In: Kobsa, A., Wahlster, W. (eds.): User Models in Dialog Systems. Berlin: Springer, pp. 4-34.

User Models in Dialog Systems

Wolfgang Wahlster and Alfred Kobsa[1]

Abstract

This chapter surveys the field of user modeling in artificial intelligence dialog systems. First, reasons why user modeling has become so important in the last few years are pointed out, and definitions are proposed for the terms 'user model' and 'user modeling component'. Research within and outside of artificial intelligence which is related to user modeling in dialog systems is discussed. In Section 2, techniques for constructing user models in the course of a dialog are presented and, in Section 3, recent proposals for representing a wide range of assumptions about a user's beliefs and goals in a system's knowledge base are surveyed. Examples for the application of user models in systems developed to date are then given, and some social implications discussed. Finally, unsolved problems like coping with collective beliefs or resource-limited processes are investigated, and prospects for application-oriented research are outlined. Although the survey is restricted to user models in natural-language dialog systems, most of the concepts and methods discussed can be extended to AI dialog systems in general.

1. Introduction

1.1. Why User Models?

During the 1970s, many special-purpose natural-language (NL) *interfaces* were developed for various domains of discourse, e. g. moon rock samples, airline fares, computer installations, payroll data, aircraft maintenance data, or university courses. These systems had no interests beyond providing the information-seeking user with relevant data by just *responding* in a mechanically cooperative way to the user's questions. The conversational setting for such dialog systems was somewhat unnatural compared to human dialogs. The user of these systems had to find an appropriate question strategy for getting the information s/he thought might be relevant for solving his/her problem, and could hardly expect any assistance from the system in this respect.

[1] This research was supported by the Special Collaborative Research Programme on Artificial Intelligence and Knowledge-Based Systems (SFB 314) of the German Science Foundation. We would like to thank G. Ferrari, Ch. Habel, M. Hecking, A. Jameson and an anonymous referee for their comments on earlier versions of this chapter. This is an extended and revised version of [WAHL86b].

In the 1980s, a new class of task-oriented dialog systems has emerged, all of which are mainly characterized by the *conversational roles* which the system and the user are expected to take on. Here, more realistic conversational settings are modeled in which the system may play the part of, for example,

- a clerk in an information booth at a train station (cf. [ALLE79]),
- a hotel manager who tries to rent all his rooms (HAM-ANS, cf. [JAME80]),
- a librarian who recommends novels that the user might like to read (GRUNDY, cf. [RICH79a, RICH79b]),
- a nosey guest at a cocktail party who wants to find out as much as possible about the user (ULLY, cf. [HAYE76]),
- an expert who assists an apprentice in repairing an air compressor (KAMP, cf. [APPE82b, APPE85]),
- a UNIX and a SINK consultant for beginners (UC, cf. [WILE84, WILE86, CHIN*]; SC, cf. [KEMK86, NESS87, HECK88]),
- a tax advisor who assists the user in filling in his/her income tax form (XTRA, cf. [KOBS86b, ALLG89]).

These systems are designed as *active* dialog partners who engage in a mixed-initiative dialog [WAHL84]. In contrast to the NL interfaces mentioned above, the user of these systems needs no prepared question strategy since the system tries to recognize the user's *intention* with respect to the domain of discourse in order to exhibit more cooperative dialog behavior (e. g. in order to provide better advice). Although even in NL interfaces such a characteristic was necessary to a limited extent, it is particularly in the above conversational settings that the construction and use of an *explicit model of the user's beliefs, goals and plans* becomes a central problem.

Thus, one reason for the recent emphasis on user modeling is the fact that such models are necessary prerequisites in order for a system to be capable of exhibiting a wide range of cooperative dialog behavior (cf. also [CARN83c, RICH83, SPAR84, BOGU85, KASS88c]). A cooperative system must certainly take into account the user's goals and plans, his/her prior knowledge about a domain, as well as false conceptions a user may possibly have concerning the domain (empirical evidence for this in the case of human expert consultation can be found in [POLL82a]). Thus *t* is no longer only the user's task to construct a mental model of the technical functioning of the system. Instead, it should also be up to the system to form assumptions about what the user believes, wants and plans, i. e. to develop a model of the user. Ideally, neither the user's nor the system's modeling task should differ from that which is required in person-to-person communication.

A simple example for such cooperative dialog behavior might look as follows:

(1) User: Where can I find the nearest gas station?
System: The nearest one which is still open is located ...

In order to respond in this cooperative way, a system must discover the presumable plans underlying the user's question, represent them in its knowledge base, examine them for hidden obstacles, and provide information to enable the user to overcome these obstacles. A user model is an indispensable prerequisite for these complex inference processes.

A second reason for the recent emphasis on user modeling is that it has become evident in the last few years that a model of the user is also an important basis for intelligent dialog behavior in general, regardless of whether the dialog is cooperative or not. Such models are required for identifying the objects which the dialog partner is talking about, for analyzing a non-literal meaning and/or indirect speech acts in his/her dialog contributions, and for determining what effects a planned dialog contribution will have on the dialog partner, etc. (cf. [ALLE86, ALLE87]). Thus a user model does not just sweeten-up a dialog system to render it more cooperative. Rather, user models constitute an indispensable prerequisite for any flexible dialog in a wider domain. They interact closely with all other components of the system and often cannot easily be separated from them.

1.2. Some Basic Definitions

Although in this chapter we will restrict the discussion to user models in NL dialog systems, most of the concepts and methods discussed can be extended to AI dialog systems in general. Furthermore, many of the. user modeling components developed for NL dialog systems accept or generate expressions of some formal representation language, since they belong to the language-independent part of the NL system. In this context, we will use the following basic definitions:

> A *user model* is a knowledge source in a natural-language dialog system which contains explicit assumptions on all aspects of the user that may be relevant to the dialog behavior of the system. These assumptions must be separable *by the system* from the rest of the system's knowledge.
>
> A *user modeling component* is that part of a dialog system whose function is to incrementally construct a user model; to store, update and delete entries; to maintain the consistency of the model; and to supply other components of the system with assumptions about the user.

We stress that user models cannot be simply defined as information that the system has about its users. Consider an NL interface to a database, which contains the relation EMPLOYEE (EMP#, NAME, AGE, BONUS) with (26, Jones, 32, 40) as one of its tuples. When Mr. Jones happens to be the user of such a system and asks 'What is my bonus?', his query is translated e.g. into (Select BONUS from EMPLOYEE where NAME = Jones), and the system responds '40'. In this case, the system has information about the user, but we would not like to say that its response was based on a user model, since the system does not view the database tuple as an explicit assumption about its current user (cf. [WAHL88]).

In many conversational situations, the system has to construct and exploit models of more than one agent. Imagine an input to a hypothetical 'marriage guidance system' such as 'This is Bob Jones. My son has problems with his wife. She always .... But I think that my son ...'. In this case, the system's advice should be based on models of Bob Jones, his son, his daughter-in-law and the relationships between these agents. With the definition given above we propose to restrict the term 'user model' to assumptions about aspects of the human directly interacting with the system. In addition, however, we introduce the concept of an *agent model* (or 'actor' model in [KOBS85a]) as a superconcept for 'user model'. Thus, in the example discussed, there are three agent models, only one of which is a user model. For user modeling the system can evaluate the observed behavior of its dialog partner, whereas in the general case of agent modeling this is not necessarily possible. It follows that not all of the techniques developed for user modeling can be generalized to agent modeling.

1.3. User Modeling in Related Areas

1.3.1. User Modeling Within Artificial Intelligence Research

Within the area of artificial intelligence (AI), research on user modeling in NL dialog systems is related to several other fields, namely to the work on *intelligent computer-aided instruction (ICAI)*, *multiple-agent planning systems*, *text comprehension and generation*, *intelligent help systems*, *game playing*, and *expert systems* (see Figure 1).

ICAI systems [SLEE82c, DEDE86] typically incorporate a "student model" which represents the student's understanding of the material to be taught. A simple modeling technique in this field is the overlay approach (e. g. [CARR77, CLAN79]), in which the student's knowledge is represented as a subset of the system's *knowledge* about the domain (cf. Section 3 and [KASS*]). Compared to NL dialog systems, however, the possible user's input to ICAI systems is usually very restricted. Sometimes only expressions in a formal language are accepted, such as mathematical formulas or statements in a programming language.

Research in the field of multiple-agent planning (e. g. [POWE79, KONO80, ROSE82, GEOR83, ROSE85, WERN88]) investigates the problems that a group of agents have in collectively constructing and/or executing a plan for a given task. To coordinate their actions, each agent has to represent the beliefs and goals of the others. Much of this work is grounded on the belief representations developed in the possible-world approach (see Section 3). Problems of natural-language communication between agents are not studied in this field of research. Moreover, for representational purposes, a number of idealizations are typically made with respect to the kinds of beliefs and goals which agents can hold. Sometimes agents cannot possess beliefs which are contradictory to other agents' beliefs (e. g. to those of the system), or cannot hold various forms of mutual and infinite-reflexive beliefs (see Section 2.3). Contradictory beliefs on the part of different agents, however, are quite common in man-computer interaction,[2] and agents which hold various forms of mutual or infinite-reflexive beliefs are practically ubiquitous in this field.

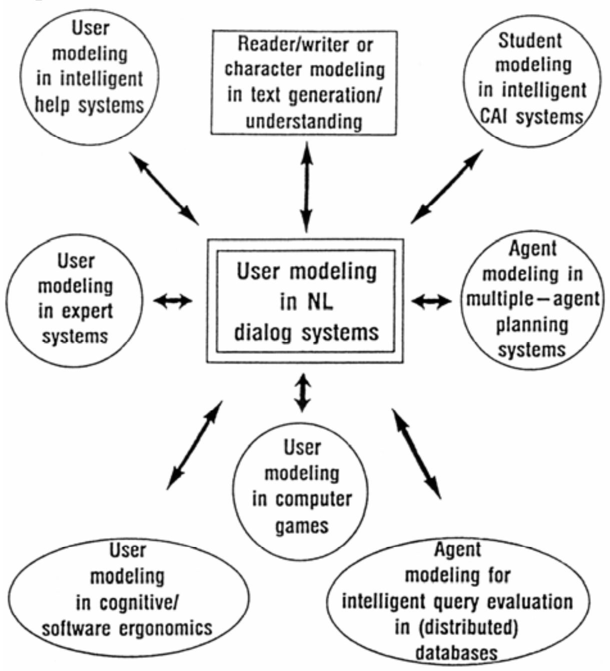

Figure 1. Relationships to other research areas

[2] A lot of work in the field of user modeling is devoted to recognizing user misconceptions, i.e. conflicts between system and user beliefs (e.g. [SCHU83b, MCCO*, QUIL*]).

The adversary nature of two-person games like chess, checkers or backgammon makes game-playing another challenging domain for user modeling research, since in contrast to most other dialog situations, the best systems here behave in a most uncooperative way. It is clear that for brilliant play a model of the opponent's strengths and weaknesses as well as an understanding of his/her playing style is needed. Such a model of the opponent forms the basis for imagining the likely consequences of a planned move. Suppose that the system, being an end game expert in chess, believes that its opponent is a very weak end game player. In such a situation the system will not accept a proposed draw, even when at a disadvantage. Today, the best game-playing systems use an asymmetric evaluation function (see e. g. [HORA87]) and analyze the opponent's previous moves in order to construct a model of his playing strategy. Unfortunately, the user modeling techniques developed so far in this subfield of AI are extremely domain-dependent (they are mostly based on various numeric parameters and scoring functions), so that they cannot be transferred directly to other communicative situations.

1.3.2. User Modeling Outside of Artificial Intelligence Research

Outside of artificial intelligence, a growing interest in user models can be observed in the field of *Human-Computer-Interaction*[3] (for surveys see [CLOW85, MURR87a, MURR87b]). In HCI, however, the notion of 'user model' is employed in a much wider sense than in artificial intelligence research. On the most general level, user models are regarded as forming some kind of mental model [GENT83] which, according to current cognitive psychology, persons usually form of the things and people with whom they interact. Such mental models provide predictive and explanatory power for understanding the interaction. Entries in AI user models, by contrast, need not necessarily be psychologically relevant, i.e. correspond to "mental entities" of the user. They must only enable the system to reconstruct the user's beliefs and goals in sufficient detail.

The G1ossary in [NORM86b] discusses three different types of models that are called 'user models' in HCI research: "[...] the individual's own personal idiosyncratic model; the generalized 'typical user' model that the designer develops to help in the formulation of the design model; or the model that an intelligent program constructs of the person with which it is interacting." In artificial intelligence, the notion of 'user model' is only used in the last sense.

The aim of research in HCI is not necessarily to develop computer systems that construct a model of the user. Instead, knowledge about users' mental models should enable designers to build systems that can be more easily understood by users, or at least to predict learnability and performance. In order to make such models explicit, formal representation methods have recently been proposed for describing the user's and the system's knowledge about the task structure. These representations include state transition networks, formal grammars, and some sort of production systems. Perhaps closer convergence of research pertaining to user modeling in HCI and in AI dialog systems (which are currently nearly independent lines of research) can be achieved on a representational basis as soon as the HCI formalisms have been further developed.

It is worth noting that dialog-based user models can help to improve the performance of systems even for standard tasks like database retrieval. Botman, Kok and Siklossy [BOTM87] describe a so-called active database (ADB) system which extends the literal response to a user question by including extra information the user might be interested in, so that less interaction with the system will be necessary. In many retrieval situations a user is not able to give a precise and complete explication of his/her preferences at the beginning of the dialog. In the ADB system

[3] Related notions are 'Cognitive Ergonomics', 'Software Ergonomics', 'Cognitive Engineering', etc.

the user can indicate which items mentioned in the response to his/her original query s/he finds interesting. Using knowledge about similarities between database values or attributes, the system tries to extract the characteristics of the items upon which the user's choice was based. These assumptions about the user's interests are stored as user profiles together with a certainty factor in the user model. The profiles are used to compute individual interest values for all relevant tuples in the database, which are then used to select and rank information before presentation to the user.

For example, when the user wants to see information about second-hand cars under $ 2,000.00 and then marks several of the displayed BMWs and Audis as interesting, the system may conclude that this choice was based on the 'make' and 'price' attributes, but not on 'color'. When updating the display, the system removes the 'color' attribute and shows more BMWs and Audis, but also includes Alfa-Romeos, because this make was often considered similar to BMWs in previous dialogs.

2. Constructing User Models

In this section, a number of knowledge sources will be identified from which information about the user can be obtained, and a number of techniques for acquiring a user model in the course of a dialog will be discussed.

2.1. Default Assumptions

In systems with a limited area of application, various beliefs and goals can be attributed to any user of the system, as long as there is no evidence to the contrary. Such default assumptions made by the system may concern, for example,

– *The user's general knowledge*: A management information system (MIS) with a user modeling component, for instance, can assume that all of its users possess basic knowledge in business administration.

– *The user's beliefs about individual facts*: An MIS installed in a firm can also assume that the user is familiar with the principal facts concerning the firm.

– *The user's goals*: All users of a hotel reservation system (cf. QAME80, HOEP83b]) can be assumed to be searching for rooms which meet their requirements.

Other possible default assumptions concern the user's beliefs about the system's goals (e. g. that the system wants to rent its rooms), or assumptions about the user's beliefs about the system's beliefs, etc.

2.2. Initial Individual User Models

One criterion which helps to differentiate between the various dialog systems that include a user modeling component is whether

(A) The system can, at the beginning of a dialog, access an already existing model of a particular user which it had constructed during previous interactions with him/her.

(B) The system possesses no previously acquired information about individual users at the beginning of the dialog.

In the former case, the recorded user model will probably prove to be a valuable source of information for the current dialog as well. An example of a system with such an initial user model is GRUNDY [RICH79a, RICH79b] which, at the end of a dialog session, records all information about the user inferred from his/her dialog behavior in a corresponding file. Whenever that user returns to the system and types in his/her name, this file will be retrieved and used as the initial user model.

Today, however, the latter type of system is developed and implemented much more often than the former. This is due not only to the legal, social and ethical problems of security and privacy in such systems, but also to the fact that for many prospective applications in the current era of distributed systems the probability that a user will consult the same system more than once is quite low.

2.3. Assumptions Based on the User's Input

Assumptions formed on the basis of the user's input into the system are the most direct and, hence, also the most reliable in problem-solving domains.

There are various techniques for transforming a user's input into assumptions about him which differ in complexity and in the reliability of the inferred assumptions. The simplest type occurs if the user directly expresses a belief or goal, as in

(2) User: I want to buy your new 500 XG1.
Assumption of the system: The user wants to buy one 500 XG1.

Unfortunately, such direct statements of the user about his/her beliefs or goals are usually very rare. Much more frequent are cases in which assumptions about the user have to be indirectly inferred from the user's input. One type of such inferences is purely *structure-based.* In this case, only the form of a user's input is used for drawing inferences about him/her, and not its meaning. The VIE-DPM system, for instance, forms the following assumptions based on a user's wh-question, and enters them into its knowledge base [KOBS85d, KOBS86a].

(3) User: To whom does Peter give the book?
Assumptions of the system:

(a) The user believes that Peter and the book exist and that Peter gives the book to somebody whose identity is unknown to the user.
(b) The user believes that the system knows to whom Peter gives the book.
(c) The user wants to be in a situation where
(*)both the user and the system have the same opinion about whom Peter gives the book to, and believe that (*)
(d) The user believes that the system believes that (a) - (d).

The infinite loops in the above definition describe specific belief constellations which are fundamental in dialog settings, namely *mutual beliefs* (or more precisely, *infinite-reflexive beliefs* - cf. [KOBS84]). All of the above assumptions can be formed simply because the user has entered a wh-question, without regard to the content of his/her question.

Other purely structural rules of inference (which were used, for example, in [ALLE80, CARB83, CARE*, SDDN83b, MCKE85c]) include

(4) If the user wants to know how to achieve an effect, then his/her goal may be to achieve that effect.
(5) If the user wants to know whether p, then his/her goal may be that p, or not p.

Other forms of inference depend on the *meaning* of a user's input to the system, and on a large amount of universal knowledge about the domain of discourse. From (3), for instance, the system could also infer the following:

(3)(e) Assumption of the system: The user believes that the book is now in the possession of the person whose identity is unknown to him/her.

This inference is based upon the content of the user's question, upon world knowledge about the meaning of 'to give' (namely that it denotes an action, the result of which being the possession of an object by the recipient), and the general inference rule

(6) If the user believes that an action occurred, then s/he also believes that the effects of the action occurred.

[KASS87a] proposes a variety of such general rules for acquiring additional assumptions about the user. Examples include:

Action rule: If the user model includes the belief that a user is familiar with an action, then the user modeling module can attribute to the user knowledge of the preconditions and effects of that action.
Concept generalization rule: If the user model indicates that the user knows several concepts that are specializations of a common, more general concept in the domain model, the user modeling module may conclude that the user knows the more general concept, and the subsumption relationship between the more general concept and the more specialized concept as well.
Inheritance rule: If the user believes a concept A has property P, and further believes that A subsumes concept B, then the user believes B has property P.

Assumptions from the user's input can also be used for inferring his/her possible plans. The systems of Allen & Perrault [ALLE80], Carberry [CARB83, CARB*] and McKeown et al. [MCKE85c] have a large number of domain-dependent precondition-effect sequences (i.e., operators) stored in their knowledge bases. They can then use inference rules such as

(7) If the user's goal is to achieve certain effects, then s/he will probably use a sequence of operators which yield this effect and whose preconditions are fulfilled

for inferring probable user plans. In all these models, however, the system and the user hold identical beliefs about these action operators (e.g. about their preconditions). This restriction is relaxed in the work of Pollack [POLL86a, POLL86b].

In contrast to the other systems, however, Pollack's model does not allow for the chaining of action operators.

Another type of domain-dependent inference is that by which, on the basis of "objective" facts about the user, assumptions are formed about his/her probable traits, in particular his/her preferences, requirements and criteria of assessment in a certain domain. Rich's [RICH79a, RICH79b] GRUNDY system, for instance, forms assumptions about the user's selection criteria for books to read from information about personality characteristics supplied by him/her. Morik & Rollinger [MORI85b] form assumptions about the user's assessment criteria for apartments from information about his/her personal situation, such as the number of children, the acceptable price range, etc. If a great number of inferences are drawn from an antecedent (as is in the purest form the case in GRUNDY), then the set of their consequents may be regarded as a *stereotype* (see Section 4.2 and [RICH*]).

Furthermore, the interpretation of *modal verbs* like 'must, should, need, may' in the user's input makes it possible to infer some of his/her wants and beliefs. Unfortunately, modals are ambiguous in most languages. For example, in German most modals can be used for expressing a want as well as a belief. Sprenger [SPRE88] has developed a computational model for disambiguating modals, so that they can be mapped onto modal operators in the semantic representation language of a user model which is integrated in the consultation system WISBER [SPRE87]. There are many technical problems relating to how a modal verb expresses a belief or goal. Even the correct interpretation of a modal like 'want' in (2) can be quite complex in other contexts. In (8),

(8) User: I want to buy a Porsche, because I would like to have a very fast car.

the user expresses the *belief that* buying a Porsche results in having a very fast car and the *goal* of having a very fast car. In (9), the clause

(9) User: I wanted to buy a BMW, but then I heard that a Porsche is even faster.

beginning with 'but' should cause the user modeling component to assume that the user now wants to buy a Porsche.

Additional assumptions about the user can be formed from *linguistic particles* in his/her input, as in

(9) User (to a text editor system): Why is this line not centered?
Assumption of the system: The user wants the line to be centered.

(10) User: I do *not* have *enough* disk space.
Assumption of the system: The user wants more disk space.

Up to now, the analysis of linguistic particles in the user's input has mostly been investigated on a theoretical basis [KOBS85a]. Such an analysis would offer a simple way of obtaining assumptions about the user. It seems, however, that interesting particles are not very frequently used in goal-oriented dialogs. Moreover, in contrast to the techniques discussed above, this method is also only applicable in natural-language dialog systems.

2.4. Assumptions Based on Dialog Contributions Made by the System

Dialog contributions of the system (i. e. answers, questions and commands) also lead to new entries in the user model. If the system had previously planned the dialog contribution by trying to anticipate its probable effect on the user, it can now hypothetically enter these effects into the user model (e.g. after answering a user's question the system can assume that the user now knows the content of the answer). These entries may give rise to the planning of additional unsolicited comments (cf. Section 4.3.3 and QAME*]) or follow-up information and explanation.

2.5. Assumptions Based on Non-Linguistic Input of the User

In Section 2.3, a number of techniques were discussed for forming assumptions about the user based on his/her input into the system. In contrast to a face-to-face conversation between humans, present dialog systems are restricted to deriving such assumptions solely from the strings typed in by the user; they cannot access any visual, acoustic or other external clues.

By expanding Norman's futuristic idea of a personal user interface on a pocket-sized "smart" card, which, when plugged into any terminal, makes it emulate the one at the user's home or office (cf. QOYC83]), one could imagine a slight deviation from that restriction, namely, that a system's user model is initialized by a kind of electronic identity card storing some of its owner's personality characteristics. The information on this card may be regarded as an initial individual user model in the sense of Section 2.2. The information thereon would be more general, however, and not restricted to the specific domain of a particular system.

At the beginning of an interaction with a dialog system, the use of such a device containing a user-provided individual profile seems no more unnatural than responding to a system's request like

(12) I'd like to know what sort of person you think you are. Please type in a few sinG1e words that you think characterize the most important aspects of yourself

as is required in Rich's [RICH79a, RICH79b] GRUNDY system. Of course, the problem remains as to whether the conveniences offered by such personality cards are not far outweighed by the dangers they give rise to in respect to loss of privacy and easier surveillance.

3. Representing User Models

In the field of user modeling, a wide range of formal means are employed for representing the assumptions of the dialog system regarding the user. What representation one opts for is usually determined by the application purpose of the system and by the demands imposed on the expressive power of the user model.

Rich's GRUNDY system, for instance, makes extensive use of simple *linear parameters.* For every personality trait in her system, one parameter indicates the assumed degree to which the personality trait characterizes the user, and another indicates the degree to which the system believes that its assumption about this specific personality trait is correct. An example of such personality traits is given in Figure 2.

Personality trait	Value	Certainty of assumption
Education	5	900
Seriousness	5	800
Piety	-3	423
Tolerate-sex	5	700
Tolerate-violence	-5	597
Tolerate-suffering	-5	597

Figure 2. Representation in the GRUNDY system

Sleeman's UMFE system [SLEE85] also uses very simple methods for representing its assumptions about the expertise of a user in a given domain. UMFE is a front end system for the explanation component of expert systems. It takes the inference chains yielded by such components and adapts them to the specific level of expertise of the current user. This is done by making sure that only those concepts are used in an explanation which the user has explicitly indicated familiarity with, or which the system has assumed him to know.

For representing the concepts that a user is probably familiar with, a sort of *overlay technique* is employed in Sleeman's model (also see [KASS*]). A simple parameter associated with the representation of each individual concept in the system's knowledge base indicates whether, according to the system's assumptions, the concept is KNOWN or NOT-KNOWN by the user, or whether the system has NO-INFORMATION in that respect. Another factor associated with each concept indicates the strength of belief with which the system holds its assumptions about the user's familiarity with these concepts. Currently, however, no use is yet made of these factors. An example of a simple overlay model (which is inspired by the UMFE system) is given in Figure 3.

As soon as the user model is to give a slightly more detailed account of what the user knows or does not know, of his/her misconceptions, or of his/her goals and plans, etc., the linear scale values must be replaced by representational structures with greater expressive power. In natural-language dialog systems, for representing knowledge about the *world,* representational schemes developed in the field of knowledge representation are employed. Such schemes are, for instance, formulas of first-order predicate calculus, network-like representational schemes, frames, etc. It thus seems most natural to extend these schemes so that the system's assumptions about the beliefs, goals and plans of the user can be represented.

Unfortunately, a simple extension is not possible. When using predicate calculus, one has to decide by what epistemological primitives notions like 'believe', 'want', etc. should be represented. Opting to regard them as sentential operators (as are, for example v, A , ~ or D) causes serious problems, since these operators would not be extensional (an operator is extensional [or truth-functional] if the truth value of a complex formula constructed by it depends solely on the truth value, and not on the "meaning", of its constituent formulas). The lack of this property causes serious problems because fundamental principles of predicate logic (as, for example, substitution of terms with same extension, existential generalization) are no longer applicable.

Concept hierarchy of the system	User model	
	user's knowledge state	certainty of assumption
INFECTIOUS-PROCESS	KNOWN	100
HEAD INFECTION	KNOWN	100
SUBOURAL-INFECTION	NOT-KNOWN	100
OTITIS-MEOIA	NO-INFORMATION	100
SINUSITIS	NO-INFORMATION	100
MENINGITIS	KNOWN	100
BACTERIAL-MENINGITIS	KNOWN	30
MYCOBACTERIUM-TB	NOT-KNOWN	70
VIRUS	NOT-KNOWN	90
FUNGAL-MENINGITIS	NOT-KNOWN	100
MYCOTIC-INFECTION	NOT-KNOWN	100
ENCEPHALITIS	NOT-KNOWN	90
SKIN-INFECTION	KNOWN	100

Figure 3. An example of an overlay model

Another possibility is to regard these notions as meta-language predicates which are applied to meta-language constants and variables that denote object-language formulas. These formulas, in return, express the content of the beliefs and goals. The system's assumption that the user believes that Peter loves Mary thus would be represented in this approach by something like 'BEL(USER,P)', where 'BEL' is a meta-language predicate, 'USER' a meta-language constant that denotes the user, and 'P' a meta-language constant that denotes an object-language formula like 'loves(peter,mary)\ Konolige [KONO81] developed a logic for belief representation that is based on this idea. For more complex belief constellations, however, such representations become computationally untenable. Moreover, as has been shown in [KOBS85a], a number of beliefs that are fundamental for dialog planning based on a user model cannot be represented in Konolige's scheme.

Another approach is to regard notions like 'believe' and 'want' as modal operators, and to use one of the many modal calculi for their axiomatization. This idea was first developed in [HINT62]. Moore [MOOR80] and Appelt [APPE85] investigated a belief and goal representation which formalizes the standard semantics of modal logic, namely *possible-world semantics.* In their representation, modal formulas are immediately translated into possible-world sentences, to which the standard deduction techniques can be applied. The modal operators 'believe' and 'want' are thereby translated into accessibility relations between possible worlds. An agent (e.g. the user) knows some fact p if p is true in all worlds which are compatible with what s/he knows.

The system's assumption that Peter loves Mary thus would be represented in this approach by a formula like

(13) $\forall W\, [K(USER,W0, W) \supset T(W,?)]$

'*W'* thereby is a meta-language variable ranging over possible worlds, 'WO' a meta-language constant denoting the world in which the user currently finds himself, and 'P' a meta-language constant denoting some object-language formula like 'loves (peter, mary)'.

'K (USER, WO, *W)"* is true iff *W\s* accessible from WO according to what the user knows. 'T(U^, P)' is true iff the formula denoted by P is true in *W.* Such a situation is depicted in Figure 4. Here the user knows the fact expressed by 'loves(peter, mary)' but is uncertain with respect to the fact expressed by 'loves(mary, John)', since there is a world which is accessible to him according to what he knows in which the negation of the latter formula is true.

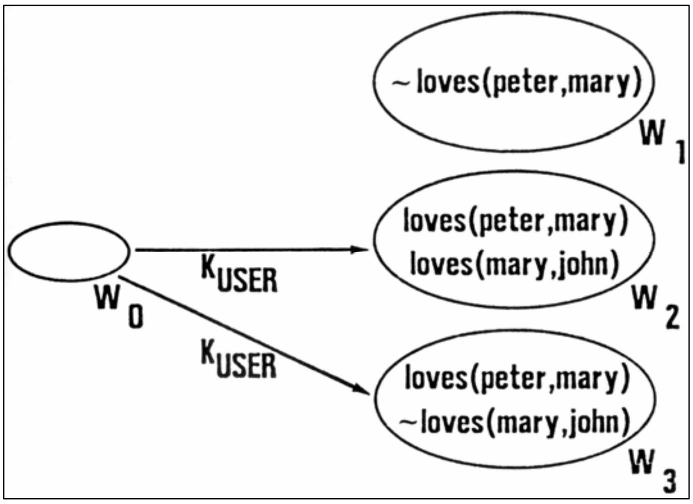

Figure 4. A possible-world representation of 'The user knows that Peter loves Mary and is uncertain whether Mary loves John'

Possible-world semantics has recently been intensively investigated as a means for belief representation [HALP85a, FAGI85, VARD85]. However, as was pointed out in [KOBS85a], as far as the representational needs in the field of user modeling are concerned, the expressive power of this approach is somewhat limited.

Another approach to belief and goal representation is the *"partition approach",* which is applicable to all forms of concept-based knowledge representations. It was first used by Cohen [COHE78] for his special network-oriented representation. The idea behind this approach is to maintain a number of separate partitions to store the system's beliefs about the domain, the system's goals, the system's assumptions about the user's beliefs about the domain, the system's assumptions about the user's goals, the system's assumptions about the user's assumptions about the system's beliefs about the domain, etc. For representing the contents of these beliefs and goals, the standard concept-based knowledge representation schemes (e.g. network-like Mary to love John' representations, predicate calculus) may then be employed. An example of such a representation with a partition for the system's beliefs and another for the system's assumptions about the user's goals is given in Figure 5.

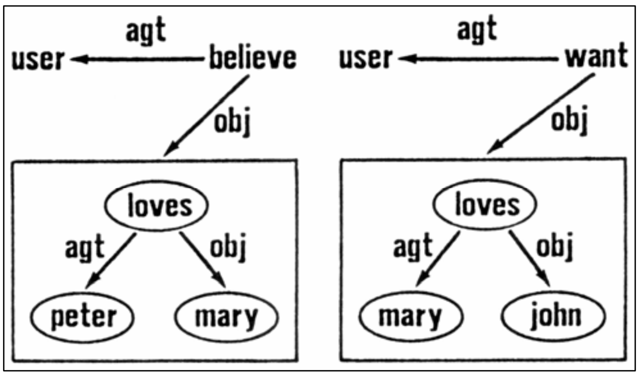

Figure 5. A partition representation of 'The user believes that Peter loves Mary and wants Mary to love John'

Unfortunately, in the form described here, this approach raises some problems which were spelled out, for example, in [MOOR80, SOUL84, KOBS85a]. In [KOBS85d], a new interpretation is therefore given for this sort of belief and want representation, and new epistemological primitives (the so-called *'acceptance attitudes') are* added to increase its expressive power for more subtle belief and goal types. In the BGP-MS system [KOBS88, ALLG89], this representation scheme is currently enhanced by *partition inheritance* which, for instance, allows the user's and system's mutual beliefs to be stored in a separate partition, the contents of which are inherited by the partitions containing the system's beliefs about the domain and the system's assumptions about the user's beliefs about the domain. Kass [KASS87a] plans to extend this representation in such a way that acceptance attitudes can also be applied to rules, and that beliefs also carry justifications for why the belief is held or not held.

4. Exploiting User Models

4.1. Exploitation Outside of the System

In most systems, user models are applied as aids which help the system in pursuing a particular set of dialog goals or in behaving cooperatively when performing a task together with the user. However, there are conversational settings where the construction of a user model is itself the *central* dialog goal (as *is* the case with, for example, exams or personnel interviews) and not just a means to an end (in these cases, the exploitation of the user model is left to human interpreters).

An early proposal for such a distinguished function of a user model in an AI system was the NL dialog system ULLY (cf. [HAYE76]). The conversational setting for ULLY was a fictitious cocktail party at which both the user and the program were "guests". The main goal of ULLY was to find out as much as possible about the user. ULLY was designed "to be a friendly and reasonably co-operative conversational partner, who is politely nosey" (cf. [HAYE76], p. 138).

The authors assumed that at the beginning of the conversation, ULLY has a certain amount of knowledge about the other party guests, such as their names, what they do, how they are related to each other (e. g. X detests Y), but does not know anything about the user. At the end of the session, ULLY should have a more or less refined idea of what the user is like.

ULLY has never been implemented, since - we think - some of the project's goals were overambitious considering the primitive state of the art of user modeling at that time. Although we have seen some advances in user modeling during the last ten years, even today some of ULLY's maxims point to areas open to further research, for example

- try to increase the level of intimacy without being impolite.
- try to keep the conversation at a level which is understandable.
- treat assertions by the user with "care".

4.2. The Relevance of User Modeling in Various Applications of Natural-Language Dialog Systems

If a system is designed as a cooperative consultant who actively recommends X (e. g. a book in GRUNDY or a hotel room in HAM-ANS) or advises the user against X, a model of the user is a necessary prerequisite for performing the corresponding speech acts (cf. [MORI84, MORI*]). A dialog system which generates only fixed and biased recommendations, on the other hand (e. g. a version of GRUNDY which always suggests the Bible first) has hardly any need for a user model. The following types of conversational behavior relevant to AI applications are listed according to an increasing need for a user model:

(A) question-answering and biased consultation
(B) cooperative question-answering
(C) cooperative consultation
(D) biased consultation pretending objectivity.

A system that responds to the literal meaning of simple questions about a flight schedule or a hotel reservation system which always offers the most expensive room first are examples of category (A). Such systems must possess limited user models only, e. g. for handling discourse phenomena like anaphoric reference and ellipsis appropriately. While all systems of this category implemented to date are simple question-answering systems, it is highly probable that biased systems will be installed as soon as the technology of NL processing becomes further commercialized. The focus of current research is shifting from systems of category (B) (e. g. CO-OP, cf. [KAPS79, KAPS82, KAPS83]) to category (C) (e.g. ARGOT, cf. [ALLE82a, ALLE83b]; UC, cf. [WILE84, WILE86]; XTRA, cf. [ALLG89, COHE*]). Systems of both of these categories include user modeling techniques for the generation of various kinds of over-answering (cf. [WAHL83]) and for the generation and recognition of (indirect) speech acts (cf. [ALLE79, PERR80, APPE82b, APPE85]). Systems of category (D), which mimic for example professional salesmanship, require a very sophisticated form of user modeling. The experimental dialog system IMP, which will be discussed below, includes some aspects of that kind of dialog behavior.

As was pointed out by Rich [RICH83], user modeling is especially important if the dialog system is used by a heterogeneous group of users and if the system shows some flexibility in what it tells the user. A typical example of this is her GRUNDY system which, in playing the part of a librarian, can select from a large number of books one that it assumes the user might like to read. The selection is made on the basis of a model of the particular user which the system constructs in the course of the dialog.

Another example of a dialog system for a heterogeneous user group is HAM-ANS which, in its hotel reservation application, plays the part of a hotel manager who tries to rent all rooms available in his/her hotel. On the basis of an explicit model of the individual user, the system can select one of four room categories and suggest a vacant room in the selected category. In this conversational setting the user is assumed to have the overall goal of getting a hotel room which meets his/her requirements.

From a greeting like that in Figure 6 (which includes the name and optionally the title, the affiliation and the home or office address of the user) and from further booking information like that provided by the user in Figure 6, the system can form some assumptions about the prospective hotel guest, e. g. his profession, the purpose of his trip (business trip, vacation, etc.) and his solvency. Several stereotypes are associated with each of these dimensions.

As in GRUNDY, stereotypes are used as collections of frequently recurring characteristics of users. In the example above, HAM-ANS may assume that Mayer is Mueller's secretary and that Mueller is a manager in a high-tech company ("Sunshine Computers"), since he holds a Ph.D. and has a secretary. Furthermore, the system assumes that Mueller is on a business trip and, because his expenses are reimbursed, he is able to spend a lot of money for a room. The user-provided information relevant for the consulting procedure is stored as factual knowledge in the user model, whereas the stereotypes triggered by these facts are stored as hypotheses, which can be withdrawn if they are disconfirmed by the subsequent dialog. Note incidentally that if the caller is booking a room for someone else, the model discussed here does not contain information about the actual user of the system, but about the prospective hotel guest instead. Hence, according to the terminology introduced, this is a case of agent modeling in general, and not specifically user modeling.[4] One can imagine future dialog systems which also model the intermediary persons (e. g. for also taking into account private interests of secretaries), but this is beyond the scope of our discussion.

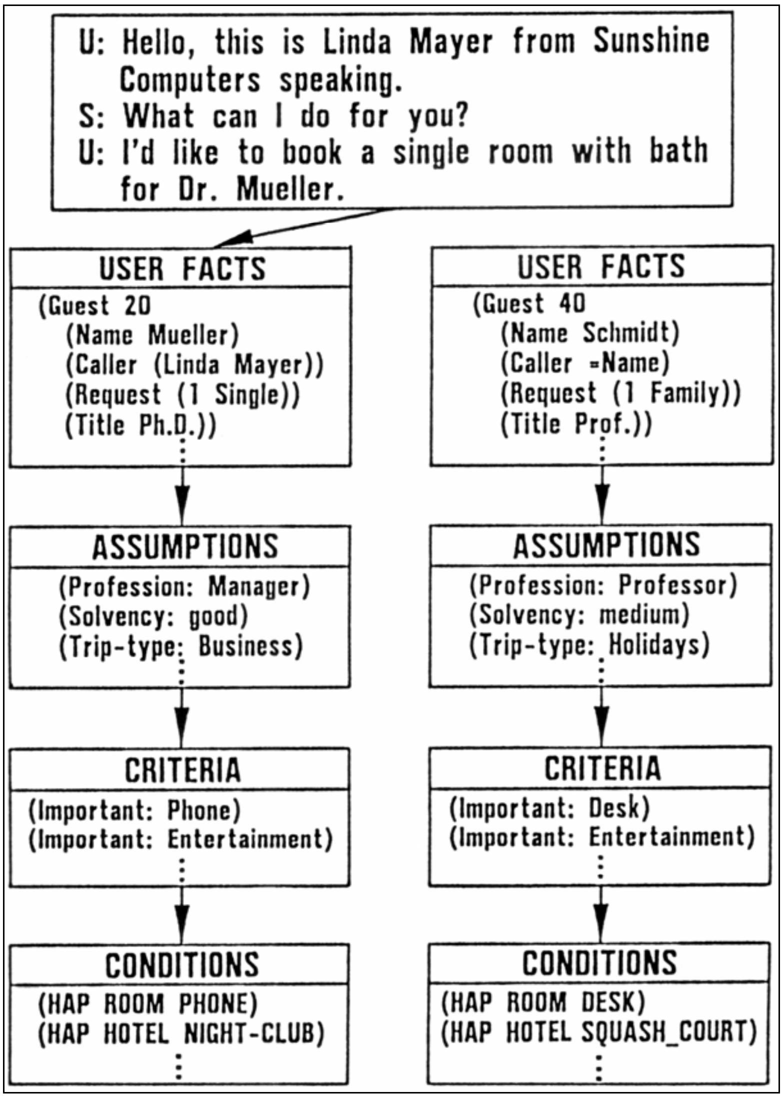

Figure 6. The use of stereotypes in HAM-ANS

HAM-ANS can characterize a particular hotel guest by several distinct stereotypes. For example, its assumptions about a user named Schmidt may be that he is a university professor who wants to spend his vacation with his family, and that he is not able to spend much money because he has five children. As soon as the system has all necessary booking facts at its

[4] Sparck Jones [SPAR84, SPAR*] calls the person being talked about in the discourse the *patient*, and the person interacting with the system (i. e. the user in our terminology) the *agent*

disposal, it checks whether there is a vacant room which meets the requirements explicitly stated by the user (e.g. period of stay, number of beds). If several rooms belonging to different categories meet these explicit requirements, the system uses the activated stereotypes to predict further demands of the user. For example, HAM-ANS may assume that for Mueller, as a manager on a business trip, a room telephone is very important. Prof. Schmidt on the other hand, being a scientist, actually needs a large desk even on his vacation, but in this particular situation a room telephone is not so important for him. These stereotypical inferences generate a list of criteria rated by different importance values. The set of derived criteria is matched against the available room categories in order to generate a strong or weak recommendation and to indicate which criteria are fulfilled and which are not. However, since individuals differ in the conditions they impose for the fulfillment of a criterion, this matching process again depends on the user profile (cf. [MORI85a,MORI85b]). For example, if the guest plans to stay for a longer period in the hotel, attractive entertainment facilities become very important. This criterion, of course, is evaluated differently by a businessman who travels by himself compared with a professor who spends his vacation together with his family.

4.3. Anticipation Feedback

A special case of exploitation of a user model (which is realized in dialog systems like HAM-ANS, NAOS and IMP) is its use in various forms of *anticipation feedback loops.* Anticipation feedback loops involve the use of the system's language analysis and interpretation components to simulate the user's interpretation of an utterance which the system plans to verbalize. The application of anticipation feedback loops is based on the implicit assumption that the system's procedures for language analysis and interpretation (but not necessarily the contents of the sources of knowledge used by these procedures) are similar to those of the user. Such a similarity assumption seems to be quite common among participants in everyday conversations. Since most people (except, of course, some computational linguists and AI researchers) cannot be assumed to have an exact model of the comprehension capacity of their dialog partners (except for some naive assumptions concerning small children and non-native speakers) it is highly plausible that they use the similarity assumption as a default rule. Although the current state of the art in NL dialog systems in no way provides sufficient evidence for assuming a far-reaching similarity, user modeling based on anticipation feedback loops can considerably improve the quality of present dialog systems.

4.3.1. Local Anticipation Feedback for Ellipsis Generation

The ellipsis generation component of the dialog system HAM-ANS incorporates a local anticipation feedback loop which helps to ensure that the system's utterances are not so brief as to be ambiguous or misleading (cf. QAME82]). Consider for example the user's question (14).

(14) User: Do five colleagues have more than three weeks vacation?

If five colleagues have in fact not more than three, but more than two weeks vacation or if only two colleagues have more than three weeks vacation, a cooperative system might respond with (15), instead of simply answering 'No'.

(15) System: No, two.

By using its ellipsis resolution component in an anticipation feedback loop, the system can recognize that (15) would be a confusing response since it is so elliptical that, without further context, no unambiguous interpretation can be assigned to it. Acceptable responses like (16) and (17) include more overlap with question (14).

(16) System: No, two weeks.
(17) System: No, two colleagues.

Having rejected (15), the system internally generates the less elliptical responses (16) or (17), depending on the intended meaning. This time, the anticipation of the user's understanding causes a positive feedback, so that the elliptical structure can now be mapped onto a well-formed NL utterance by the surface transformation component of the system (for a more complete account of HAM-ANS's user modeling in ellipsis generation cf. QAME82]).

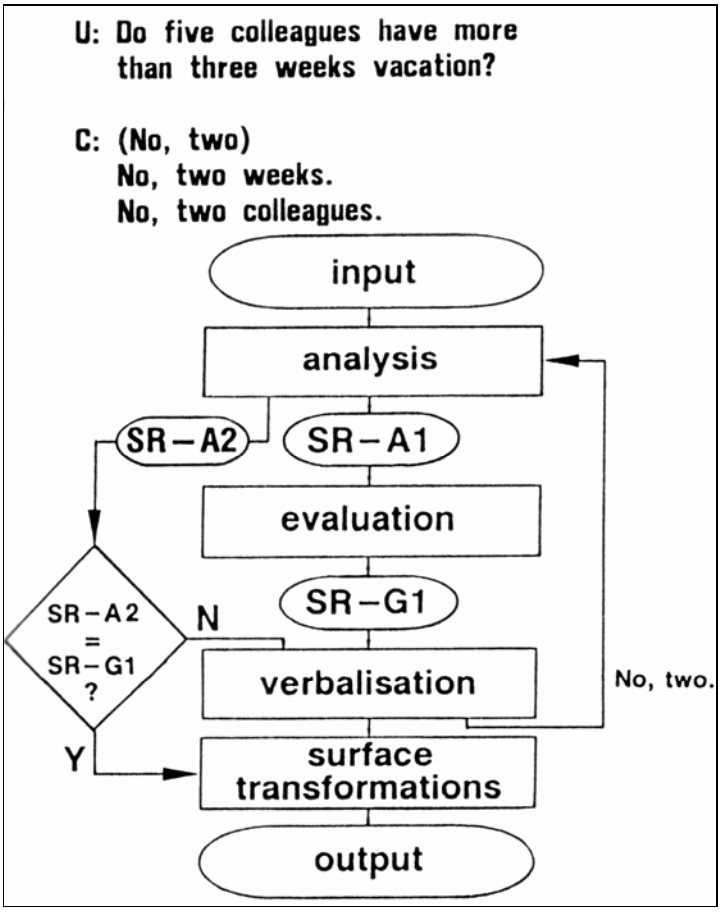

Figure 7. A simple anticipation feedback loop

Figure 7 shows an extremely simplified version of the general structure of a dialog system with such an anticipation feedback loop for user modeling. The parsing of an input results in the semantic representation construction SR-Al, whose evaluation yields SR-G1 as the semantic representation of the intended response. The result of a preliminary verbalization process for SR-G1 (which includes the above-mentioned ellipsis generation) is fed back into the system's analysis component, which is equipped with an ellipsis resolution component. The semantic representation construction SR-A2, which forms the result of the ellipsis resolution process (and which may consist of a disjunction if the input is ambiguous) is then compared with the intended meaning of the response, namely SR-G1. If SR-A2 matches SR-G1, the structure fed into the analysis component is transformed and finally outputted, otherwise an alternative verbalization of SR-G1 is checked in the next iteration of the feedback process.

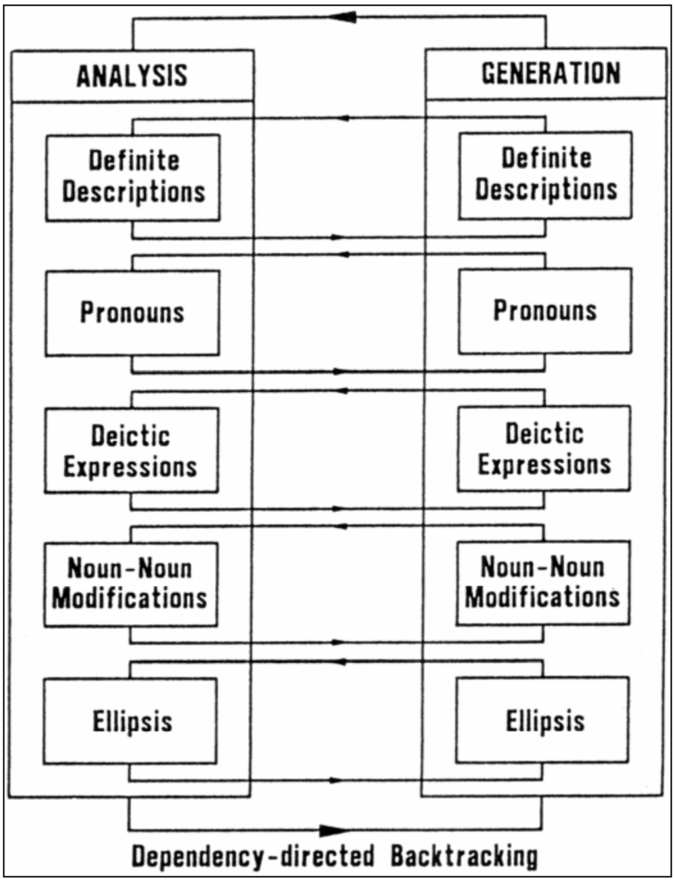

Figure 8. Local and global anticipation feedback loops

The anticipation process discussed above is based on a local feedback loop, i- e. one in which the generation procedures of one particular aspect of an utterance are linked with the recognition procedures for the same aspect.

A variety of phenomena, such as the generation of definite descriptions, pronouns, deictic expressions and noun-noun modifications can be handled in a dialog system either by a corresponding multiplicity of local feedback loops (cf. [WAHL83, MARB83]) or by a larger one within which the whole generated utterance would be tested once, just before applying the final surface transformations (see Figure 8). Such Global anticipation feedback loops require dependency-directed backtracking, so that the system returns to the adequate decision point in the generation process whenever the result of the feedback process is negative.

4.3.2. A G1obal Anticipation Feedback Loop

An example of a more Global anticipation feedback loop is the imagination component of the system NAOS, which generates NL descriptions of real-world image sequences. [NOVA84] describes the design of a user model for NAOS, which contains a representation of the imagined scene resulting from an anticipation of the user's understanding of a planned utterance. Such a system would take into account the present state of the dialog and the assumed knowledge and dialog goals of the user. On this basis, the representation of the imagined scene (which always contains some uncertainty) and the actual scene observed by the system must be matched in order to check whether the planned utterance will be interpreted by the user in a consistent manner and whether it will generate the intended perceptual impression in adequate detail. For instance, it is quite clear that the image sequence of a car accident in which one was involved is described differently to the police, the lawyer of the opposing party, or a good friend.

This application shows that a user model cannot completely rely on representation schemes only suited for the representation of discrete domains. Instead, they must also provide means for the representation of continuous domains, e.g. of the fuzzy regions of an object's trajectory which might form part of the image anticipated by the speaker for a planned utterance like (18).

(18) A red BMW passed a small VW.

4.3.3. Anticipation Feedback in Evaluation-Oriented Discourse

Jameson [JAME83, JAME*] has studied a very advanced form of anticipation feedback for user modeling in his IMP system. IMP was designed to answer the questions of a user whose sole dialog goal was assumed to be the assessment of some object. On the basis of explicit assumptions about the user's standards and prior expectations, IMP goes beyond the literal answering of questions by adding unsolicited comments that influence the evaluation of the object under discussion. In evaluation-oriented dialogs (e. g. personnel selection interviews, dialogs with travel agents, hotel managers or other salespeople) the selection of such comments is influenced by the nature of the speaker's bias (e. g. a desire to present something in a favorable light). Since listeners take these facts into account when interpreting such comments, IMP anticipates the interpretation of the user while taking into account the bias which the user possibly ascribes to it. IMP's actual bias can be positive, negative or objective, and it can attempt to maintain a positive, negative or objective image. The actual and projected bias may differ (e.g. usually a salesman's actual bias is positive though he attempts to maintain a neutral image).

On the representational level IMP's user model includes an explicit record of the user's expectations concerning a particular room offered for rent. This record is updated according to the user's anticipated interpretation of the system's answer. The most interesting procedural aspect of IMP's user modeling is an anticipation feedback loop which takes into account the effect of a planned utterance on the user's impression of the room and helps to filter out comments which would betray the system's bias.

IMP's user model is also used to select expressions like 'but', 'by the way' and 'unfortunately', which precede the additional comments generated by the system. An example is given in (19):

(19) User: What about facilities outside the room?
System: There's a kitchen and there's a bathtub and by the way, the room is very large.
User: Is the room on a quiet street?
System: No, in fact unfortunately there's a lot of noise from the street.
[JAME83], p.618)

For selecting such connectives and sentential adverbs the system compares the shift in the expected value produced by the present comment with that produced by the preceding statement. IMP shows that for an explanation of the complex semantics and pragmatics of such expressions a user model is necessary.

As we have shown, the use of an anticipation feedback loop is an important part of the process of designing a successful utterance. In essence, anticipation on the part of the speaker means answering question (20) (cf. [ROSN81]).

(20) If / heard this utterance relative to a set of beliefs B and goals G, where B and G are what / assume the hearer believes and wants, respectively, then what would be the effect on me?

It is obvious that a dual question like (21) is useful for the hearer's analysis of utterances.

(21) If / had made this statement, on the basis of what beliefs and goals would have said it?

4.4. Correcting User Misconceptions

In the last few years, a number of techniques have been developed for recognizing misconceptions underlying unsuitable dialog contributions of the user, and for responding to these dialog contributions in an appropriate way (for a brief survey, see [WEBB83]). Each of these methods is restricted to a very particular type of user misconception. The work of Kaplan [KAPS79, KAPS82, KAPS83], for instance, is concerned with the detection of false presuppositions in users' database queries. These include merely "syntactic" presuppositions regarding the existence of the objects mentioned in the discourse, such as

(22) How many female employees in division 3 are managers?
(23) Presupposition: there are female employees in division 3.

Kaplan also considers "indirect" existential presuppositions which can only be inferred using domain knowledge, such as in

(24) User: Which students got a grade of F in CS105 in Spring 1980?
(25) System: CS105 was not given in Spring 1980.

[MAYS80b] is concerned with the recognition of violations of argument restrictions presupposed in users' questions, such as in

(26) User: Which undergraduates teach courses?
(27) System: Undergraduates don't teach courses. Only the faculty teaches courses.

A similar mechanism for detecting "intensional" errors has been implemented in HAM-ANS in its application as an access system to a relational database containing information about research voyages to the antarctic sea [MARB87, MARB88]. The user is informed if relations whose existence is implied by the user's query are not modeled in the database.

All these techniques for detecting and responding to user misconceptions are *not* based on an explicitly represented user model (though an "implicit" user model is hidden in the operation of the mechanisms which implement these techniques). Such an approach is fully justified in limited domains. In more flexible systems, however, it will become necessary that the assumptions about the user inferred by such techniques be explicitly represented in the user model (e.g. for using them in the determination of the user's level of expertise) and that the handling of user misconceptions be performed on the basis of such an explicit model. Moreover, it can be expected that most types of user misconceptions not described above can only be adequately dealt with on the basis of an explicit user model.

A good example of work along these lines is McCoy's ROMPER system [MCCO85a, MCCO85b, MCCO*] which corrects users' object misclassification (i.e. their subsumption under a wrong superconcept in a concept hierarchy) and the attribution to objects of properties which they do not actually have. Response patterns exist for each type of misconception which enable the system to correct the user. An example of a response to an object misclassification looks as follows:

(28) User: I thought that whales were fish.

(29) System: No, they are mammals. You may have thought they were fish because they are fin-bearing and live in the water. However, they are mammals since, (while fish have gills) whales breathe through lungs and feed their young with milk.

Another example of misconception handling on the basis of a user model is Quilici's [QUIL*] AQUA system. AQUA recognizes user misconceptions about UNIX inferable from his/her reports on problems which s/he has had in interaction with UNIX, and selects appropriate responses. A misconception is discovered by checking each recognized user belief about UNIX against the system's expert knowledge. If the incorrect belief is detected in a catalogue of frequent misconceptions held by the system, the user can be immediately corrected. Otherwise, the type of misconception is determined and associated explanation patterns tested as to whether their preconditions match the recognized user and system belief situation. When an explanation matches completely, the user can be informed of the recognized misconception such as in

(30) User: So to remove a file, I have to own the directory that contains it.

(31) System: No, to remove a file, you need to have write permission on the directory that contains it. You do not need to own the directory that contains it. You need to own that directory when you do not already have write permission on it.

4.5. Some Social Problems in Exploiting User Models

If the user is *aware* that the system with which s/he is interacting is equipped with a user modeling component, then s/he has the chance to apply all the techniques for hiding his/her beliefs, goals and plans which s/he constantly uses in every-day person-to-person interaction. This includes, for example, pretending that one knows less or more about a domain, hiding one's goals and plans, beating about the bush, etc. In some applications (e. g. tutoring or unsolicited advice) the user is necessarily aware of the existence of a modeling component; in other applications its existence might not be so apparent to him/her.

Control of the user over the user modeling component means, for example, that the user can inspect the model which the system has made of him/her in the course of a dialog. However, since the number of assumptions made by the system in a normal interaction can be expected to be enormous (cf. Section 2.3.), is it not possible for the user to inspect them all. Another problem is rendering these assumptions comprehensible to him/her. The representational categories employed for user modeling often do not correspond to the usual "folk-psychological" belief and goal categories. Thus the contents of user models are often very difficult to translate into ordinary Eng1ish.

Control of the user over the user modeling component may also mean that the user is entitled to alter the assumptions that s/he has inspected. In this case, as well, the problem arises of how to explain the consequences of such alterations to the user. One might argue that such an option might seduce him/her to change actually correct assumptions in order to make them correspond to his/her (more positive) self concept. In the case of cooperative systems and computer-based tutoring, however, it is the user's problem if the system's cooperativeness declines as a result of these manipulations of the user model.

Another interpretation of a user's control over a system's user modeling component is that the user has the possibility to "switch off" this component and to communicate with the remaining "unintelligent" system alone, if s/he does not consent to being modeled by the system. In some cases (e. g. tutoring), this is not possible due to the nature of the application. In other cases (e. g. cooperative information), it is certainly not advisable to do so. However, there are also applications (e. g. unsolicited advice) where such a demand on the part of the user may be justified. The technical question remains, however, whether it is possible to separate the user modeling component from the remaining system in such a way that it is possible to "switch it off".

User modeling therefore involves the risk of *misunderstandings* (as is also the case in normal human communication). It can be expected, however, that the gain in cooperativity will greatly exceed that risk. Often it is only this cooperativity which enables the casual user to gain access to the system at all. In traditional dialog systems, by the way, there is also the problem of misconception, however only on the part of the user (e. g. about the meaning of some command or some technical term, which, when undetected, may lead to a wrong interpretation of the results.) In any case, a necessary characteristic of a user model must be that the assumptions stored in it be revisable if counterevidence is observed (cf. [DOYL80]). As we mentioned already in our introductory definitions (see Section 1.2) it is one of the tasks of a user modeling component to maintain the consistency of the model. Therefore a truth maintenance system (TMS, also known as belief revision or reason maintenance system, cf. [DOYL79, DOYL83]) is an important subpart of a user modeling component. Actually, the TMS is a crucial factor in the effectiveness of recent user modeling shells like GUMS (cf. [FINI86, FINI*]) and TRUMP (a TMS-and Rule-based User Modeling Prototype, cf. [BONA87]).

5. Open Questions and Future Research in User Modeling

5.1. Coping with Collective Beliefs

In the research discussed so far, user modeling was restricted to models of individuals. However, as Wilks & Bien [WILK83] have pointed out, it is necessary in some dialog situations to model the beliefs of organizations (e. g. NSF's view of AI research), of states (e. g. Germany's view of the European Community), or of classes of individuals (e.g. the sales division's view of AI). Sometimes in a conversation, an individual (e.g. a sales representative, a district attorney, a diplomat) has to advocate the beliefs and goals of such a collective third party without referring to his/her personal beliefs or goals. Thus it may be reasonable for a cooperative NL interface to a commercial DBMS to restrict the dialog-independent user model, for example to the point of view of the sales division, the research division, or the production division of a company (the view mechanism in database technology may be seen as a primitive precursor of such a group-oriented user modeling scheme). Based on the assumption that an individual user inherits all beliefs and goals from the division s/he is working for, a simple user model may in such a professional environment ignore the "private" beliefs of the dialog partner, and model only the collective ones.

However, for most conversational settings, this is an oversimplified approach since often in the course of the dialog "private" beliefs and "inherited" beliefs are mixed (see Figure 9). For example, a sales representative may begin a conversation with a client by referring to a "private" belief like (32), but later on express the beliefs and goals of his employer, as in (33).

(32) Last time we met we had an excellent dinner together.
(33) This product is going to be a big seller.

Here the open question is how the system can decide which beliefs are to be regarded as "inherited", and therefore to be passed on to that part of the knowledge base which is reserved for the representation of collective beliefs, and which ones are to be stored in the individual user model. Sometimes even a human listener has difficulty coming to a decision and therefore must ask a meta-question like (34).

(34) Is this your personal opinion or an official statement of your company?

Although our current techniques of plan recognition and speech act interpretation must be greatly refined before we can cope with this problem, a future solution may have great practical importance in certain domains, e.g., in consultative and argumentative dialogs and in knowledge acquisition dialogs with expert systems.

Another open question related to the previous one is: How can the system construct, maintain, and update consistent permanent models of collective beliefs on the basis of dialog sessions with individual users? At least for the sake of storage economy, it is necessary for a dialog system with heterogeneous user groups to identify consistent sets of beliefs and construct inheritance networks of beliefs. A solution to this problem must rely on techniques for machine learning, especially inductive inference and stereotype formation.

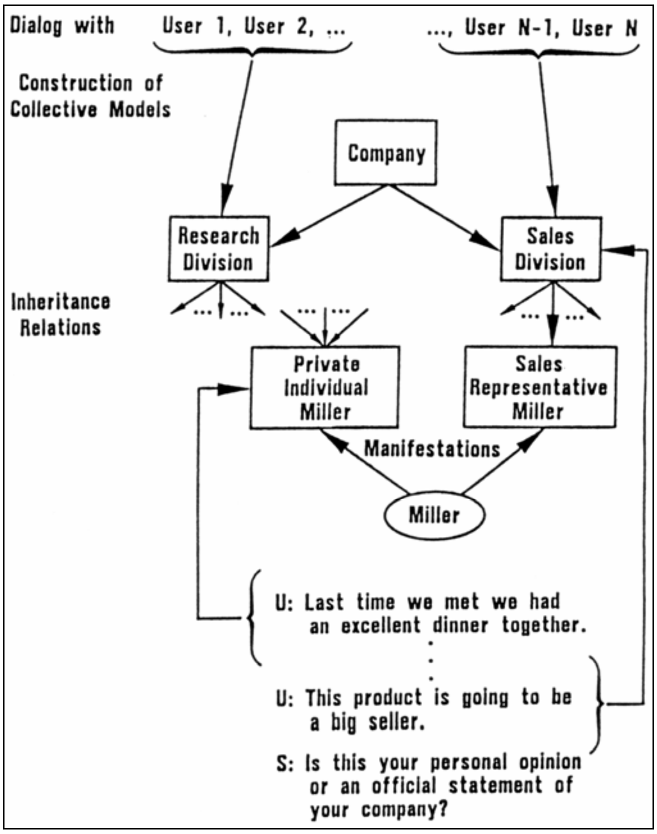

Figure 9. Copying with individual and collective beliefs in a user model

5.2. Adaptability to Diverse Conversational Settings

Today, an important trend in the design of NL dialog systems is the development of *transportable* systems. HAM-ANS (cf. [HOEP83b]), TEAM (cf. [MARP83]) and LDC [BALL84] are examples of transportable systems that can be adapted to different domains by the experts in these fields (rather than by AI or linguistics specialists).

In [HOEP83b] we proposed to go beyond domain-independence by creating dialog systems adaptable to applications that differ not only with respect to the domain of discourse, but also to dialog type, user type and intended system behavior: *transmutable systems* as we call them (see Figure 10).

Such are transportable and adaptable to diverse conversational settings and dialog games (cf. [LEVI77]). As a fictitious scenario for the application of a trans-mutable system let's suppose that the system's main task is to play the part of a sales representative for a particular company. Sometimes the sales manager interviews the system concerning its job performance. If s/he is dissatisfied with the system's performance, s/he asks an expert in sales promotion to instruct the system.

If you recall the DEC system family XCALIBUR, XSEL and XCON, you will notice that such a scenario is not too unrealistic: XCALIBUR [CARN83c] is a natural-language interface to XSEL, an automated salesman assistant which advises a customer on the selection of appropriate VAX components. XSEL then produces a sales order and passes it on to the XCON expert system for an automatic configuration of the ordered components.

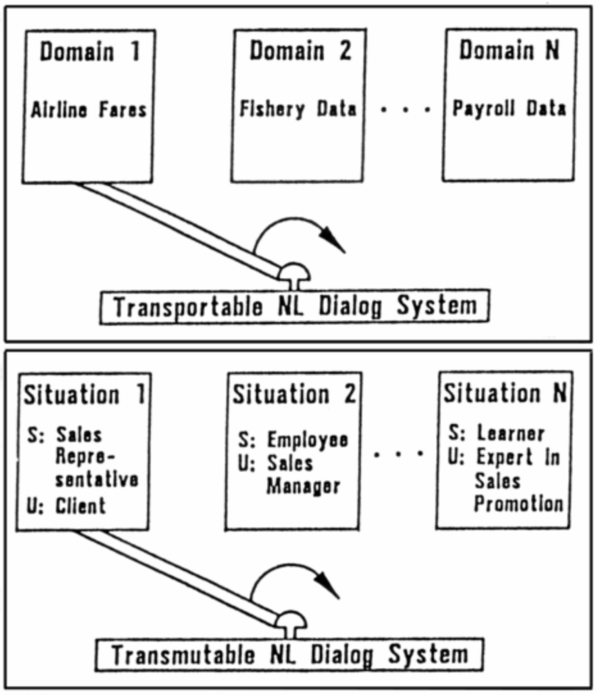

Figure 10. Transportable vs. transmutable systems

Nilsson [NILS83] proposed the development of a class of computer systems that are never to be turned off. A dialog system with continuing existence seems to provide a good framework for basic research on transmutable systems. Such a system should have a constantly changing model of its various users and should be able to adapt to diverse conversational settings and to take on various communicative roles.

The major problem that developers of transmutable systems are confronted with is the lack of a representational vocabulary for the declarative description of the relationship between the system and the user, the system's intended dialog behavior, and the associated conversational tactics. Although some of the research discussed in this volume tackles aspects of this problem, a general solution requires lengthy future research on user modeling.

5.3. User Modeling as a Resource-Limited Process

Before asking one's supervisor for a raise in salary, one usually prepares oneself for this dialog by analyzing mutual beliefs and by anticipating the various possible plans and goals of the dialog partner. But in most everyday conversations, the evaluation of complicated embedded, reflexive, or mutual beliefs and goals as well as a detailed plan analysis is a luxury which one may often not be able to afford. Thus, a computationally plausible outline of a user modeling component must provide some built-in processing restrictions that result from resource-limited real-time understanding of real-life discourse. Rosner [ROSN81] has pointed out that if one carries the user modeling approach to its extreme, there is a very good chance that nothing will get said at all (or if said, not understood) because the speaker and hearer will spend all of their time simulating each other's roles without actually doing anything else. Obviously this process must stop somewhere, but the question is where. Allen & Perrault [ALLE80], for instance, have recognized that their plan-based approach to user modeling may have to be supplemented with a rule-based one. We believe that a promising direction is the compilation of the results of a detailed plan-based analysis of speech act sequences into more efficient "shallow" conversational rules (for a first step in this direction, see the "dialog act plans" in [KOBS85a]).

A related question which so far has not been settled is at what time nested beliefs should be constructed. In the approach proposed by Perrault, Cohen and Allen, most of the possible perspectives on beliefs are already considered as computed (e.g. what one believes one's supervisor believes about one's beliefs about the supervisor's beliefs about a salary increase). Wilks [WILK83, WILK87] adopts the opposite point of view and proposes in his multiple environment approach as a least-effort hypothesis that nested beliefs should only be constructed when needed.

5.4. Prospects for Application-Oriented Research

As can be seen, research on user modeling has been very active in the last few years. A number of interesting proposals have been made concerning representation systems for user models, methods for generating assumptions about the user from his/her natural-language input, and strategies for the planning of the system's dialog contributions on the basis of its model of the user. Also, several prototype systems have been implemented in order to exemplify and further investigate these proposals.

However, a great number of fundamental problems still remain unsolved, some of which have been mentioned in the last few sections. As far as possible applications are concerned, an additional problem arises in that the development of flexible user modeling components is rather costly. Therefore, application-oriented research will probably lead in two directions:

- *Development of user modeling components in help systems for widely used existing systems*

Such a component would observe the user's interaction with the host system, form assumptions about the user's plans and experience, and provide user-specific help on request. Initial efforts along these lines are help systems for UNIX and UNK-like operating systems [WILE84, WILE86, KEMK86], a help system for the SCRIBE text editor [RICH82], and a system which provides unsolicited advice for users of the VMS operating system [FINI83].

Although work on user modeling to date has concentrated on interfaces to software systems, it should be pointed out that there are possible applications of the methods described above to many electronics-based consumer products, even photocopiers or microwave ovens. In order not to discourage the user by the system's complexity and to help him/her to take full advantage of the capabilities of such multifunctional products, future systems must include an adaptive interface based on an explicit user model. Quinn and Russell [QUIN86] describe a photocopier with an intelligent interface, which analyzes the user's task description (e. g. 'make multiple sets of double-sided copies of an article from a book'), generates a plan to attain the expressed goal, and tells the user what s/he should do next. In testing the system, the critical role of user modeling for plan execution monitoring and the interpretation of the user's behavior (e. g. his replanning after clearing an unexpected misfeed) became clear. The system's individual user model must include information about frequency and recency of particular tasks, frequency and type of successes and failures, points of confusion, and type of help requested and received. The experiments described in [QUIN86] indicate that an intelligent interface must model more than just the operations of the machine and the actions of the user. It must also model the extent of the user's knowledge and plan when to inform the user how to handle problems.

- *Development of general user modeling components*

Another possible line of research seems to be the development of general user modeling components, comparable to the development of expert system shells in the last few years. Such components might provide, for example, a representation scheme for assumptions about a user's beliefs and goals, an inference mechanism for this representation, a mechanism for detecting inconsistencies in the assumptions and for belief revision, etc. Such a shell could then be filled with domain-specific inference rules as well as domain-specific default assumptions about any user, and then be integrated into a dialog system. Beginning research in this direction is the GUMS system [FINI86, FINI*], the TRUMP system [BONA87], the BGP-MS system [KOBS88, ALLG88] and the work of [KASS87a].

6. References

[ALLE79] Allen, J. F. (1979): A Plan-Based Approach to Speech Act Recognition. TR 131/79, Dept. of Computer Science, Univ. of Toronto, Cda.

[ALLE80] Allen, J. F. and C. R. Perrault (1980): Analyzing Intention in Utterances. Artificial Intelligence 15, 143-178.

[ALLE82a] Allen, J. F. (1982): ARGOT: A System Overview. TR101, Computer Science Dept., Univ. of Rochester, Rochester, NY.

[ALLE82b] Allen, J. F. (1983): Rcognizing Intentions from Natural Language Utterances. In: M. Brady and R.C. Berwick, eds.: Computational Models of Discource. Cambridge, MA: MIT Press

[ALLE86] Allen, J. F. and D. J. Litman (1986): Plans, Goals, and Natural Language. In: Proceedings of the IEEE 74, 939-947.

[ALLE87] Allen, J. F. (1987): Understanding Natural Language. Menlo Park, CA: Benjamin/Cummings.

[ALLG89] Allgayer, J., K. Harbusch, A. Kobsa, C. Reddig, N. Reithinger and D. Schmauks (1989): XTRA: A Natural-Language Access System for Expert Systems.International Journal of Man-Machine Studies.

[APPE82b] Appelt, D. E. (1982): Planning Natural-Language Utterances to Satisfy Multiple Goals. Technical Note 259, SRI International, Menlo Park, CA.

[APPE85] Appelt, D. E. (1985): Planning English Sentences. Cambridge etc.: Cambridge Univ. Press.

[BALL84] Ballard, B., J. Lusth and N. Tinkham (1984): LDC-1: a Transportable, Knowledge-Based Natural Language Processor for Office Environments. ACM Trans, on Office Information Systems 2, 1-25.

[BOGU85] Boguraev, B. K. (1985): User Modelling in Cooperative Natural Language Front Ends. In: G. Nigel Gilbert and C. Heath, eds.: Social Action and Artificial Intelligence. London: Gower Press.

[BONA87] Bonarini, A. (1987): User Modeling in Person-Machine Conversation: A Computational Approach. In: Proc. Of COGNITIVA-87, Paris France, 377-382.

[BOTM87] Botman, A. M., A. J. Kok and L. Sikollsy (1987): Using User Models and Automatic Expertise Acquisition in Man-Machine Interaction. In: Proc. Of the 7th International Workshop on Expert Systems and their Applications, Avignon, France, 389-398.

[CARB83] Carberry, S. (1983): Tracking User Goals in an Information-Seeking Environment. Proceedings of the AAAI-83, Washington, DC, 59-63.

[CARN83c] Carbonell, J. G., W. M. Boggs, M. L. Mauldin and P. G. Anick (1983): The XCALBUR Project: A Natuaral Language Interface to Expert Systems. I: Proc. IJCAI-83, 653-656

[CARR77] Carr, B. and I. Goldstein (1977): Overlays: A Theory of Modeling for Computer Aided Instruction. AI Memo 406, MIT, Cambridge, MA.

[CHIN*] Chin, D. N.: KNOME: Modeling What the User Knows in UC. In: Kobsa, A., Wahlster, W. (eds.): User Models in Dialog Systems. Berlin: Springer, pp. 74-107.

[CLAN79] Clancey, W. J. (1979): Transfer of Rule-Based Expertise through a Tutorial Dialogue. Report STAN-CS-769, Computer Science Dept., Stanford University, Stanford, CA.

[CLOW85] Clows, I. I. Cole, F. Arshad, C. Hopkins and A. Hockley (1985): User Modelling Techniques for Interactive Systems. In: P. Johnson and S. Cook, eds.: People and Computers: Designing the Interface. Cambridge: Cambridge Univ. Press.

[COHE78] Cohen, P. R. (1978): On Knowing What to Say: Planning Speech Acts. TR-118, Dept. of Computer Science, Univ. of Toronto, Cda.

[COHE*] Cohen, R., M. Jones (1982): Incorporating User Models into Expert Systems for Educational Diagnosis. In: Kobsa, A., Wahlster, W. (eds.): User Models in Dialog Systems. Berlin: Springer, pp. 313-333.

[DEDE86] Dede, D. (1986): A Review and Synthesis of Recent Research in Intelligent Computer-Assisted Instruction. International Journal of Man-Machine Studies 24, 329-353.

[DOYL79] Doyle, J. (1979): A Truth Maintenance System. Artificial Intelligence 12, 231-272.

[DOYL80] Doyle, J. and P. London (1980): A Selected Descriptor-Indexed Bibliography to the Literature on Belief Revision. SIGART Newsletter 71, 7-23.

[DOYL83] Doyle, J. (1983): The Ins and Outs of Reason Maintenance. In: Proc. IJCAI-83, 352-354.

[FAGI85] Fagin, R. and J. Y. Halpern (1985): Belief, Awareness and Limited Reasoning: Preliminary Report. Proceedings of the 9th International Joint Conference on Artificial Intelligence, Los Angeles, CA, 491-501.

[FINI83] Finin, T. W. (1983): Providing Help and Advice in Task Oriented Systems. Proceedings of the 8th International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany, 176-178.

[FINI86] Finin, T. W. and D. Drager (1986): GUMS 1: A General User Modeling System. In: Proc. of the 6th Canadian Conference on Artificial Intelligence, Montreal, Canada, 24-29.

[FINI*] Finin, T. W.: GUMS: A General User Modeling Shell. In: Kobsa, A., Wahlster, W. (eds.): User Models in Dialog Systems. Berlin: Springer, pp. 411-430.

[GENT83] Gentner, D. and A. L. Stevens (1983): Mental Models. Hillsdale, NJ: Lawrence Erlbaum.

[GEOR83] Georgeff, M. (1983): Communication and Interaction in Multi-Agent Planning Systems. Proceedings of the AAAI-83, Washington, DC, 125-129.

[HALP85a] Halpern, J. Y. and Y. O. Moses (1985): A Guide to the Modal Logics of Knowledge and Belief: Preliminary Draft. Proceedings of the 9th International Joint Conference on Artificial Intelligence, Los Angeles, CA, 480-490.

[HAYE76] Hayes, P. J. and M. A. Rosner (1976): ULLY: A Program for Handling Conversations. Proc. of the 1976 AISB Summer Conference, Edinburgh, 137-147.

[HECK88] Hecking, M., C. Kemke, E. Nessen, D. Dengler, M. Gutmann and G. Hector (1988): The SINK Consultant: A Progress Report. Memo No. 28, Dept. of Computer Science, Univ. of Saarbrucken, Fed. Rep. of Germany.

[HINT62] Hintikka, J. (1962): Knowledge and Belief. Ithica, NY: Cornell Univ. Press.

[HOEP83b] Hoeppner, W., T. Christaller, H. Marburger, K. Morik, B. Nebel, M. O'Leary and W. Wahlster (1983): Beyond Domain-Independence: Experience with the Development of a German Language Access System to Highly Diverse Background Systems. In: Proc. IJCAI-83, 588-594.

[HORA87] Horacek, H., H. Kaindl and M. Wagner (1987): Probabilities in Game-Playing: Possible Meanings and Applications. In: B. Buchberger and J. Retti, eds.: 3. Osterreichische Artificial-Intelligence-Tagung. Berlin: Springer.

[JAME80] Jameson, A., W. Hoeppner and W. Wahlster (1980): The Natural Language System HAM-RPM as a Hotel Manager: Some Representational Prerequisites. In: R. Wilhelm (ed.): GI - 10. Jahrestagung Saarbrücken. Berlin: Springer.

[JAME82] Jameson, A. and W. Wahlster (1982): User Modelling in Anaphora Generation: Ellipsis and Definite Description. Proceedings of the 1982 European Conference on Artificial Intelligence, Orsay, France, 222-227.

[JAME83] 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. Proceedings of the 8th International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany, 616-620.

[JAME*] Jameson, A.: But What Will the Listener Think? Belief Ascription and Image Maintenance in Dialog. In: Kobsa, A., Wahlster, W. (eds.): User Models in Dialog Systems. Berlin: Springer, pp. 255-311.

[JOYC83] Joyce, C. and L. Wingerson (1983): Can We Adjust to Computer Culture? New Scientist, April 1983, 72-73.

[KAPS79] Kaplan, S. J. (1979): Cooperative Responses from a Portable Natural Language Data Base Query System. Ph.D. Thesis, Univ. of Pennsylvania, Dept. of Computer and Information Science. Also published as: Report HPP-79-19, Computer Science Dept., Stanford Univ., Stanford, CA.

[KAPS82] Kaplan, S. J. (1982): Cooperative Responses from a Portable Natural Language Query System. Artificial Intelligence 19, 165-188.

[KAPS83] Kaplan, S. J. (1983): Cooperative Responses from a Portable Natural Language Database Query System. In: M. Brady and R. C. Berwick, eds.: Computational Models of Discourse. Cambridge, MA: MIT Press.

[KASS87a] Kass, R. (1987): Implicit Acquisition of User Models in Cooperative Advisory Systems. Technical Report MS-CIS-87-05, Dept. of Computer and Information Science, Univ. of Pennsylvania, Philadelphia, PA.

[KASS88c] Kass, R. and T. Finin (1988): The Need for User Models in Generating Expert System Explanations. Technical Report MS-CIS-88-37, Dept. of Computer and Information Science, Univ. of Pennsylvania, Philadelphia, PA.

[KASS*] Kass, R.: Student Modeling in Intelligent Tutoring Systems: Implications for User Modeling. In: Kobsa, A., Wahlster, W. (eds.): User Models in Dialog Systems. Berlin: Springer, pp.386-410.

[KEMK86] Kemke, C. (1986): The SINK Consultant: Requirements, Design, and Implementation of an Intelligent Help System for a UNIX Derivative. Report No. 11, AI Laboratory, Dept. of Computer Science, Univ. of Saarbrücken, Fed. Rep. of Germany.

[KOBS84a] Kobsa, A. (1984): Three Steps in Constructing Mutual Belief Models from User Assertions. Proceedings of the 6th European Conference on Artificial Intelligence, Pisa, Italy, 423-427.

[KOBS85a] Kobsa, A. (1985a): Benutzermodellierung in Dialogsystemen. Berlin: Springer.

[KOBS85d] Kobsa, A. (1985): VIE-DPM: A User Model in a Natural-Language Dialogue System. In: J. Laubsch, ed.: GWAI-84, 8th German Workshop on Artificial Intelligence. Berlin: Springer.

[KOBS86a] Kobsa, A. (1986a): Generating a User Model from, Wh-Questions in the VIE-LANG system. To appear in the Proceedings of the 1984 GLDV Conference, Heidelberg, West Germany.

[KOBS88] Kobsa, A. (1988): User Models and Dialog Models: United They Stand... In: Computational Linguistics 14(3), Special Issue on User Modeling (A. Kobsa and W. Wahlster, eds.), Discussion Section, 91-95.

[KOBS86b] Kobsa, A., J. Allgayer, C. Reddig, N. Reithinger, D. Schmauks, K. Harbusch and W. Wahlster (1986): Combining Deictic Gestures and Natural Language for Referent Identification. Technical Report, SFB 314, Dept. of Computer Science, Univ. of Saarbrücken, West Germany (forthcoming).

[KONO80] Konolige, K. and N. J. Nilsson (1980): Multiple Agent Planning Systems. Proceedings of the AAAI-80, Stanford, CA, 138-141.

[KONO81] Konolige, K. (1981): A First-Order Formalisation of Knowledge and Action for a Multi-Agent Planning System. In: J. E. Hayes, D. Michie and Y-H Pao, eds.: Machine Intelligence 10. Chichester: Ellis Horwood.

[LEVI77] Levin, J. A. and. A. Moore (1977): Dialogue-Games: Metacommunication Structures for Natural Language Interaction. Cognitive Science 1, 395-420.

[MARB83] Marburger, H. and W. Wahlster (1983): Case Role Filling as a Side Effect of Visual Search. Proceedings of the First European ACL Meeting, Pisa, Italy, 188-195. Martin, P., D. Appelt and F. Pereira (1983): Transportability and Generality in a Natural-Language Interface System. Proceedings of the 8th International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany, 573-581.

[MARB87] Marburger, H. (1987): A Strategy for Producing Cooperative NL Reactions in a Database Interface. In: P. Jorrand and V. Sgurev, eds.: Artificial Intelligence II: Methodology, Systems, Applications. Amsterdam etc.: North-Holland.

[MARB88] Marburger, H. (1988): Generierung kooperativer natürlichsprachlicher Antworten in einem Dialogsystem mit Zugang zu relationalen Datenbanken. Ph. D. Thesis, Dept. of Computer Science, Univ. of Saarbrücken, Fed. Rep. of Germany.

[MARP83] Martin, P., D. Appelt and F. Pereira (1983): Transportability and Generality in a Natural-Language Interface System. In: Proc. IJCAI-83, 573 - 581.

[MAYS80b] Mays, E. (1980): Failures in Natural Language Systems: Applications to Data Base Query Systems. In: Proc. AAAI-80, 327-330.

[MCCO85a] McCoy, K. F. (1985a): The Role of Perspective in Responding to Property Misconceptions. Proceedings of the 9th International Joint Conference on Artificial Intelligence, Los Angeles, CA, 791-793.

[MCCO85b] McCoy, K. F. (1985b): Correcting Object-Related Misconceptions. Ph.D. Thesis, MS-CIS-85-57, Dept. of Computer and Information Science, University of Pennsylvania, Philadelphia, PA.

[MCCO*] McCoy, K. F.: Highlighting a User Model to Respond to Misconceptions. In: Kobsa, A., Wahlster, W. (eds.): User Models in Dialog Systems. Berlin: Springer, pp.233-254.

[MCKE85c] McKeown, K. R., M. Wish and K. Matthews (1985): Tailoring Explanations for the User. Proceedings of the 9th International Joint Conference on Artificial Intelligence, Los Angeles, CA, 794-798.

[MOOR80] Moore, R. C. (1980): Reasoning about Knowledge and Action. Technical Note 191, AI Center, SRI International, Menlo Park, CA.

[MORI84] Morik, K. (1984): Partnermodellierung und Interessenprofile bei Dialogsystemen der Künstlichen Intelligenz. In: C.-R. Rollinger, ed.: Probleme des (Text-) Verste-hens. Tubingen: Niemayer.

[MORI85a] Morik, K. (1985): User Modeling, Dialog Structure, and Dialog Strategy in HAM-ANS. In: Proc. of the 2nd Conference of the European Chapter of the Association for Computational Linguistics, Geneva, Switzerland, 268-273.

[MORI85b] Morik, K. and C.-R. Rollinger (1985): The Real Estate Agent - Modeling Users By Uncertain Reasoning. The AI Magazine 6, Summer 1985, 44-52.

[MORI*] Morik, K.: User Models and Conversational Settings: Modeling the User's Wants. In: Kobsa, A., Wahlster, W. (eds.): User Models in Dialog Systems. Berlin: Springer, pp.364-385.

[MURR87a] Murray, D. M. (1987): A Survey of User Cognitive Modelling. Report DITC 92/87, National Physics Laboratory, Teddington, England.

[MURR87b] Murray, D. M. (1987): Embedded User Models. In: H.-J. Bullinger and B. Shackel, eds.: Human-Computer Interaction: INTERACT'87. Amsterdam: North Holland.

[NESS87] Nessen, E. (1987): SC-UM: User Modeling in the SINK Consultant. Memo No. 18, Dept. of Computer Science, Univ. of Saarbrücken, Fed. Rep. of Germany.

[NILS83] Nilsson, N. (1983): Artificial Intelligence Prepares for 2001. The AI Magazine, Winter 1983, 7-14.

[NORM86b] Norman, D. A. and S. W. Draper, eds. (1986): User Centered System Design: New Perspectives on Human-Computer Interaction. Hillsdale, NJ: Erlbaum.

[NOVA84] Novak, H.-J. and B. Neumann (1984): Szenenbeschreibung und Imagination in NAOS. In: C.-R. Rollinger, ed.: Probleme des (Text-)Verstehens. Tübingen: Niemayer.

[POLL82a] Pollack, M. E., J. Hirschberg and B. Webber (1982): User Participation in the Reasoning Process of Expert Systems. MS CIS-82-9, Dept. of Computer and Information Science, Univ. of Pennsylvania.

[POLL86a] Pollack, M. E. (1986): A Model of Plan Inference that Distinguishes between the Beliefs of Actors and Observers. In: Proc. of the 24th Annual Meeting of the Association for Computational Linguistics, New York, NY, 207-214.

[POLL86b] Pollack, M. E. (1986): Inferring Domain Plans in Question-Answering. Ph.D. Thesis, Dept. of Computer and Information Science, Univ. of Pennsylvania, Philadelphia, PA.

[PERR80] Perrault, C. R. and J. F. Allen (1980): A Plan-Based Analysis of Indirect Speech Acts. American Journal of Computational Linguistics 6, 167-182.

[POWE79] Power, R. (1979): The Organisation of Purposeful Dialogues. Linguistics 17, 107-152.

[QUIL*] Quilici, A.: Detecting and Responding to Plan-Oriented Misconceptions. In: Kobsa, A., Wahlster, W. (eds.): User Models in Dialog Systems. Berlin: Springer, pp.108-132.

[QUIN86] Quinn, L. and D. M. Russell (1986): Intelligent Interfaces: User Models and Planners. In: Proc. of the CHF86 Conference on Human Factors in Computing Systems, Boston, MA, 314-320.

[RICH79a] Rich, E. (1979a): Building and Exploiting User Models. Ph.D. Dissertation, Computer Science Dept., Carnegie-Mellon University, Pittsburgh, PA.

[RICH79b] Rich, E. (1979b): User Modelling via Stereotypes. Cognitive Science 3, 329-345.

[RICH82] Rich, E. (1982): Programs as Data for Their Help Systems. Proceedings of the National Computer Conference, Houston, TX, 481-485.

[RICH83] Rich, E. (1983): Users are Individuals: Individualizing User Models. Int. J. Man-Machine Studies 18, 199-214.

[RICH*] Rich, E.: Stereotypes and User Modeling. In: Kobsa, A., Wahlster, W. (eds.): User Models in Dialog Systems. Berlin: Springer, pp.35-51.

[ROSE82] Rosenschein, J. S. (1982): Synchronization of Multi-Agent Plans. Proceedings of the AAAI-82, Pittsburgh, PA, 115-119.

[ROSE85] Rosenschein, J. S. (1985): Rational Interaction: Cooperation among Intelligent Agents. Ph.D. Thesis, Stanford Univ., CA.

[ROSN81] Rosner, M. (1981): Some Thoughts about Actions in Conversation. Proceedings of the Workshop on Models of Dialogue: Theory and Application. Linkoping, Sweden, 105-116.

[SOUL84] Soulhi, S. (1984): Representing Knowledge about Knowledge and Mutual Knowledge. In: Proc. Of the 10th International Conference on Computational Linguistics, Stanford, CA, 194-199.

[SCHU83b] Schuster, E. and T. W. Finin (1983b): Understanding Misunderstandings. MS-CIS-83-12, Dept. of Computer and Information Science, Univ. of Pennsylvania.

[SIDN83b] Sidner, C. L. (1983b): What the Speaker Means: The Recognition of Speaker's Plans in Discourse. Computers and Mathematics with Applications 9, 71-82.

[SLEE82c] Sleeman, D. H. and J. S. Brown, eds. (1982c): Intelligent Tutoring Systems. London: Academic Press.

[SLEE85] Sleeman, D. H. (1985): UMFE: A User Modelling Front End Subsystem. Int. J. Man-Mach. Comm. (in press).

[SPAR84] Sparck Jones, K. (1984): User Models and Expert Systems. Technical Report No. 61, Computer Laboratory, University of Cambridge, Cambridge, England.

[SPAR*] Sparck Jones, K.: Realism about User Modeling. In: Kobsa, A., Wahlster, W. (eds.): User Models in Dialog Systems. Berlin: Springer, pp.336-363.

[SPRE87] Sprenger, M. (1987): Konzeption des Partnermodells in WISBER. Technical Report No. 19, Project WISBER, Computer Science Dept., Univ. of 1 Fed. Rep. of Germany.

[SPRE88] Sprenger, M. (1988): Interpretation von Modalverben zur Konstruktion von Partnermodelleinträgen. Memo No. 18, Project WISBER, Computer Dept., Univ. of Hamburg, Fed. Rep. of Germany.

[VARD85] Vardi, M. Y. (1985): A Model-Theoretic Analysis of Monotonic Knowledge. Proceedings of the 9th International Joint Conference on Artificial Intelligence, Los Angeles, CA, 509-512.

[WAHL83] Wahlster, W., H. Marburger, A. Jameson and S. Busemann (1983): Over-Answering Yes-No-Questions: Extended Responses in a NL Interface to a Vision System. Proceedings of the 8th International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany, 643-646.

[WAHL84] Wahlster, W. (1984): Cooperative Access Systems. Future Generation Computer Systems 1, 103-111.

[WAHL86b] Wahlster, W. and A. Kobsa (1986): Dialog-Based User Models. In: Proc. of the IEEE-74 (Special Issue on Natural Language Processing), 948-960.

[WAHL88] Wahlster, W. (1988): Distinguishing User Models from Discourse Models. In: Computational Linguistics 14(3), Special Issue on User Modeling (A. Kobsa and W. Wahlster, eds.), Discussion Section, 101-103.

[WEBB83] Webber, B. L. and E. Mays (1983): Varieties of User Misconceptions: Detection and Correction. In: Proc. IJCAI-83, 650-652.

[WERN88] Werner, E. (1988): Toward a Theory of Communication and Cooperation for Multiagent Planning. In: Proc. of the 1988 Conference on Theoretical Aspects of Reasoning about Knowledge, Monterey, CA (to appear).

[WILE84] Wilensky, R., Y. Arens and D. N. Chin (1984): Talking to UNIX in English: An Overview of UC. In: Communications of the ACM 27, 574-593.

[WILE86] Wilensky, R., Y. Arens and D. Chin (1984): Talking to UNIX in English: An Overview of UC. In: CACM 27, 574-593.

[WILK83] Wilks, Y. and J. Bien (1983): Beliefs, Points of View and Multiple Environments. Cognitive Science 7, 95-119.

[WILK87] Wilks, Y. and A. Ballim (1987): Multiple Agents and the Ascription of Belief. In: Proc. IJCAI-87, 119-124.