

# MODELING THE USER IN NATURAL LANGUAGE SYSTEMS

Robert Kass and Tim Finin

Department of Computer and Information Science/D2  
School of Engineering and Applied Science  
University of Pennsylvania  
Philadelphia, PA 19104

For intelligent interactive systems to communicate with humans in a natural manner, they must have knowledge about the system users. This paper explores the role of *user modeling* in such systems. It begins with a characterization of what a user model is and how it can be used. The types of information that a user model may be required to keep about a user are then identified and discussed. User models themselves can vary greatly depending on the requirements of the situation and the implementation, so several *dimensions* along which they can be classified are presented. Since acquiring the knowledge for a user model is a fundamental problem in user modeling, a section is devoted to this topic. Next, the benefits and costs of implementing a user modeling component for a system are weighed in light of several aspects of the interaction requirements that may be imposed by the system. Finally, the current state of research in user modeling is summarized, and future research topics that must be addressed in order to achieve powerful, general user modeling systems are assessed.

## 1 INTRODUCTION

Systems that use natural language as a means of communication must do so in a natural manner. One of the features of communication between people is that they acquire and use considerable knowledge about their conversational partners. In order for machines to interact with people in a comfortable, natural manner, they too will have to acquire and use knowledge of the people with whom they are interacting.

Early research on natural language interfaces tended to view natural language as a "very high level" query language. One of the important results of research in the latter half of the 1970s (Waltz 1978, Kaplan 1982) is the realization that natural language communication is much more. The use of natural language for communication includes a host of conventions that must be followed in the dialog (Grice 1975). A person interacting with a computer via natural language will assume that these conventions are being followed, and will be quite unsatisfied if they are not. Most of these conventions require, in one way or another, that a conversational participant have particular knowledge about the goals, plans, capabilities, attitudes, and beliefs of the other person.

This paper analyzes the role of user models in systems that interact with individual users in a natural

language. Although the necessity of having and using a model of the user has been seen for some time, only within the last few years has it been actively pursued as a research topic. This research has been driven, in part, by attempts to create natural language interfaces to systems that can be characterized as *cooperative problem solvers*. Examples of such systems include intelligent interfaces to expert systems (Finin et al 1986, Carbonell et al 1983), database systems (Carberry 1985, Webber 1986), intelligent tutoring systems (Kass 1987b), and help and advisory systems (Wilensky et al 1984).

### 1.1 AN OVERVIEW OF THIS PAPER

In the remainder of this section, the kinds of user models and systems to be discussed in this paper will be characterized, including a general definition of a user model and an outline of how it can be used by a cooperative, interactive system that converses in natural language. The next section addresses the question "What is to be modeled?" by looking in some depth at the types of information that might be contained in a user model. These can be broadly classified as the user's goals (and the plans he may use to achieve them), capabilities, attitudes, and knowledge or belief. In section 3 a set of dimensions along which user models can

Copyright 1988 by the Association for Computational Linguistics. Permission to copy without fee all or part of this material is granted provided that the copies are not made for direct commercial advantage and the *CL* reference and this copyright notice are included on the first page. To copy otherwise, or to republish, requires a fee and/or specific permission.

0362-613X/88/0100●●\$03.00

be classified is presented, while section 4 considers the methods that might be used to acquire information of the user, especially of his goals, plans, and beliefs. Section 5 considers several high-level features that have an impact on the design of a user modeling system, such as which participant in the interaction bears responsibility for ensuring the communication, or what the penalty for an error in the user model is. These considerations have an impact on the potential benefits and costs of employing a user model. The concluding section raises some issues that will require additional research in order to produce a powerful, general user modeling system.

## 1.2 WHAT IS A USER MODEL?

Specifying what a user model is is not an easy task. An initial, general definition is presented here, but is then narrowed to focus on explicit, knowledge-based models. The various ways in which these user models can support a cooperative problem solving system are then outlined.

The term "user model" has been used in many different contexts to describe knowledge that is used to support a man-machine interface. An initial definition for "user model" might be the following:

*A user model is the knowledge about the user, either explicitly or implicitly encoded, that is used by the system to improve the interaction.*

This definition is at once too strong and too weak. The definition is too strong in that it limits the range of modeling a natural language system might do to the user of the system only. Many situations require a natural language system to deal with several models concurrently, as will be demonstrated later in this paper. The definition is too weak since it endows every interactive system with some kind of user model, usually of the implicit variety. The following paragraphs clarify these issues, and in so doing restrict the class of models to be considered.

## AGENT MODELS

Imagine a futuristic data base query system: not only do humans communicate with the system to obtain information, but other software systems, or even other computer systems might query the data base as well. The individuals using the data base might be quite diverse. Rather than force all users to conform to interaction requirements imposed by the system, the system strives to communicate with them at their own level. Such a system will need to model both people and machines. A second situation is when a person uses an application such as an advisory system on behalf of another individual; the advisor in this case may be required to concurrently model both individuals.

A useful distinction when discussing situations in which multiple models may be required is one between *agent* models and *user* models. Agent models are models of individual entities, regardless of their relation to

the system doing the modeling, while user models are models of the individuals currently using the system. The class of user models is thus a subclass of the class of agent models. Most of the discussion in this paper applies to the broader class of agent models, however, the term "user model" is well established and hard to avoid. Thus "user model" will be used in the remainder of this paper, even in situations where "agent model" is technically more correct.

## EXPLICIT MODELS

Agent models that encode the knowledge of the agent implicitly are not very interesting. In such systems, the model knowledge really consists of the assumptions about the agent made by the designers of the system. Thus even the FORTRAN compiler can be said to have an implicit agent model.

A more interesting class of models is one in which the information about the agent is *explicitly* encoded, such as models that are designed along the lines of knowledge bases. In the context of agent models, four features of explicitly encoded models are important.

1. *Separate Knowledge Base*: Information about an agent is collected in a separate module rather than distributed throughout the system.
2. *Explicit Representation*: The knowledge in the agent model is encoded in a representation language that is sufficiently expressive. Such a representation language will typically provide a set of inferential services, allowing some of the knowledge of an agent to be implicit, but automatically inferred when needed.
3. *Support for Abstraction*: The modeling system provides ways to describe abstract as well as concrete entities. For example, the system might be able to discuss classes of users and their general properties as well as individuals.
4. *Multiple Use*: Since the user model is explicitly represented as a separate module, it can be used in several different ways (e.g., to support a dialog or to classify a new user). This requires that the knowledge be represented in a more general way that does not favor one use at the expense of another. It is highly desirable to express the knowledge in a way that allows it to be reasoned about as well as reasoned with.

Agent models that have these features fit nicely into current work in the broader field of knowledge representation. In fact, Brian Smith's *knowledge representation hypothesis* (Smith 1982) could be paraphrased to address agent modeling as follows:

*Any agent model will be comprised of structural ingredients that a) we as external observers naturally take to represent a propositional account of the knowledge the system has of the agent and b) independent of such external semantical attribution, play a formal but causal and essential role in the behavior that manifests that knowledge.*

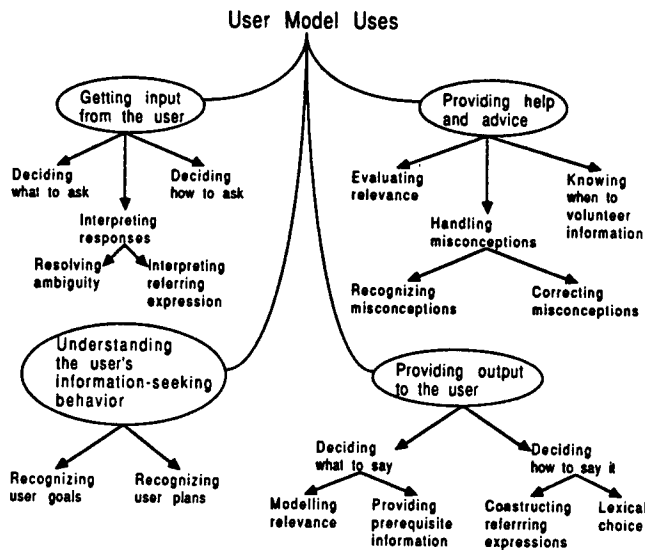


Figure 1. Uses for Knowledge of the User.

### 1.3 HOW USER MODELS CAN BE USED

The knowledge about a user that a model provides can be used in a number of ways in a natural language system. These uses are generally categorized in the taxonomy in Figure 1. At the top level, user models can be used to support (1) the task of recognizing and interpreting the information seeking behavior of a user, (2) providing the user with help and advice, (3) eliciting information from the user, and (4) providing information to him. Situations where user models are used for many of these purposes can be seen in the examples presented throughout this paper.

The characterization of user models remains quite broad to allow consideration of a wide range of factors involved in building user models. These factors provide dimensions upon which the various types of user models can be plotted. Section 3 explores these dimensions to provide a better understanding of the range of user modeling possibilities. Given this range of possible types of user models, methods for their acquisition can be discussed (section 4), along with factors that influence the feasibility and attractiveness of particular types of user models for given applications (section 5). First, however, the types of information a user model should be expected to keep are discussed.

## 2 THE CONTENTS OF A USER MODEL

A primary means of characterizing user models is by the type of knowledge they contain. This knowledge can be classified into four categories: goals and plans, capabilities, attitudes, and knowledge or belief. Each of these categories will be examined in this section to see situations where such knowledge is needed, and examples of how that knowledge is used in natural language systems.

## 2.1 GOALS AND PLANS

The *goal* of a user is some state of affairs he wishes to achieve. A *plan* is some sequence of actions or events that is expected to result in the realization of a particular state of affairs. Thus plans are means for accomplishing goals. Furthermore, each step in a plan has its own subgoal to achieve, which may be realized by yet another subplan of the overall plan. As a result, goals and plans are intimately related to one another, and one can seldom discuss one without discussing the other.

Knowledge of user goals and plans is essential in a natural language system. Individuals participate in a conversation with particular goals they wish to achieve. Examples of such goals are obtaining information, communicating information, causing an action to be performed, and so on. A cooperative participant in a conversation will attempt to discover the goals of other participants in an effort to help those goals to be achieved, if possible.

Recognizing an individual's goal (or goals) may range from being a straightforward task, to one that is very difficult. Situations in which a natural language system must infer goals or plans of user (roughly in order of increasing difficulty) include:

- the user directly states a goal
- the user's goal may be indirectly inferred from the user's utterances
- the user has incorrect or incomplete goals and plans
- the user has multiple goals and plans.

### DIRECT GOALS

In the simplest situations the user may directly state a goal, such as

“How do I get to Twelve Oaks Mall from here?”

The speaker's goal is to obtain information. A hearer is capable of recognizing this goal directly from the question, without further inference.

### INDIRECT GOALS

Unfortunately, people frequently do not state their goals directly. Instead, they may expect the hearer to infer their goal from their utterance. For example, when a speaker says,

“Can you tell me what time it is?”

the hearer readily infers that the questioner wishes to know what the current time is. The inferences required by the hearer may often be rather involved. Gershman looked at this problem with respect to an Automatic Yellow Pages Advisor (AYPA) (Gershman 1981). A sample interaction with this system might begin with the user stating:

“My windshield is broken, help.”

The AYP system must infer that the user wishes to replace the windshield and hence needs to know about automotive repair shops that replace windshields, or glass shops that handle automotive glass.

Allen and Perrault (1980) studied interactions that occur between an information-booth attendant in a train station and people who come to the booth to ask questions. An example of such an interaction is

Q. The 3:15 train to Windsor?

A. Gate 10.

From the question alone it is unclear what goal Q has in mind. However, the attendant has a model of the goals individuals who ask questions at train stations have. The attendant assumes Q has the goal of meeting or boarding the 3:15 train to Windsor. Once the attendant has determined Q's goal, he then tries to provide information to help Q achieve that goal. In Allen's model, the attendant seeks to find *obstacles* to the questioner's goal. Obstacles are subgoals in the plan of the Q that cannot be easily achieved by Q without assistance. In this case the obstacle in Q's plan of boarding the train is finding the location of the train, which the attendant resolves by telling Q which gate the train will leave from.

#### INCORRECT OR INCOMPLETE GOALS AND PLANS

Sometimes the plans or goals that can be inferred from the user's utterances may be incomplete or incorrect. Goodman (1985) has addressed the problem of incorrect utterances in the context of miscommunication in referring to objects. He currently is working on dealing with miscommunication on a larger scale to deal with miscommunication at the level of plans and goals (Goodman 1986). Sidner and Israel (1981) have also studied the problem of recognizing when a user's plan is incorrect, by keeping a library of "buggy" plans.<sup>1</sup> Incomplete specification of a goal by the user can be dealt with via clarification subdialogs, where the system attempts to elicit more information from the user before continuing. Litman and Allen (1984) have presented a model for recognizing plans in such situations.

Situations where user goals are incomplete or incorrect violate what Pollack calls the *appropriate query assumption* (Pollack 1985). The appropriate query assumption is adopted by many systems when they assume that the user is capable of correctly formulating a question to a system that will result in the system providing the information they need. As pointed out in Pollack et al (1982) this is frequently not the case. Individuals seeking advice from an expert often do not know what information they need, or how to express that need. Consequently such individuals will tend to make statements that do not provide enough information, or that indicate they have a plan that will not work. A system that makes the appropriate query assumption must be able to reason about the true intentions of the user when making a response. Often this response must

address the user goals inferred by the system, and *not* the goal explicit in the user's question.

#### MULTIPLE GOALS AND PLANS

A further complication is the need to recognize multiple goals that a user might have. Allen, Frisch, and Litman distinguish between *task* goals and *communicative* goals in a discourse. The communicative goal is the immediate goal of the utterance. Thus in the question

"Can you tell me what time the next train to the airport departs?"

the communicative goal of the questioner is to discover when the next train leaves. The task goal of the user is to board the train. Carberry's TRACK system (Carberry 1983, and this issue) allows for a complex domain of goals and plans. TRACK builds a tree of goals and plans that have been mentioned in a dialog. One node in the tree is recognized as the *focused goal*, the goal the user is currently pursuing. The path from the focused goal to the root of the tree represents the *global context* of the focused goal. The global context represents goals that are still viewed as active by the system. Other nodes in the tree represent goals that have been active in the past, or have been considered as possible goals of the user by the system. As the user shifts plans, some of these other nodes in the tree may become reactivated.

#### 2.2 CAPABILITIES

Some natural language systems need to model the capabilities of their users. These capabilities may be of two types: physical capabilities, such as the ability to physically perform some action that the system may recommend, or (for lack of a better term) mental capabilities, such as the ability of a user to understand a recommendation or explanation provided by the system.

Systems that make recommendations involving actions on the part of the user must have knowledge of whether the user is physically capable of performing such actions. Expert and advisory systems have perhaps the strongest need for this form of knowledge. An expert system frequently asks the user questions to get information about the world. For example, medical diagnostic systems often need to know the results of particular tests that have been run or could be run. The system needs to know whether the user is capable of performing such tests or acquiring such data. Likewise, a recommendation made by an expert system or an advisor is of little use if the user is not capable of following the recommendation.

A natural language system also needs to judge whether the user will be able to understand a response or explanation the system might make. Wallis and Shortliffe (1982) addressed this issue by controlling the amount of explanation provided, based on the expertise level of the current user. Paris's TAILOR system (Paris 1987) goes beyond the work of Wallis and Shortliffe by providing different types of explanations depending on

the user's domain knowledge. Paris, comparing explanations of phenomena from a range of encyclopedias, found that explanations geared towards persons naive to the domain focused on procedural accounts of the phenomena, while explanations for domain experts tended to give a hierarchical explanation of the components of the phenomena. TAILOR consequently generates radically different explanations depending on whether the user is considered to be naive or expert with respect to the domain of explanation. Webber and Finin (1984) have surveyed ways that an interactive system might reason about its user's capabilities to improve the interaction.

Care should be taken to distinguish between mental capabilities and domain knowledge possessed by the user. In each of the examples above, some global categorization of the user has been made (into classes such as naive or expert) with respect to the domain. This category is used as the basis for a judgment of the user's mental capabilities. Much more could be done: modeling of mental capabilities of users should also involve modeling of human learning, memory, and cognitive load limitations. Such modeling capabilities would allow a natural language system to tailor the length and content of explanations, based on the amount of information the user is capable of assimilating. Modeling of this sort seems a long way off, however. Cognitive scientists are just beginning to address some of the issues raised here, with current work focusing on very simple domains, such as how humans learn to use a four-function calculator (Halasz and Moran 1983).

### 2.3 ATTITUDES

People are subjective. They hold beliefs on various issues that may be well founded or totally unfounded. They exhibit preferences and bias toward particular options or solutions. A natural language system may often need to recognize the bias and preferences a user has in order to communicate effectively.

One of the earliest user modeling systems dealt with modeling user preferences. GRUNDY (Rich 1979) recommended books to users, based on a set of self-descriptive attributes that the users provided and on user reactions to books recommended by the system. Although GRUNDY dealt with personal preferences and attitudes, it had the advantage of being able to directly acquire these attitudes by asking the user. In most situations it is not socially acceptable to question a user about particular attitudes, hence the system must resort to acquiring this information implicitly—based on the behavior of the user. The Real-Estate Advisor (Morik and Rollinger 1985) and HAM-ANS (Hoepfner et al 1983, Morik 1988) do this to some degree in the domains of apartment and hotel room rentals. The user will express some preferences about particular types of rooms or locations, and each system can then make deeper inferences about preferences the user might

have. This information is used to tailor the information provided and the suggestions made by the systems.

A natural language system needs to consider personal attitudes when generating responses. The choice of words used, the order of presentation or the presence or lack of specific items in an answer can drastically alter the impact a response has on the user. Jameson (1983, 1988) addresses this issue in the system IMP. IMP takes the role of an informant who responds to questions from a user concerned with evaluating a particular object (in this case, an apartment). IMP can assume a particular bias (for or against the apartment in question, or neutral) and uses this bias in the responses it makes to the user. Thus if IMP is favorably biased towards a particular apartment, it will include additional but related information in responses that favorably represent the apartment, while attempting to temper negative features with qualifiers or additional non-negative features. Thus IMP strives to be a cooperative, biased system while appearing to be objective.

Swartout (1983) and McKeown (1985a) address the effects of the user's perspective or point of view on the explanations generated by a system. In the XPLAIN system built to generate explanations for the Digitalis Therapy Advisor, Swartout uses a very rudimentary technique to represent points of view. Attached to each rule in the knowledge base is a list of viewpoints. Only rules with a viewpoint held by the user are used in generating an explanation. McKeown uses intersecting multiple hierarchies in the domain knowledge base to represent the different perspectives a user might have. This partitioning of the knowledge base allows the system to distinguish between different types of information that support a particular fact. When selecting what to say the system can choose information that supports the point the system is trying to make, and that agrees with the perspective of the user.

Utterances from the user must be considered in light of potential bias as well. Sparck Jones (1984) considers a situation where an expert system is used to compute benefits for retired people. The system is used directly by an *agent* who talks to the actual people under consideration by the system (the *patients*).<sup>2</sup> In this case the system must recognize potential bias on the parts of both agent and patient. The patient may withhold information or try to "fudge" information in order to improve their benefits, while the bias of the agent may color information about the patient by the way the agent provides the information to the system.

### 2.4 KNOWLEDGE AND BELIEF

Any complete model of a user will include information about what the user knows, or what he believes. In the context of modeling other individuals, an agent does not have access to objective truth and hence cannot really distinguish whether a proposition is known or simply believed to be true. Thus the terms *knowledge* and *belief* will be used interchangeably.

Modeling the knowledge of a user involves a variety of things. First, there is the knowledge the user has of the domain of the application system itself. In addition, a user model may need to model information the user has about concepts beyond the actual domain of the application (which might be called *commonsense* or *world* knowledge). Finally, any user, being an intelligent agent, has a model of other agents (including the system) and even of himself or herself. These models are recursive, in that the user's model of the system will include information about what the user believes the system believes about the user, about what the user believes the system believes the user believes about the system, and so on. In the following paragraphs each type of belief is explored in more detail.

#### DOMAIN KNOWLEDGE

Knowing what the user believes to be true about the application domain is useful for many types of natural language systems. In generating responses, knowledge of the concepts and terms the user understands or is familiar with allows the system to produce responses incorporating those concepts and terms, while avoiding concepts the system feels the user might not understand. This is especially true for intelligent help systems (Finin 1982), which must provide clear, understandable explanations to be truly helpful. Providing definitions of database items (such as the TEXT system does (McKeown 1985b)) has a similar requirement to express the definition at a level of detail and in terms the user understands. UC also uses its user model (KNOME) (Chin 1988) to help tailor responses, such as determining whether to explain a command by using an analogy to commands the user already knows.

Knowing what the user believes is also important when requesting information from the user. As Webber and Finin have pointed out (Webber and Finin 1984), systems that ask questions of the user (such as expert systems) should recognize that users may not be able to understand some questions, particularly when the system uses terminology or concepts the user is unfamiliar with. Such systems need knowledge of the user to aid in formalizing such questions.

Modeling user knowledge of the application domain can take on two forms: *overlay* models and *perturbation* models.<sup>3</sup> An overlay model is based on the assumption that the user's knowledge is a subset of the domain knowledge. An overlay user model can thus be thought of as a template that is "laid over" the domain knowledge base. Domain concepts can then be marked as "known" or "not known" (or with some other method, such as an evidential scheme), reflecting beliefs inferred about the user. Overlay modeling is a very attractive technique because it is easy to implement and can be very effective. Unfortunately the underlying assumption of an overlay model, that the user's knowledge is a subset of the domain knowledge of the system, is quite wrong. An overlay model can not account for users who

organize their knowledge of the domain in a structure different from that used in the domain model, nor can it account for misconceptions users may hold about knowledge in the knowledge base.

The perturbation model is capable of representing user beliefs that the overlay model cannot handle. A perturbation user model assumes that the beliefs held by the user are similar to the knowledge the system has, although the user may hold beliefs that differ from the system's in some areas. These differences in the user model can be viewed as perturbations of the knowledge in the domain knowledge base. Thus the perturbation user model is still built with respect to the domain model, but allows for some deviation in the structure of that knowledge.

McCoy's ROMPER system (McCoy 1985, and this issue) assumes a perturbation model in dealing with misconceptions the user might have about the meaning of terms or the relationship of concepts in the domain of financial instruments. When the user is recognized to hold a belief that is inconsistent with its own domain model, ROMPER tries to correct this misconception by providing an explanation that refutes the incorrect information and supplies the user with corrective information. The domain knowledge in the ROMPER system is represented in a KL-ONE-like semantic network. ROMPER considers user misconceptions that result from *misclassification* of a concept ("I thought a whale was a fish") or *misattribution* ("What is the interest rate on this stock?").

#### WORLD KNOWLEDGE

Often a natural language system requires knowledge beyond the narrow scope of the application domain in order to interact with the user in an appropriate manner. Sparck Jones (1984) has classified three types of knowledge about the user that an expert system might keep:

- *Decision Properties*: domain-related properties used by the system in its reasoning process.
- *Non-Decision Properties*: properties not directly used in making a decision, but that may be useful. Examples of such properties might be the name, age, or sex of the user.
- *Subjective Properties*: non-decision properties that tend to change over time.

Decision properties primarily influence the effectiveness of expert system performance. Non-decision properties can influence the efficiency of the system by enabling inferences that reduce the number of questions the system may need to ask the user. All three types of properties influence the acceptability of the system, the manner in which the system interacts with the user. Static non-decision properties and subjective properties comprise knowledge of the user outside the domain of the underlying application system. While such knowledge may not influence the effectiveness of the under-

lying system, it has a great impact on the efficiency and acceptability of the system. Hence world or common-sense knowledge is useful for a natural language system to enhance its ability to interact with the user.

A special case of modeling information outside the domain of the application is when that information is closely related to the domain. Schuster (1984, 1985) has explored this in the context of the tutoring system VP<sup>2</sup> for students learning a second language. Such students tend to use the grammar of their native language as a model for the grammar of the language they are learning. Since VP<sup>2</sup> has knowledge of the native language of the student, it can be much more effective in recognizing misconceptions the student might have when they make mistakes. A tutoring system would also be able to use this second language knowledge in introducing new material, since frequently such material would have much in common with the student's native language.

#### KNOWLEDGE OF OTHER AGENTS

A final form of user knowledge that is very important for natural language systems is knowledge about other agents. As an interaction with a user progresses, not only will the system be building a model of the beliefs, goals, capabilities, and attitudes of the user, the user will also be building a model of the system. Sidner and Israel (1981) make the point that when individuals communicate, the speaker will have an *intended meaning*, consisting of both a *propositional attitude* and the *propositional content* of the utterance. The speaker expects the hearer to recognize the intended meaning, even though it is not explicitly stated. Thus a system must reason about what model the user has of the system when making an utterance, because this will affect what the system can conclude about what the user intends the system to understand by the user's statement.

A further complication in the modeling a user's knowledge of other individuals are *infinite-reflexive* beliefs (Kobsa 1984). An example of such a belief is the following situation:

S believes that U believes p.  
 S believes that U believes that S believes that U believes p.  
 ⋮

An important instance of such infinite-reflexive beliefs are mutual beliefs. A mutual belief occurs when two agents believe a fact, and further believe that the other believes the fact, and believes that they both believe the fact, and so on. Kobsa has pointed out that in the context of user modeling only *one-sided* mutual beliefs, i.e., what the system believes is mutually believed, are of interest.

User's beliefs about other agents and mutual beliefs cause significant representational difficulties. Kobsa (1985) lists three techniques that have been used to represent beliefs of other agents:

- The *syntactic* approach, where the beliefs of an agent are represented in terms of derivability in a first-order object-language theory of the agent (Konolige 1983, Joshi et al 1984, Joshi 1982);
- The *semantic* approach, where knowledge and wants are represented by the accessibility relationships between possible worlds in a modal logic (Moore 1984, Halpern and Moses 1985, Fagin and Halpern 1985);
- The *partition* approach, where beliefs and wants of agents are represented in separate structures that can be nested within each other to arbitrary depths (Kobsa 1985, Kobsa 1988, Wilks and Bien 1983).

While the first two approaches are primarily formal attempts, the partition approach has been implemented by Kobsa in the VIE-DPM system. VIE-DPM uses a KL-ONE-like semantic network to represent both generic and individual concepts. The individual concepts (and associated individualized roles) form elementary situation descriptions. Every agent modeled by the system (including the system itself) can be thought of as looking at this knowledge base from a particular point of view, or context. The context contains the *acceptance attitude* the agent has towards each individual concept and role in the knowledge base. An acceptance attitude can be either belief, disbelief, or no belief.<sup>4</sup> An agent A's beliefs about another agent B is formed by applying acceptance attitudes in A's context to the acceptance attitudes of B. This technique can be applied as often as needed to build complex belief structures involving multiple agents. Kobsa has further extended the representation to handle infinite-reflexive beliefs in a straightforward manner.

To summarize, several types of knowledge may be required for a natural language system to effectively communicate with the user. This knowledge can be classified into four categories: goals and plans, capabilities, attitudes, and knowledge or belief. Not all of this information may be required for any given application. Each type of information is needed in some forms of interaction, however, and a truly versatile natural language system would require all forms.

### 3 THE DIMENSIONS OF A USER MODEL

User models are not a homogeneous lot. The range of applications for which they may be used and the different types of knowledge they may contain indicate that a variety of user models exist. In this section the types of user models themselves, classified according to several dimensions are studied.

Several user modeling dimensions have been proposed in the past. Finin and Drager (1986) have distinguished between models for individual users and models for classes of users (the *degree of specialization*) and between long- or short-term models (the *temporal extent* of the model). Sparck Jones (1984) adds a third, whether the model is *static* or *dynamic*. Static models

do not change once they are built, while dynamic models change over time. This dimension is the *modifiability* dimension of the model.

Rich (1979, 1983), likewise has proposed these three dimensions, but treats the modifiability category a little differently. Instead of static models, she describes *explicit* models, models defined explicitly by the user and that remain permanent for the extent of the session. Examples of explicit models are "login" files or customizable environments. She uses the term *implicit* model for models that are acquired during the course of a session and that are hence dynamic. This characterization seems to mix two separate issues: the method of model acquisition, and the modifiability of the model. Thus the modifiability category will be limited to refer only to whether the model can change during a session, while the acquisition issues will be discussed in the next section.

Three other modeling dimensions are of interest: the *method of use* (either descriptive or prescriptive), the *number of agents* (modeling a given agent may depend upon the models of other agents as well), and the number of models (more than one model may be necessary to model an individual agent). Figure 2 summarizes these dimensions.

### 3.1 DEGREE OF SPECIALIZATION

User models may be *generic* or *individual*. A generic user model assumes a homogeneous set of users—all individuals using the program are similar enough with respect to the application that they can be treated as the same type of user. Most of the natural language systems that focus on inferring the goals and plans of the user use a single, generic model. These systems include ARGOT (Allen et al 1982), TRACK (Carberry 1983, and this issue), EXCALIBUR (Carbonell et al 1983) and AYP A (Gershman 1981).

Individual user models contain information specific to a single user. A user modeling system that keeps individual models thus will have a separate model for each user of the system. This may become very expensive in terms of storage requirements, particularly if the system has a large number of users.

A natural way to combine the system's knowledge about classes of users with its knowledge of individuals is through the use of *stereotype* models. A stereotype is a cluster of characteristics that tend to be related to each other. When building a model of a user, certain pieces of information serve as *triggers* (Rich 1979) to a stereotype. A trigger will cause the system to include its associated cluster of characteristics into the individual user model (unless overridden by other information). Systems that have used stereotypes such as GRUNDY (Rich 1979), the Real-Estate Advisor (Morik and Rollinger 1985) and GUMS<sub>1</sub> (Finin and Drager 1986) further enhance the use of stereotypes by allowing them to be arranged in a hierarchy. As more information is discovered about the user, more specific stereotypes are

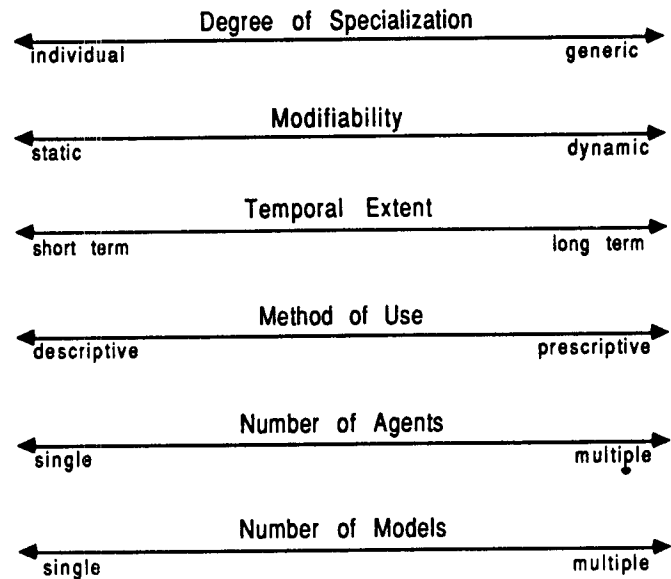


Figure 2. Dimensions of a User Model.

activated (moving down the tree as in GUMS<sub>1</sub>), or the user model invokes several stereotypes concurrently (as in GRUNDY).

A user modeling system might use a combination of these approaches. Consider a database query system. A generic user model may be employed for areas where the user population is homogeneous, such as modeling the goals of users of the system. At the same time, individual models might be kept of the domain knowledge of the users, their perspective on the system, and the level of detail they expect from the system.

### 3.2 MODIFIABILITY

Users models can be static or dynamic. A static user model is one where the model does not change during the course of interaction with the user, while dynamic models can be updated as new information is learned. A static model can be either pre-encoded (as is implicitly done with most programs) or acquired during an initial session with the user before entering the actual topic of the discourse. Dynamic models will incorporate new information about the user as it becomes available during the course of an interaction. User models that track the goals and plans of the user must be dynamic.

Different types of knowledge may require different degrees of modifiability. Goal and plan modeling requires a dynamic model, but user attitudes or beliefs about domain knowledge in many situations may effectively be modeled with static information. Sparck Jones (1984) refers to objective properties of the user (things like age and sex) that are not expected to change over the course of a session. Objective properties, consisting of the decision and non-decision properties in her classification, require only static modeling. On the other hand, subjective properties are changeable and hence require a dynamic model.



### 3.3 TEMPORAL EXTENT

At the extremes, user models can be short term or long term. A short-term model might be one that is built during the course of a conversation, or even during the course of discussing a particular topic, then discarded at the end. Generic, dynamic user models are thus usually short term since they have no facility for remembering information about an individual user.<sup>5</sup> On the other hand, individual models and static models will be long term. Static models by their nature are long term, while individual models are of little use if the information they retain from session to session is no longer applicable.

### 3.4 METHOD OF USE

User models may be used either *descriptively* or *prescriptively*. The descriptive use of a user model is the more "traditional" approach to user models. In this view the user model is simply a data base of information about the user. An application queries the user model to discover the current view the system has of the user. Prescriptive use of a user model involves letting the model simulate the user for the benefit of the system.

An example of a prescriptive use of a user model is in *anticipation feedback loops* (Wahlster and Kobsa 1988). In an anticipation feedback loop the system's language analysis and interpretation components are used to simulate the user's interpretation of a potential response of the system. The HAM-ANS system (Hoepfner et al 1983) uses an anticipation feedback loop in its ellipsis generation component to ensure that the response contemplated by the system is not so brief as to be ambiguous or misleading. Jameson's IMP system (Jameson 1983, 1988) also makes use of an anticipation feedback loop to consider how its proposed response will affect the user's evaluation of the apartment under consideration.

### 3.5 NUMBER OF AGENTS

User-machine interaction need not be one-on-one. In some situations a system may need to actively deal with several individuals, or at least with their models. Recall Sparck Jones's (1984) distinction between the *agent* and *patient* in an expert system: the agent is the actual individual communicating with the system, while the patient is the object of the expert system's diagnosis or analysis. The patient may be human or not (for example, it might be a broken piece of equipment). In the case where the patient is a human, the system must be aware that system requests, explanations, and recommendations will have an impact on both the agent and patient, and that impact may be decidedly different on each individual. In her example of an expert system that advises on benefits for retired people, the agent is responsible for providing information to the system about the patient. The system must have a model of the patient not only for its analysis, but also to guide the communication with the patient. In this case, however, the only way of obtaining that model is through another

individual who will filter information based on his own bias. Thus the system must use its model of the model the agent has of the patient in building its own model of the patient.

### 3.6 NUMBER OF MODELS

It is even possible to have multiple models for a given user. Some of the systems that employ stereotypes, such as GRUNDY, address this by allowing the user model to inherit characteristics from several stereotypes at once. When interaction with an individual triggers several different stereotypes, conflicts between stereotypes must be resolved in some manner. GRUNDY uses a numeric weighting method to indicate the degree of belief the system has in each item in the user model. When new information is added, either directly or through the triggering of another stereotype, evidence combination rules are invoked to resolve differences and strengthen similarities. Thus GRUNDY still maintains a single model of the user and attempts to resolve differences within that model.

The ability to combine stereotypes is also useful for building composite models that cover more than one domain. For example, consider building a modeling system for a person's familiarity with the operating system of a computer, such as was done with the VMS operating system in (Shrager 1981, Shrager and Finin 1982, Finin 1983). The overall domain, knowledge of the VMS system, is quite large and non-homogeneous and can be broken down into many subdomains (e.g., the file system, text editors, the DCL commands interface, interprocess communication, etc). It is more reasonable to build stereotypes that represent a person's familiarity with the subdomains rather than the overall domain. Rather than build global stereotypes such as VMS-Novice and VMS-Expert that attempt to model a stereotypical user's knowledge of the entire domain, it is more appropriate to build separate stereotype systems to cover each subdomain. This allows one to model a particular user as being simultaneously an *emacs-novice* and a *teco-expert*.

Wahlster and Kobsa (1988) consider a situation where a system may require multiple, independent models for a single individual. Among humans this happens all the time when individuals represent businesses or different organizations. Quite often two statements like the following will occur during the course of a business conversation.

"Last time we met we had an excellent dinner together."

"This product is going to be a big seller."

The first statement is made by a salesman speaking as a "normal human," perhaps as a friend of the client. The second statement is made with the "salesman hat" on. Modeling such a situation cannot be handled by multiple stereotype inheritance, because frequently the two hats of the user will be drastically inconsistent. Further-

more, the inconsistencies should not be resolved. Rather it is necessary to be able to switch from one hat to another. This problem is compounded because the two models of an individual are not separate. For example, the goals and plans of the individual may involve switching hats at various points in the conversation. Thus there needs to be a central model of the user, with submodels that are disjoint from each other. The system must then be able to decide which submodel is necessary, and recognize when to switch submodels.

#### 4 ACQUIRING USER MODELS

How a user model is acquired is central to the whole enterprise of building user models. A user model is not useful unless it can support the needs of the larger system that uses it. The ability of a user model to support requests to it depends crucially on the relevance, accuracy, and amount of knowledge the user model has. This in turn depends on the acquisition of such knowledge for the user model. In this section two methods of user model acquisition are discussed, and techniques that have been used to acquire various types of knowledge about the user, particularly the user's goals, plans, and beliefs, will be described.

##### 4.1 METHOD OF ACQUISITION

The knowledge that a user model contains can be acquired in two ways: *explicitly* or *implicitly*. Explicitly acquired knowledge is knowledge that is obtained when an individual provides specific facts to the user model. Explicit knowledge acquisition most often occurs with knowledge acquired for generic user models or for stereotypes. In these cases the user model is usually hand built by the system implementor according to the expectations the designers have for the class or classes of users of the system.

Knowledge can also be acquired explicitly from the user. For example, when a user accesses the system for the first time, the system may begin by asking the user a series of questions that will give the system an adequate amount of information about the new user. This is how GRUNDY acquires most of its individualized information about the user. When a person uses the system for the first time GRUNDY asks for a list of words describing the user. From this list GRUNDY makes judgments about which stereotypes most accurately fit the user (the stereotypes had been hand coded by the system designer) and thus forms an opinion about the preferences of the user based on this initial list of attributes.

Acquiring knowledge about the user implicitly is usually more difficult than acquiring it explicitly. Implicit user model acquisition means that the user model is built by observing the behavior of the user and inferring facts about the user from the observed behavior. For a natural language system this means that the user modeller must be able to "eavesdrop" on the

system-user interaction and make its judgments based on the conversation between the two.

##### 4.2 TECHNIQUES FOR ACQUIRING USER MODELS

In this section techniques that have been used to acquire information for a user model are presented, focusing primarily on how to acquire knowledge about user goals, plans, and beliefs, since these areas have received the most attention to date.

##### GOALS

At any given time, a computer system user will usually have several goals that he is trying to accomplish. Some of these goals may be assumed to apply to all users of the system. For example, a database query system can assume at the very least that the user has the goal of obtaining information from the system. These general goals may either be encoded explicitly in a generic user model, or may be omitted altogether, being assumed in the design of the system itself.

A user modeling system will also need to model user's immediate goals. Sometimes the goals are explicitly stated by the user. For example:

"I want to get to the airport, when does the next train depart?"

Often they are not. Frequently people do not explicitly state their goal, but expect the hearer to infer that goal from the utterance. Thus a speaker who says,

"When does the next train to the airport depart?"

probably has the same goal as the speaker of the first sentence, but the hearer must reason from the statement to determine that goal. This sort of goal inference from indirect questions was part of the work done by Allen and Perault (1980).

##### PLANS

As goals become more complex, the task of inferring a user's goals becomes mixed with the task of inferring the plans held by the user. Much work has been done in recognizing plans held by users. Kautz and Allen (1986) have categorized past approaches to plan inference as using either the *explanation-based* approach, the *parsing* approach, or the *likely inference* approach.

In the explanation approach, the system attempts to come up with a set of assumptions that will explain the behavior of the user. The TRACK system (Carberry 1983, and this issue) uses such an approach. In the context of a system to advise students about college courses, a user might ask,

"Is Professor Smith teaching Expert Systems next semester?"

TRACK will recognize three possible plans the user might have that would explain this statement.

1. The student may want to take Expert Systems, taught by Professor Smith.

2. The student may want to take Expert Systems, regardless of the professor.
3. The student may want to take a course taught by Professor Smith.

TRACK maintains a tree of the possible plans the user may have and refines its judgment as more information becomes available.

The plan parsing approach was first used by Genesereth for the MACSYMA Advisor (Genesereth 1979, 1982). Available to the MACSYMA Advisor is a record of the past interaction of the user with the symbolic mathematics system MACSYMA. When the user encounters a problem and asks the Advisor for help, the MACSYMA Advisor is able to parse the past interaction of the user with the system to come up with the plan the user is pursuing. Such an approach depends on the availability of a great deal of information about the plan steps executed by the user. Plan parsing has not been used for user modeling in natural language systems because of the difficulty in getting such information from a solely natural language interaction.

The likely inference approach relies on heuristics to reduce the space of possible plans that a system might attribute to the user. This approach is used by Pollack (Pollack 1985, Pollack 1986) to infer the plans of users who present inappropriate queries to the system. Pollack reasons that the inappropriate query by the user was an attempt to achieve some subgoal in the user's larger plan. Since this subgoal has failed, Pollack's system tries to identify what the overall goal is, and suggest an action that will salvage the user's plan.

The plan inference approaches rely on two things to accomplish their task. First, all plan inference mechanisms must have a lot of knowledge about the domain and about the kinds of plans the user might have. Many systems implicitly assume that they know all possible plans that may be used to achieve the goals recognizable by the system. Some systems (such as the system described by Sidner and Israel (1981) and Shrager and Finin (1982) augment their domain knowledge with a *bad plan library*—a collection of plans that will not achieve the goals they seek, but that are likely to be employed by a user.

#### BELIEFS

Acquiring knowledge about user beliefs is a much more open-ended task than acquiring knowledge about goals and plans. Goals and plans have an inherent structure that helps acquisition of such information. Inferring the user's plan reaps the side benefit of inferring not only the main goal of the user, but also a number of subgoals for the steps in the plan. User plans tend to persist during a conversation, so new plan inference does not need to be going on continuously. Beliefs of the user, on the other hand, lack that unifying structure. Inferring user beliefs implicitly requires the user modeling system

to be constantly alert for clues it can use to make inferences about user beliefs.

Knowledge about user beliefs can be acquired in many ways. Sometimes users make explicit statements about what they do or don't know. If the system presumes that a user has accurate knowledge of his own beliefs and that the user is not lying (a reasonable assumption for the level of systems today), such explicit statements can be used to directly update the user model.

Even when users do not explicitly state their beliefs, statements they make may contain information that can be used to infer user beliefs. Kaplan (1982) points out that user questions to a database system (as well as other systems) often depend on presuppositions held by the user. For example, the question

“Who was the 39th president?”

presupposes that there was a 39th president. A user modeling system may thus add this belief to its model of the user. When a presupposition is wrong (does not agree with the domain knowledge of the system), it may be possible to infer more information about the beliefs of the user. The incorrect presupposition may reflect an object-related misconception, in which case a system such as ROMPER (McCoy 1985, 1986) could detect whether the misconception was due to a misclassification of the concept, or a misattribution. Such a misconception may indicate a misunderstanding about other, related terms as well.<sup>6</sup>

Other techniques can be used to infer beliefs of the user based on the user's interaction with the system, but with conclusions that are less certain. These approaches can be classified as either primarily *recognition oriented* or primarily *constructive*.

The recognition approaches use the statements made by the user in an attempt to recognize pre-encoded information in the user model that applies to the user. Stereotype modeling uses this approach: a stereotype is a way of making assumptions about an individual user's beliefs that cannot be directly inferred from interaction with the system. Thus if the user indicates knowledge of a concept that triggers a stereotype, the whole collection of assumptions in the stereotype can be added to the model of the individual user (Rich 1979, Morik and Rollinger 1985, Chin 1988, Finin and Drager 1986). Stereotype modeling enables a robust model of an individual user to be developed after only a short period of interaction.

Constructive modeling attempts to build up an individual user model primarily from the information provided in the interaction between the user and the system. For example, a user modeling system might assume that the information provided by the system to the user is believed by the user thereafter. This assumption is reasonable, since if the user does not understand what the system says (or does not believe it), he is likely to seek clarification (Rich 1983), in which case the

errant assumption will be quickly corrected. Another approach is based on Grice's Cooperative Principle (Grice 1975). If the system assumes that the user is behaving in a cooperative manner, it can draw inferences about what the user believes is relevant, and about the user's knowledge or lack of knowledge. Perrault (1987) has recently proposed a theory of speech acts that implements Grice's Maxims as *default rules* (Reiter 1980). Kass and Finin (Kass 1987a, Kass and Finin 1987c) have taken a related approach, suggesting a set of default rules for acquiring knowledge about the user in cooperative advisory systems, based on assumptions about the type of interaction and general features of human behavior.

Another technique mixes the implicit and explicit methods of acquiring knowledge about the user, by allowing the user modeling module to directly query the user. In human conversation this seems to happen frequently: often a hearer will interrupt the speaker to clarify a statement the speaker has made, or to seek elaboration or justification for a statement. In the environment of a natural language system one could envision a user modeling module that occasionally proposes a question to the user that would help the user modeling module choose between two or more possible assumptions about the user that are considered important to the main focus of the conversation.<sup>7</sup>

Finally, there is a close relationship between knowledge acquisition and knowledge representation. The very nature of user modeling implies uncertainty of the knowledge acquired about the user. Often a user model may make assumptions about the user that need to be retracted when more information is obtained. In addition, the subject being modeled is dynamic—as an interaction progresses the user being modeled will learn new information, alter plans, and change goals. The knowledge representation for a user model must be able to accommodate this change in knowledge about the user. To cope with the non-monotonicity of the user model, the knowledge representation system used will need to have some form of a truth maintenance system (Doyle 1979), or employ a form of evidential reasoning.

## 5 DESIGN CONSIDERATIONS FOR USER MODELS

Incorporating a user model into a natural language system may provide great benefits, but it also has some associated costs. The type of information the model is expected to maintain and how the model is used will affect the overall cost for employing a user modeling system. This section focuses primarily on how to weigh the benefits of employing a user model against the cost of acquiring that model. The benefit provided by a user model can be measured by comparing the performance of the system with a user model to the performance of the system without the user model. The cost of a user model may manifest itself in various ways. On systems that must do a lot of implicit modeling, the cost may

appear as a great demand for computational resources such as processor time and memory space. On systems that employ stereotypes or a generic user model, the cost may be in development time: the man hours spent by the system implementors encoding knowledge about the user. For some systems the cost of employing a user model may be very great, while the benefit is slight. Thus the issue of when user models should *not* be used is important as well.

Several characteristics of the underlying application determine the relative benefits and costs of using a user modeling system. These issues are:<sup>8</sup>

- Who bears the burden of responsibility for communication in the interaction?
- What is the penalty for error?
- How rich is the interaction space?
- How adaptable must the system be, and how quickly must it adapt?
- What mode of interaction will be used by the system?

The following subsections will discuss how each of these issues affects the costs and benefits of a user modeling module, concluding with a summary of what types of systems may be expected to profitably employ a user model.

### 5.1 RESPONSIBILITY

In any dialog, one or more of the participants takes the responsibility to ensure that the communication is successful. In human dialogs this burden is usually shared by all participants, but not always. Tutors and advisors often assume most of the burden of responsibility for ensuring that the student or advisee understands the material presented, and that questions from the student or advisee are correctly handled by the tutor or advisor.

Systems that make the appropriate query assumption place the communication responsibility primarily on the shoulders of the user. Since the system assumes the user always provides appropriate queries, the user modeling module has much less work to do. The system can be content to answer the user's queries without having to worry about the possibility of bad plans, or goals that differ from those inferred directly from the user's statement. In the extreme, any failure in understanding can be blamed on the user. Thus the cost of acquiring a user model is not high. On the other hand, a user model may not provide much benefit since the system need not worry about user goals outside the range of those explicitly stated by the user.

A system that bears the responsibility for communication (thus *not* assuming the user makes appropriate queries) has different user modeling requirements. Such systems (for example, consultative expert systems like MYCIN) need to know the knowledge of the user to aid in generating explanations and in posing questions to the user. Goal and plan recognition is not very important since these tend to be defined by the system itself.

A user model can be quite beneficial in improving the acceptability (and maybe the efficiency) of the system. On the other hand, implicit acquisition of knowledge about the user is difficult since the user participation is constrained to responding to the system. Thus the user model will probably need to be acquired explicitly, either through generic models and stereotypes, or by explicit query of the user.

Systems that share the burden of responsibility with the user require the most complex user models. When responsibility is shared, the system must be able to recognize when the user wants to shift topics or alter the focus of the interaction. Thus the system will require a very rich representation of possible user goals and plans to be able to recognize when the user shifts away from the system's plan or goal. A user model thus seems essential to support such mixed initiative interactions. Although goal and plan inference will be more difficult, the user modeling module should have more opportunity to acquire information from the user in a free-flowing exchange. Consequently the costs for acquiring knowledge about user beliefs may be less than in the two previous situations. Systems in which there is a real sharing of the responsibility are, for the most part, still a research goal. Reichman (1981) has analyzed this in the context of human-human dialogs in some detail. Sergot (1983) has studied the architecture of interactive logic programming systems where the initiative of asking and answering queries can be mixed. In the author's own work, the assumption of a shared responsibility between system and user has proven beneficial in acquiring knowledge about the user implicitly.

### 5.2 PENALTY FOR ERROR

How will an error in the user model influence the performance of the application system? A high penalty for error means the user modeling module must limit the assumptions it makes about the user to those that are well justified. Use of stereotypes would be severely limited and inferences that were less than certain would be avoided. As a consequence, the user model may be less helpful to the application system. A high penalty for error thus reduces the benefits that may be obtained by employing a user modeling system. A low penalty for error, on the other hand, allows the user model to make assumptions if it has some justification. Mistakes will be made, but overall the model should be very helpful to the underlying system.

Penalty for error is related to responsibility for communication. A high penalty for error in the user model can only occur when the system assumes some responsibility for the communication. In fact, systems that are solely responsible for ensuring that communication succeeds in an interaction will tend to have the highest penalty for error. In mixed initiative dialogs both user and system are free to interrupt the conversation to correct mistakes that may occur. When the system assumes sole responsibility, the user has no method to

stop the system and try to correct a mistake that has been made. Thus the lack of flexibility in such systems severely impairs the benefits of a user model.

### 5.3 RICHNESS OF INTERACTION SPACE

The range of interaction a system is expected to handle greatly affects the user modeling requirements. If the possible user goals are very limited (such as meeting or boarding trains) or the domain is limited, a user model need not record much information about user. Such situations do not require individual models of the user, and need only very simple acquisition techniques. Acquisition of knowledge about the user might be a simple search to see which collection of information best matches the behavior of the user.

When the range of interaction increases, more is required of the user model. Inferring user plans is a typical example. The number of possible plans a user might have grows explosively as the complexity of the task increases. It is not possible to record all possible plans and simply search for a match. Instead, typical or likely plans must be entered by the system designers, or complex inferencing techniques must be employed.

The range of possible users also influences the degree of specialization needed in the user model. If the users form a homogeneous class, a generic user model can be built that encompasses much of the information that a system might need to know about the user. Thus knowledge acquisition costs are limited to the time required by the system designers to encode the generic model, with very little effort for implicit modeling. As the range of possible users increases, so does the cost of acquiring information about them. On the other hand, user modeling is more important when the set of users is diverse, so the system is able to tailor its interaction to fit the particular user.

### 5.4 ADAPTABILITY

Adaptability is closely related to the richness of the interaction space and to the penalty for error. The greater the range of possible users, the more the system will be required to adapt. If the penalty for error is high as well, the acquisition abilities of the user model must be very good. The more adaptable the system must be, the greater the learning ability of the user modeling module must be.

Adaptability also concerns how quickly the system is required to adapt. Some systems may deal with a wide range of users, but the user modeler has a relatively long time to develop a model of the individual. Such systems have a low penalty for error. If the system must adapt very quickly, stereotyping will be necessary, including the ability for the system to synthesize new, useful stereotypes when it recognizes the need. Such a user model will need to be concerned not only with modeling the current user, but also potential future users.

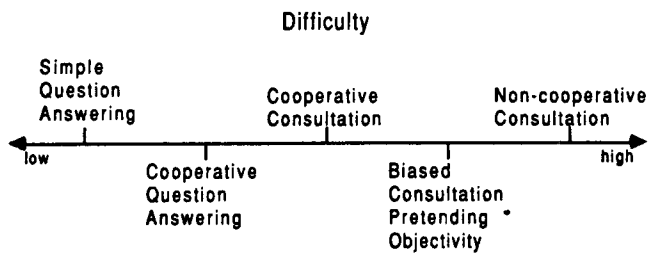


Figure 3. Relative Difficulty of Modeling the User in Different Types of Interaction.

### 5.5 MODE OF INTERACTION

The mode of interaction with the user will also influence the relative cost and benefits of employing a user model. Wahlster and Kobsa (1988) present a scale of four modes of man-machine interaction that place increasing requirements on the user modeling capabilities of a system:

- Simple question answering or biased consultation
- Cooperative question answering
- Cooperative consultation
- Biased consultation pretending objectivity

Figure 3 shows these four modes plus a final, very difficult category:

- Non-cooperative interaction

The following paragraphs take a short look at the user modeling requirements of each.

No explicit user model is required for simple question answering systems such as current database query systems. Such systems are not concerned with user goals and plans, beyond the assumption that the user is seeking information. A minimal user model might be employed to model user knowledge of the domain itself. Biased consultation has similar requirements. No matter what the user says the consultant will make the same recommendation. The only aid a user model might be in helping the system select information likely to sway the user.

Cooperative question answering requires the system to have some idea of the goals of the user. Typically the range of goals the system can be expected to recognize will be quite limited, since the system is being used primarily as an information source. The system must also be able to recognize when a response could lead to a user misconception. Such systems typically can employ a generic user model since there will be little differentiation among users from the standpoint of the question answering system.

Cooperative consultation requires an extensive user model. As noted in Pollack et al (1982), a consultation between an expert and the individual asking advice is

like a negotiation. A consultation system must be able to recognize and understand a wide variety of user goals, further compounded by the fact that they may involve many misconceptions about facts in the domain of consultation. A good consultant should even be able to recognize analogies to other domains that the user is making (Schuster 1984, 1985). Such consultations frequently involve extended interactions where much information about the user can be collected. In most cases this information about the user should be retained, since it is likely further consultations will occur. Thus user models for cooperative consultation need to record all types of information about the user, and save this information in long-term individual user models.

A biased consultation in which the system pretends objectivity (such as an electronic salesman) requires even more inferences about the user than cooperative consultation. Biased consultation requires a deep model of user attitudes, and how particular terms or concepts affect the attitude of the user. The system must have good models of what the user feels is cooperative conversation (since the system must appear objective) and of the user's model of the system (since the system must ensure that the user feels the system is objective).

Non-cooperative interaction makes the acquisition of information about the user very difficult. Even with cooperative interaction, much of the information assumed about the user is uncertain. If the user is not cooperating with the system, the possibility of the user lying, or withholding the truth, further complicates the acquisition of knowledge about the user. The system must be able to reason about the motivations of the user and be able to discern what information is likely to be untrue, and what information should not be influenced by the non-cooperative goals or attitudes of the user. User models in such situations require very extensive knowledge about people in general, and categories of people in particular.

### 5.6 SUMMARY

Given these criteria for judging the costs and benefits of a user model, some conclusions can be drawn about the types of systems that can profitably employ a user model. First, user models should only be used in situations where the range of interaction is sufficiently great that the user model can significantly affect the performance of the system. This does not preclude their use in more limited interactions, but the costs of implementing the user model can easily exceed the benefits that might be gained, particularly compared to other interaction techniques (such as menus) that are easier to implement and quite effective when the range of interaction is limited.

The fact that the user model will be used to alter the behavior of the system implies that the system will assume some degree of responsibility for ensuring the communication between user and system. This means the mode of interaction should at least be cooperative.

Given the range of interaction types presented in Figure 3, cooperative question answering and cooperative consultation are appropriate types of interactions for using a user model. The more difficult forms of interaction, such as biased consultation pretending objectivity or non-cooperative forms of interaction, are very difficult and at present have little practical use in the types of applications being built.

Finally, user models are currently viable only in situations where there is a low penalty for error. A high penalty for error demands very robust user models, requiring either extensive explicit coding of the user model, or sophisticated acquisition techniques. The human costs of coding a robust user model are very high, while sophisticated acquisition techniques will not be forthcoming soon. Thus in applications where the penalty for error is high, responsibility needs to remain on the shoulders of the user, with user modeling playing at most a secondary role.

## 6 CONCLUSION

The ability to interact with people in an easy and natural manner is the promise natural language interfaces hold for computer systems. To realize this promise, systems need to acquire and use various kinds of information about the people with whom they are interacting. That is, they need models of their users.

Sophisticated user models can serve many important functions in natural language systems: they can be used to tailor the interaction to an individual user, to increase the system's cooperativeness, and to correct or even prevent misconceptions by the user. This paper has made several general points about the role of user models in question answering systems.

- What constitutes a user model is a matter of some debate. The view taken in this paper is that a user model is an explicit source of knowledge containing the beliefs and assumptions the system holds about the user.
- User models must hold many diverse types of information. Natural language systems need to know about the user's goals and plans, capabilities, attitudes, and beliefs.
- User models can be classified along various dimensions. In general terms, these dimensions characterize the agents being modeled, how the model changes with time, and how it is used.
- The acquisition of information about the user is a central problem that must be faced. The process can be explicit, implicit, or a mixture of the two. The techniques used for acquisition depend on the kind of information.
- Environmental issues, such as how the model will be used, place added constraints on the type of user model that may be employed in a particular implementation.

To date, most of the work involving the kind of user models discussed in this paper is in an early research stage. This research typically focuses on just one aspect of the overall user modeling problem, such as plan recognition or modeling multiple agents. There is still a great deal of research to be done in these individual areas. Goal recognition and modeling is central to many AI problems and has not yet been adequately handled in any real systems. Many of the ways that a user model can improve natural language interaction have not yet been explored. In the context of generation systems, for example, no existing systems use their knowledge of the user as a factor in the lexical choice problem.

Addressing individual problems in user modeling and looking at particular applications where a user model can help have been appropriate research strategies in early investigations. Ultimately, however, user modeling must be addressed from a more global point of view. A rich, interactive system will need to model many things about many human agents. This information can form a central knowledge base for reasoning about agents in many contexts.

The notion of a central user modeling facility has motivated work on a *general user modeling system* or *general user modeling module* (Finin and Drager 1986, Kass 1987a, Kass and Finin 1987c). A general user modeling system would provide an environment for building systems that used a user model, including various facilities for maintaining and updating user models. A general user modeling module is an independent component of a larger system that provides information about the user to other modules, much like a data base or knowledge base. The interface to the general user modeling module is well-defined, enabling it to be used in a variety of systems with little or no customization.

Future work in user modeling for natural language systems should focus in two directions: establishing how user models should be used in systems that communicate in natural language, and determining how user models can be built more effectively. Many authors have emphasized the need for user models in certain contexts, or have demonstrated that the availability of user model information can improve the behavior of a system. This work needs to be extended to identify *what* information applications will expect a user model to have, *how* that information should be provided to the application, and *when* the information needs to be available. Answers to these questions will help define the services that a user modeling component must provide.

The second focus of research should be on building user models. This work could progress in two ways. First, the task of explicitly building user models (such as building stereotypes) could be made easier. Research in this area seems to parallel efforts to find better ways to acquire knowledge for knowledge bases from experts. However, if general user modeling modules that can

function in diverse systems are to be built, the focus must be placed on the second approach: implicit user model acquisition. In this regard, a user modeling module could be general either with respect to the underlying domain or to the type of interaction. At this time, domain generality seems both a useful and practical goal. The work described in Kass (1987a) and Kass and Finin (1987c) is a beginning in this area, presenting a set of domain general user model acquisition rules for cooperative consultation situations.

User modeling is not an easy task. Effective user modeling requires sophisticated knowledge representation, acquisition, and reasoning abilities—no wonder user modeling is such a new field. On the other hand, advances in any of these areas should provide immediate benefits to user modeling. Thus progress in some of the fundamental areas of AI can result in progress in user modeling as well.

#### ACKNOWLEDGEMENTS

This work was partially supported by grant ARMY/DAAG-29-84-K-0061 from the Army Research Office, grant DARPA/ONR-N00014-85-K-0807 from DARPA, and a grant from the Digital Equipment Corporation.

#### REFERENCES

- Allen, James F. and Perrault, C. Raymond 1980 Analyzing Intention in Utterances. *Artificial Intelligence* 15: 143–178.
- Allen, James F.; Frisch, Alan M.; and Litman, Diane J. 1982 ARGOT: the Rochester Dialogue System. In *Proceedings of the 2nd National Conference on Artificial Intelligence*: 66–70.
- Brown, J.S. and Burton, R.R. 1978 Diagnostic Models for Procedural Bugs in Basic Mathematical Skills. *Cognitive Science* 2: 155–192.
- Carberry, Sandra 1983 Tracking User Goals in an Information Seeking Environment. In *Proceedings of the 3rd National Conference on Artificial Intelligence*: 59–63.
- Carberry, Sandra 1985 *Pragmatic Modeling in Information System Interfaces*. Ph.D. thesis, Department of Computer and Information Science, University of Delaware, Newark, DE.
- Carberry, Sandra (this issue) Modeling the User's Plans and Goals.
- Carbonell, J.R. 1970 AI in CAI: An Artificial Intelligence Approach to Computer-Aided Instruction. *IEEE Transactions on Man-Machine Systems* 11: 190–202.
- Carbonell, Jaime G.; Boggs, W. Mark; Mauldin, Michael L.; and Anick, Peter G. 1983 The XCALIBUR Project: a Natural Language Interface to Expert Systems. In *8th International Conference on Artificial Intelligence*: 653–656.
- Carr, Brian and Goldstein, Ira P. 1977 *Overlays: A Theory of Modeling for Computer-Aided Instruction*. Technical Report AI Memo 406, MIT Artificial Intelligence Laboratory, Cambridge, MA.
- Chin, David N. 1988 KHOME: Modeling What the User Knows in UC. In Kobsa, Alfred and Wahlster, Wolfgang (eds.), *User Models in Dialog Systems*, Springer Verlag, Berlin—New York.
- Doyle, Jon 1979 A Truth Maintenance System. *Artificial Intelligence* 12(3): 231–272.
- Fagin, Ronald and Halpern, Joseph Y. 1985 Belief, Awareness and Limited Reasoning: Preliminary Report. In *9th International Conference on Artificial Intelligence*: 491–501.
- Finin, Tim 1982 *Help and Advice in Task-Oriented Systems*. Technical Report MS-CIS-82-22, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA.
- Finin, Tim 1983 Providing Help and Advice in Task-Oriented Systems. In *8th International Conference on Artificial Intelligence*: 176–178.
- Finin, Tim and Drager, David 1986 GUMS<sub>1</sub>: a General User Modeling System. In *Proceedings of the 1986 Conference of the Canadian Society for Computational Studies of Intelligence*: 24–30.
- Finin, Tim; Joshi, Aravind; and Webber, Bonnie 1986 Natural Language Interactions with Artificial Experts. *Proceedings of the IEEE* 74: 921–938.
- Genesereth, Michael 1979 The Role of Plans in Automated Consultation. In *6th International Conference on Artificial Intelligence*: 311–319.
- Genesereth, Michael R. 1982 The Role of Plans in Intelligent Teaching Systems. In Sleeman, D. and Brown, J. S. (eds.), *Intelligent Tutoring Systems*, 137–156, Academic Press, New York, NY.
- Gershman, A. 1981 Finding Out What the User Wants—Steps Toward an Automated Yellow Pages Assistant. In *7th International Conference on Artificial Intelligence*: 423–425.
- Goodman, Bradley A. 1985 *Communication and Miscommunication*. Technical Report 5681, Bolt, Beranek, and Newman.
- Goodman, Bradley A. 1986 Miscommunication and Plan Recognition. Unpublished paper from UM86, the International Workshop on User Modeling, Maria Laach, West Germany.
- Grice, H.P. 1975 Logic and Conversation. In Cole, P. and Morgan, J.L. (eds.), *Syntax and Semantics* 3, Academic Press, New York, NY.
- Halasz, Frank G. and Moran, Thomas P. 1983 Mental Models and Problem Solving in Using a Calculator. In *Proceedings of the Human Factors in Computer Systems Conference*: 212–216.
- Halpern, Joseph Y. and Moses, Yoram 1985 A Guide to the Modal Logics of Knowledge and Belief: Preliminary Draft. In *9th International Conference on Artificial Intelligence*: 480–490.
- Hoepfner, Wolfgang; Christaller, Thomas; Marburger, Heinz; Morik, Katharina; Nebel, Bernhard; O'Leary, Mike; and Wahlster, Wolfgang 1983 Beyond Domain Independence: Experience with the Development of a German Language Access System to Highly Diverse Background Systems. In *8th International Conference on Artificial Intelligence*: 588–594.
- Jameson, A. 1983 Impression Monitoring in Evaluation-Oriented Dialog: The Role of the Listener's Assumed Expectations and Values in the Generation of Informative Statements. In *8th International Conference on Artificial Intelligence*: 616–620.
- Jameson, Anthony 1988 But What Will the Listener Think? Belief Ascription and Image Maintenance in Dialog. In Kobsa, Alfred and Wahlster, Wolfgang (eds.), *User Models in Dialog Systems*, Springer Verlag, Berlin—New York.
- Johnson, W. Lewis and Soloway, Elliot 1984 Intention-Based Diagnosis of Programming Errors. In *Proceedings of the 4th National Conference on Artificial Intelligence*: 162–168.
- Joshi, Aravind K. 1982 Mutual Beliefs in Question Answering Systems. In Smith, N. (ed.), *Mutual Belief*, Academic Press, New York.
- Joshi, A.; Webber, Bonnie; and Weischedel, Ralph 1984 Living Up to Expectations: Computing Expert Responses. In *Proceedings of the 4th National Conference on Artificial Intelligence*.
- Kaplan, S.J. 1982 Cooperative Responses from a Portable Natural Language Database Query System. *Artificial Intelligence* 19(2): 165–188.
- Kass, Robert 1987 *Implicit Acquisition of User Models in Cooperative Advisory Systems*. Technical Report MS-CIS-87-05, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA.
- Kass, Robert 1987 The Role of User Modeling in Intelligent Tutoring Systems. In Kobsa, Alfred and Wahlster, Wolfgang (eds.), *User Models in Dialog Systems*. Springer Verlag, Berlin—New York. (An earlier version of this paper appeared as Technical Report Number MS-CIS-86-58, Department of Computer Science, University of Pennsylvania, Philadelphia, PA.)



- Kass, Robert 1987 Rules for the Implicit Acquisition of Knowledge About the User. In *Proceedings of the 6th National Conference on Artificial Intelligence*. (Also available as technical report Number MS-CIS-87-10, Department of Computer Science, University of Pennsylvania, Philadelphia, PA.)
- Kautz, Henry A. and Allen, James F. 1986 Generalized Plan Recognition. In *Proceedings of the 5th National Conference on Artificial Intelligence*: 32–37.
- Kobsa, Alfred 1984 Three Steps in Constructing Mutual Belief Models from User Assertions. In *Proceedings of the 6th European Conference on Artificial Intelligence*: 423–427.
- Kobsa, Alfred 1985 Using Situation Descriptions and Russellian Attitudes for Representing Beliefs and Wants. In *9th International Conference on Artificial Intelligence*: 513–515.
- Kobsa, Alfred 1988 A Taxonomy of Beliefs and Goals for User Models in Dialog Systems. In Kobsa, Alfred and Wahlster, Wolfgang (eds.), *User Models in Dialog Systems*, Springer Verlag, Berlin—New York.
- Konolige, Kurt 1983 A Deductive Model of Belief. In *8th International Conference on Artificial Intelligence*: 377–381.
- Litman, D. and Allen, J. 1984 A Plan Recognition Model for Clarification Subdialogs. In *Proceedings of the 10th International Conference on Computational Linguistics*: 302–311.
- McCoy, Kathleen F. 1985 *Correcting Object-Related Misconceptions*. Technical Report MS-CIS-85-57, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA.
- McCoy, Kathleen F. 1988 Highlighting User Model to Respond to Misconceptions. In Kobsa, Alfred and Wahlster, Wolfgang (eds.), *User Models in Dialog Systems*, Springer Verlag, Berlin—New York.
- McCoy, Kathleen F. (this issue) Reasoning on a Highlighted User Model to Respond to Misconceptions.
- McKeown, Kathleen R. 1985 Discourse Strategies for Generating Natural-Language Text. *Artificial Intelligence* 27: 1–41.
- McKeown, Kathleen R. 1985 Tailoring Explanations for the User. In *9th International Conference on Artificial Intelligence*: 794–798.
- Moore, Robert C. 1984 A Formal Theory of Knowledge and Action. In Moore, R.C. and Hobbs, J. (eds.), *Formal Theories of the Commonsense World*, Ablex Publishing, Norwood, NJ; 319–358.
- Morik, Katharina 1986 User Modeling and Conversational Settings: Modeling the User's Wants. In Kobsa, Alfred and Wahlster, Wolfgang (eds.), *User Models in Dialog Systems*, Springer Verlag, Berlin—New York.
- Morik, Katharina and Rollinger, Claus-Rainer 1985 The Real-Estate Agent—Modeling Users by Uncertain Reasoning. *AI Magazine* 6: 44–52.
- Paris, Cecile L. (this issue) Tailoring Object Descriptions to the User's Level of Expertise. *Linguistics Special Issue on User Modeling*.
- Perrault, C. Raymond 1987 *An Application of Default Logic to Speech Act Theory*. Report No. CSLI-87-90, Center for the Study of Language and Information, Stanford, CA.
- Pollack, Martha E. 1985 Information Sought and Information Provided: An Empirical Study of User/Expert Dialogues. In *Proceedings of the Human Factors in Computer Systems Conference*: 155–159.
- Pollack, Martha E. 1986 *Inferring Domain Plans in Question Answering*. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA.
- Pollack, Martha E.; Hirschberg, Julia; and Webber, Bonnie 1982 User Participation in the Reasoning Processes of Expert Systems. In *Proceedings of the 2nd National Conference on Artificial Intelligence*: 358–361. (A longer version of this paper appears as Technical Report MS-CIS-82-9, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA.)
- Reichman, Rachel 1981 *Plain-Speaking: A Theory and Grammar of Spontaneous Discourse*. Ph.D. thesis, Harvard University, Cambridge, MA.
- Reiter, Raymond 1980 A Logic for Default Reasoning. *Artificial Intelligence* 13(1): 81–132.
- Rich, Elaine 1979 User Modeling Via Stereotypes. *Cognitive Science* 3: 329–354.
- Rich, Elaine 1983 Users as Individuals: Individualizing User Models. *International Journal of Man-Machine Studies* 18: 199–214.
- Schuster, Ethel 1984 *VP<sup>2</sup>: The Role of User Modeling in Correcting Errors in Second Language Learning*. Technical Report MS-CIS-84-66, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA.
- Schuster, Ethel 1985 Grammars as User Models. In *9th International Conference on Artificial Intelligence*: 20–22.
- Sergot, M. 1983 A Query-the-User Facility of Logic Programming. In Degano, P. and Sandewall, E. (eds.), *Integrated Interactive Computing Systems*, North-Holland.
- Shrager, J. 1981 *Invoking a Beginner's Aid Processor by Recognizing JCL Goal*. Technical Report MS-CIS-81-07, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA.
- Shrager, J. and Finin, Tim 1982 An Expert System that Volunteers Advice. In *Proceedings of the 2nd National Conference on Artificial Intelligence*: 339–340.
- Sidner, Candace L. and Israel, David J. 1981 Recognizing Intended Meaning and Speakers' Plans. In *7th International Conference on Artificial Intelligence*: 203–208.
- Sleeman, D. 1982 Assessing Aspects of Competence in Basic Algebra. In Sleeman, D. and Brown, J.S. (eds.), *Intelligent Tutoring Systems*, Academic Press, New York, NY; 185–200.
- Sleeman, D. and Brown, J.S. 1982 *Intelligent Tutoring Systems*, Academic Press, New York, NY.
- Sleeman, D.; Appelt, Doug; Konolige, Kurt; Rich, Elaine; Sridharan, N.S.; and Swartout, Bill 1985 User Modeling Panel. In *9th International Conference on Artificial Intelligence*: 1298–1302.
- Smith, Brian 1982 *Reflection and Semantics in a Procedural Language*. Ph.D. thesis, MIT, Cambridge, MA. (Also available as Technical Report MIT/LCS/TR-272.)
- Sparck Jones, Karen 1984 *User Models and Expert Systems*. Technical Report 61, Computer Laboratory, University of Cambridge, Cambridge, England.
- Swartout, William R. 1983 XPLAIN: A System for Creating and Explaining Expert Consulting Programs. *Artificial Intelligence* 21: 285–325.
- Wahlster, W. and Kobsa, Alfred (eds.) 1988 *User Models in Dialog Systems*, Springer Verlag, Berlin—New York.
- Wallis, J. W. and Shortliffe, E. H. 1982 *Explanatory Power for Medical Reasoning Expert Systems: Studies in the Representation of Causal Relationships for Clinical Consultations*. Technical Report STAN-CS-82-923, Department of Computer Science, Stanford University, Stanford, CA.
- Waltz, D.L. 1978 An English Language Question Answering System for a Large Relational Database. *Communications of the ACM* 21(7):526–39.
- Webber, Bonnie Lynn 1986 Questions, Answers, and Responses: Interacting with Knowledge Base Systems. In Brodie, M. and Mylopoulos, J. (eds.), *On Knowledge Base Systems*, Springer Verlag, Berlin—New York.
- Webber, Bonnie Lynn and Finin, Tim 1984 In Response: Next Steps in Natural Language Interaction. In Reitman, W. (ed.), *Artificial Intelligence Applications for Business*, Ablex Publishing Company, Norwood, NJ.
- Wilensky, R.; Arens, Y.; and Chin, D. 1984 Talking to UNIX in English: an Overview of UC. *Communications of the ACM* 27: 574–593.
- Wilks, Y. and Bien, J. 1983 Beliefs, Points of View, and Multiple Environments. *Cognitive Science* 7: 95–119.

## NOTES

Authors' current addresses:

Robert Kass, Center for Machine Intelligence, 2001 Commonwealth Blvd., Ann Arbor, MI 48105;

Tim Finin, Unisys Paoli Research Center, P.O. Box 517, Paoli, PA 19301.

1. The use of such "bug libraries" has proven very successful in student modeling for intelligent tutoring systems. (Brown and Burton 1978, Sleeman 1982, Johnson and Soloway 1984) are examples of just a few intelligent tutoring systems that profitably employ this idea.
2. There is an unfortunate conflict in terminology here. Sparck Jones uses the term "agent" in the sense of an individual who performs a task for another. Thus for Sparck-Jones the agent is the actual individual interacting with the system. Hence in our terminology the system may have agent models for both Sparck-Jones's "agent" and "patient," with the model for the individual Sparck-Jones calls the "agent" actually being a user model.
3. Both the overlay and perturbation models were developed in work on intelligent tutoring systems. The overlay model was first defined by Carr and Goldstein (1977) and used in their Wumpus Advisor (WUSOR) user model, although Carbonell (1970) used an overlay technique in the SCHOLAR program, considered to be the first of the intelligent tutoring systems. A perturbation model was used by Brown and Burton in representing bugs students had in learning multicolumn subtraction (Brown and Burton 1978) and has since been used by many others. See Sleeman and Brown (1982) for a collection of seminal papers on intelligent tutoring systems, or Kass (1987b) for a look at user modeling for intelligent tutoring systems.
4. This is how acceptance attitudes were implemented in VIE-DPM. A wider range of values for the acceptance attitudes, such as a four-valued logic or numeric weights, could easily be used instead.
5. Although it is conceivable that each interaction with an individual user might refine the generic model of all users in some way. Thus such a user model would converge on the "average user" after many sessions.
6. The terms used in a user's statements also provide information about beliefs of the user, but not as much as one might hope. At first glance, it seems that if the user makes use of a word, he has knowledge about the concept to which that word refers. Most of the time this is true. However, people will sometimes use a term that they really don't understand, simply because others have used it. Inferences based simply on the use of terms should be made with care (or with a low level of trust).
7. A very clever system might even be able to incorporate questions from the user modeling module into questions from the application in an attempt to meet two needs simultaneously.
8. The first three issues are suggested by Sridharan in Sleeman et al (1985).