. Finin. GUMS—a general user modelling shell. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, Springer Verlag, Berlin—New York, 1988.

[6] T. Finin. Providing help and advice in task oriented systems. In 8^{th} *International Conference on Artificial Intelligence*, pages 176–178, 1983.

[7] T. Finin and D. Drager. $GUMS_1$: a general user modelling system. In *Proceedings of the 1986 Conference of the Canadian Society for Computational Studies of Intelligence*, pages 24–30, 1986.

[8] H. P. Grice. Logic and conversation. In P. Cole and J. L. Morgan, editors, *Syntax and Semantics*, Academic Press, New York, 1975.

[9] R. Kass. *Implicit Acquisition of User Models in Cooperative Advisory Systems*. Technical Report MS-CIS-87-05, Department of Computer and Information Science, University of Pennsylvania, 1987.

[10] R. Kass. *Modelling User Beliefs for Good Explanation*. Technical Report MS-CIS-87-77, Department of Computer and Information Science, University of Pennsylvania, 1987. Submitted to the 26th Meeting of the Association for Computational Linguistics.

[11] R. Kass. Student modelling in intelligent tutoring systems—implications for user modelling. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, Springer Verlag, Berlin—New York, 1988.

[12] R. Kass and T. Finin. *General User Modelling: A Facility to Support Intelligent Interaction*. Technical Report MS-CIS-87-78, Department of Computer and Information Science, University of Pennsylvania, 1987.

[13] R. Kass and T. Finin. *The Need for User Models in Generating Expert System Explanations*. Technical Report MS-CIS-87-83, Department of Computer and Information Science, University of Pennsylvania, 1987.

[14] R. Kass and T. Finin. Rules for the implicit acquisition of knowledge about the user. In *Proceedings of the* 6^{th} *National Conference on Artificial Intelligence*, pages 295–300, 1987.

[15] R. Kass and T. Finin. User models in intelligent interfaces: why they are needed, problems with their implementation. In B. Neches and T. Kaczmarek, editors, *AAAI-86 Workshop on Intelligent Interfaces*, 1986.

[16] J. McCarthy. Circumscription - a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1):27–40, 1980.

[17] K. F. McCoy. Reasoning on a highlighted user model to respond to misconceptions. *Computational Linguistics*, Special Issue on User Modelling, 1988.

[18] K. R. McKeown. Tailoring explanations for the user. In 9^{th} *International Conference on Artificial Intelligence*, pages 794–798, 1985.

[19] C. L. Paris. Tailoring object descriptions to a user's level of expertise. *Computational Linguistics*, Special Issue on User Modelling, 1988.

[20] M. E. Pollack. *Inferring Domain Plans in Question-Answering*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, 1986.

[21] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1):81–132, 1980.

[22] D. H. Sleeman. UMFE: a user modelling front end subsystem. *International Journal of Man-Machine Studies*, 23:71–88, 1985.

[23] D. H. Sleeman and M. J. Smith. Modelling student's problem solving. *Artificial Intelligence*, 16:171–187, 1981.

[24] J. W. Wallis and E. H. Shortliffe. Customizing explanations using causal knowledge. In B. G. Buchanan and E. H. Shortliffe, editors, *Rule-Based Expert Systems*, Addison-Wesley, Reading, MA, 1984.

[25] R. Wilensky, J. Mayfield, A. Albert, D. N. Chin, C. Cox, M. Luria, J. Martin, and D. Wu. *UC - A Progress Report*. Technical Report UCB/CSD 87/303, Department of EECS, University of California, Berkeley, 1986.

[26] A. Y. Zissos and I. H. Witten. User modelling for a computer coach: a case study. *International Journal of Man-Machine Studies*, 23:729–750, 1985.