Handbook of Human-Computer Interaction
Second, completely revised edition
M. Helander, T.K. Landauer, P. Prabhu (eds.)

Chapter 3

Mental Models and User Models

Robert B. Allen
Bellcore
Morristown, New Jersey, USA

3.1 Introduction

The expectations a user has about a computer's behavior come from *mental models* (Figure 1); while the "expectations" a computer has of a user come from *user models* (Figure 2). The two types of models are similar in that they produce expectations one "intelligent agent" (the user or the computer) has of another. The fundamental distinction between them is that mental models are inside the head while user models occur inside a computer. Thus, mental models can be modified only indirectly by training while user models can be examined and manipulated directly.

3.2 Mental Models

Models are approximations to objects or processes which maintain some essential aspects of the original. In cognitive psychology, mental models are usually considered to be the ways in which people model *processes*. The emphasis on process distinguishes mental models from other types of cognitive organizers such as schemas. Models of processes may be thought of as simple machines or transducers which combine or transform inputs to produce outputs. While some discussions about mental models focus on the representation, the approach here considers mental models as the combination of a representation and the mechanisms associated with those representations (see Anderson, 1983).

A mental model synthesizes several steps of a process and organizes them as a unit. A mental model does not have to represent all of the steps which compose the actual process (e.g., the model of a computer program or a detailed account of the computer's transistors). Indeed, mental models may be incomplete and may even be internally inconsistent. The representation in a mental model is, obviously, not the same as the real-world processes it is modeling. The mental models may be termed *analogs* of real-world processes because they incorporate some, but not all, aspects of the real-world process (Gentner and Gentner, 1983). Mental models are also termed *user's modes* (Norman, 1983) although the expression is avoided here because of confusion with the term "user models" (Section 3.3).

Because they are not directly observable, several different types of evidence have been used to infer the characteristics of mental models:

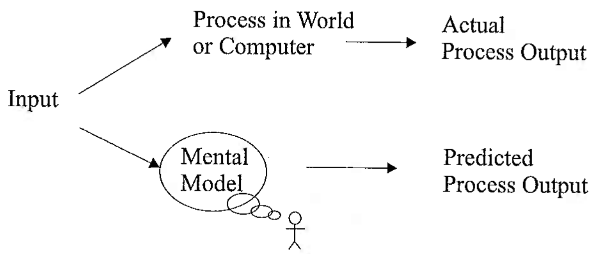

Figure 1. Process and mental model of that process.

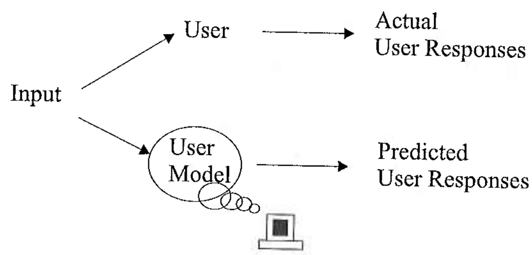

Figure 2. User responses and user model predicting those responses.

- **Predictions:** Users can predict what will happen next in a sequential process and how changes in one part of the system will be reflected in other parts of the system. However the most informative aspect of their predictions are often the errors from the model.

- **Explanations and Diagnosis:** Explanations about the causes of an event and diagnoses of the reasons for malfunctions reflect mental models.

- **Training:** People who are trained to perform tasks with a coherent account (i.e., a conceptual model, see below) of those tasks complete them better than people who are not trained with the model.

- **Other:** Evidence is also obtained from reaction times for eye movements and answering questions about processes.

3.2.1 Conceptual Models

Because mental models are in the user's head, it is helpful to have models of mental models in order to discuss them. These models of mental models may be termed *conceptual models*. Several classes of conceptual models may be identified:

Metaphor: Metaphor uses of the similarity of one process with which a person is familiar to teach that person about a different process (Carroll, 1988). For instance, a filing cabinet for paper records may be used to explain a computer file system. Indeed, the metaphor may be built into the interface (as in the use of filing cabinet icons). The ways in which a metaphor is incorporated into a mental model are difficult to examine and probably vary greatly from user to user. Moreover, a metaphor can be counterproductive because the metaphor is rarely a perfect match to the actual process and incorrect generalizations from the metaphor can result in poor performance on the task (Halasz, 1983). For instance, if word processing is introduced by analogy to typing on paper pages, word-wrapping on the screen makes little sense.

Surrogates: Surrogates are descriptions of the mechanisms underlying the process. For a pocket calculator, surrogate models would describe its function in terms of registers and stacks. As noted by Young (1983), surrogate models are not well suited to describing user-level interaction. For example, they provide poor explanations for the learnability of a system (e.g., how to use a calculator to do simple arithmetic operations).

Mappings, Task-Action Grammars, and Plans: Another class of conceptual model describes the links between the task the users must complete and the actions required to complete those tasks. Several models have this property. Young (1983) describes task-action *mappings*, which are simple pairings between tasks and actions. He compares this with the surrogate model (above) for describing the performance on simple tasks with three calculator architectures. In distinction to surrogates, he claims that mappings are suitable for describing learnability and as a basis for design.

Grammars are of interest because of their ability to describe systematic variations of complex sequences. Specifically, grammars computer dialogs are often described as a type of linguistic interaction. Grammars can be designed which specify actions as terminal nodes. TAG (Task-Action Grammars, Payne and Greene, 1986) is a framework for describing the grammars of specific command languages. It consists of definitions of tasks, categorization of tasks, and rules with schemas for combining tasks.

Planning models can also integrate tasks and actions. The GOMS model (Goals, Operators, Methods, and Selection; Card et al., 1983; Kieras, this volume) is a plan-based model which has been used widely to describe complex computer-based tasks.

Propositional Knowledge: The process assumptions of conceptual models not always explicit. Johnson-Laird (1983) has proposed that *propositional knowledge* (which he terms a "mental model") is the basis for most logical reasoning. However, his thesis has little direct application to human-computer interaction and it is not considered further here.

3.2.2 Mental Models for Computing Systems and Applications

Although mental models have been studied in physics and mathematics, the vast majority of research on them has been based on computer-human interaction. Many aspects of human-interaction with computers involve complex processes, thus people who interact with computer systems must have some type of mental model of those processes. There are several levels of processes in computing about which a person might build a model; these include a computer model such as the hardware, the operating system, software, and applications (such as text editors and spreadsheets). Indeed, an unresolved theoretical issue is whether it is possible to have multiple mental models active simultaneously and, if so, how they might interact.

Mental Models and Computing Systems: Computing systems are complex and their use and maintenance requires elaborate planning. Tasks such as system or network administration involve understanding many different subsystems. Users of computer systems also have mental models of those systems, although presumably these are very simple models compared to the system administrators. For instance, a user might have a (largely incorrect) mental model that Internet response times are based on the physical distance email messages have to travel. As noted above, training based on conceptual models about processes can lead to improved performance on tasks requiring an understanding of those processes.

Mental Models and Computing Applications: Users of computer applications have mental models of the effects of commands in operating these computing applications. For instance, users have expectations about the effects of entering values in using a spreadsheet. Halasz and Moran (1983) compared the effects of styles of training about simple calculators. Students given task-focused training were better at simple tasks than students trained with conceptual models of the calculator. On the other hand, the students trained with conceptual models were better at tasks that went beyond the initial training.

Borgman (1986) compared two styles of training on a command-based information retrieval system. In one type of training a conceptual model (what she called a "mental model") was used an analogy between the retrieval system and a traditional card catalog. The other training style was giving examples of how specific procedures would be accomplished. As in Halasz' study, the users trained by analogy performed better on tasks that required inferences beyond what was covered in the training.

When a user attempts to apply knowledge from a mental model for one task to another task, the *transfer* may show synergy or conflict. For instance, Douglas and Moran (1983) report that users familiar with traditional typewriters had more difficulty learning about electronic word processors than others. Mental models of "experts" may actually be counterproductive during transfer if they are not relevant to the task at hand. Singley and Anderson (1985) compared transfer among multiple text editors. The transfer was modeled by ACT (Anderson, 1983) in terms of the overlap in the number of rules needed to complete tasks with each of the text editors.

Designer's Mental Models: Design of complex systems requires mental models (Simon, 1969). Computer-related design tasks (as well as related design tasks such as the design of video-games, educational applications, and CD-ROMs) may involve the interaction of several different mental models. These may include models of the capabilities of the tools, models of the partially completed work and models of the user's interests and capabilities (Fischer, 1991).

Effective programmers seems to have a conceptual model of their programs as a machine for transforming inputs to outputs. Littman et al. (1986) compared two groups of programmers in a program debugging task. One group was instructed to attempt to systematically understand the program while fixing bugs. The other group was instructed to fix the bugs without attempting to form an overview of the programmer's function. The members of the first group were much better at understanding the interaction of the components of the program and they were also better at fixing the bugs.

3.2.3 Structuring Content to Improve Mental Models

The most important practical application of understanding students' mental models is for training. This section explores the issues in enhancing mental models of students with complex training programs. Section 3.3.11 examines interactive student models in which the students' knowledge is modeled to improve the performance, tutoring, and help systems.

Selection of appropriate text and graphics can aid the development of mental models. For instance, parts of a document could be highlighted to emphasize their relation to a particular concept (e.g., Kobsa et al., 1994). Training material about dynamic processes may include diagrams and other techniques for improving the learner's mental models. Hegarty and Just (1993) and Kieras (1988) have examined the optimal level of realism to present in schematic diagrams.

Beyond the local effects of media on mental models, the organization of the entire content of a manual or a course may be designed to improve the development of the user's mental models. *Scaffolding* is the process of training a student on core concepts and then gradually expanded the training. The "training wheels" approach (Carroll, 1990) is a type of scaffolding for training about computer systems.

Animation of data or scenarios which evolve over time should be especially useful for developing mental models because the causal relations in a process can be clearly illustrated. Gonzalez (1996) examined many properties of animations and found that factors such as the smoothness of the transitions were important for performance of tasks which had been presented with the animations. Because performance improved, it may be assumed that the mental models are also improved.

Interaction with a virtual environment can allow users to focus on those topics with which they are least familiar. The utility of organizers for improving the recall of information is well established. Lokuge et al. (1996) used this broad sense of mental models to develop a representation of the relationships among tourist sights in Boston. They then built a hypertext presentation which aids in the presentation of those relationships to people unfamiliar with Boston.

3.3 User Models

As indicated in Figure 2, the user model is the model computer software has of the user. In the following sections, several issues for user modeling are discussed (Sections 3.3.1-3.3.6) and then application areas are examined (Sections 3.3.7-3.3.11).

3.3.1 User Model Inputs

User models have parameters which can distinguish users. Sometimes these are set explicitly by the user and sometimes they are inferred by the computer from the user's past responses and behavior.

Explicit Profiles: In some user modeling techniques, users must create a profile of their interests. For example, in the information filtering technique know as *Selective dissemination of information* (SDI, Section 3.3.8), users must specify what terms match their interests. However, users may not have a clear memory of preferences or may not want to give an honest response. In addition, performance will be better if the user understands the model enough to select the discriminative terms (i.e., has a mental model of the user model mechanism). To some degree all entries in an explicit user profile are a type of self-categorization.

Inferences from User Behavior: An unobtrusive recording of movie preferences might simply collect information from a set-top box what movies a person had the set tuned to and how long the set stayed turned to that movie. In addition, assumptions are necessary to interpret this type of data. It is not safe to assume that a user is looking at their TV screen all the time, thus the amount of time a video is displayed on that screen may not be an accurate measure of the person's interest in that material. On the other hand, this type of data often has considerable value. Morita and Shinoda (1994) found a positive correlation between the amount of time a person spent look at a document and the ratings of interest in that document.

3.3.2 Degree of Personalization

User Models may be personalized to different degrees. They may range from baserate predictions to totally individualized predictions. All of these models should be better than random predictions.

Baserate Predictions: A *baserate* prediction is what would be expected for any individual (Allen, 1990). Baserate models might not even be considered user models since they are not differentiated across individuals. One example is of a baserate predication could follow statistical norms (e.g., that a person would like a

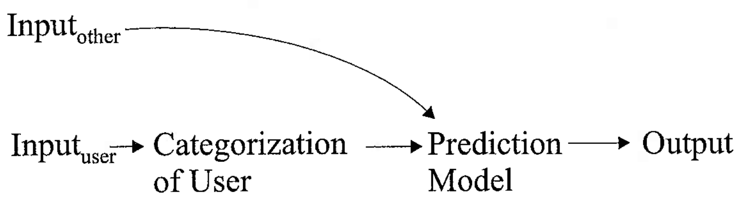

Figure 3. Categorization of user input.

best-selling book). Other types of baserate prediction could be derived from laws or social norms. In some cases, when the population is very consistent, baserates may be very accurate. In other cases, they may apply to only a small portion of the population.

User Categorization: Rather than developing separate models for each individual, users are often categorized and models developed based on those categories (as illustrated in Figure 3). Of course, the complexity of the categories may vary greatly from simple demographics (e.g., age) to complex personality constructs. There are many examples of categorization in user modeling. One example is the classification of utterances according to their function as *speech acts* (Section 3.3.10).

Another example of a user categorization is the novice-expert distinction. It might be expected that there would be consistent differences in the behavior of experts and novices which could be useful, for instance, in constructing different types of training. While there are clearly differences in the knowledge of novices and experts on a given task (e.g., Egan, 1988; Mayer, 1988), difficulties arise when attempting to classify people as novices or experts or in attempting to generalize expertise in one area to expertise in other areas. There are many dimensions of expertise and although users may be an expert on one set of commands, they may be novices in other areas. Thus, broad classification of users as experts or novices does not often seem to be helpful.

Still another example of categorization of users is a *stereotype* (Rich, 1989). Rich's "Grundy" system predicted preferences for fiction books, primarily using a user's self-descriptions (e.g., feminist) with adjectives as inputs. Unfortunately, it was difficult to tell whether Grundy's predictions were better than simple baserate predictions. To provide control conditions for a Grundy-like system for predicting book preferences, baserate predictions were found to make significantly better predictions than a random selection and a simple 'male/female' dichotomy further improved the predictions (Allen, 1990).

Kass and Finin (1991) describe GUMS (Generalized User Modeling System) in which they introduce the notion of hierarchical stereotypes and inference mechanisms based on that those stereotypes. This formalism is helpful for reasoning about users; however, it seems removed from the usual psychological notion of a stereotype. For instance, GUMS stereotypes include 'rational agents' and 'cooperative agents'. In addition, it is not clear that the stereotypes people have can be combined with or derived from other stereotypes in a systematic way.

3.3.3 User Models in Complex Systems

Blackboard Systems: A user model may be a component of more a complex system which includes other components. In some cases, a user model is said to contain all the knowledge a system has about the user. For instance, Wahlster and Kobsa (1989, p. 6) state: "A user model is a knowledge source in a NL dialog system which contains explicit assumptions on all aspects of the user that may be relevant to the dialog behavior of the system". In other cases, that knowledge is subdivided into several specific models such as task models and situation models. In training systems (Section 3.3.11), the "student model" holds task-relevant state and the "user model" applies to long-term knowledge about the user such as demographic information. This situation is similar to the suggestion that several different mental models may be active for a programming task (Section 3.2.2).

Figure 4 shows a typical collection of knowledge sources which includes user, task, and situation. The inputs are combined with data from various repositories on a blackboard. As described in the previous section, the *task expert* has information about what the user is trying to accomplish and possible strategies for accomplishing those goals. The *situation expert* contributes knowledge about the environment in which the

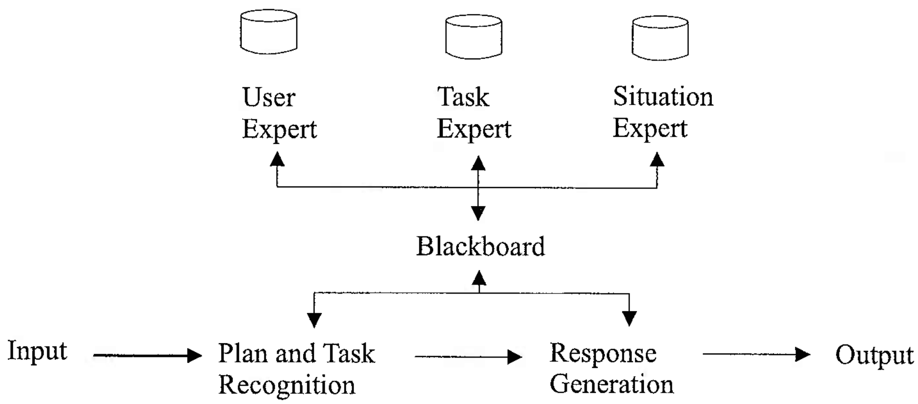

Figure 4. Typical components of a blackboard system.

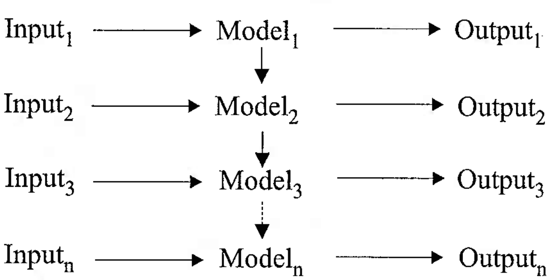

Figure 5. Adaptation of model across sequential events.

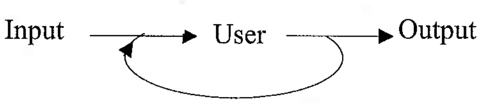

Figure 6. Model with feedback.

user is trying to complete the task. There are many possible ways of organizing the knowledge required for this type of system. Other components which have been proposed include *system, domain,* and *discourse* processors. Clearly, it is possible to partition these models in many different ways and simply proposing different sets of models is not necessarily helpful. Modularity is generally a useful design principle. Unfortunately, information about users may be very tangled and inconsistent. It may be difficult to separate the user model from the other models components. Even for Figure 3, it seems somewhat arbitrary to declare that the user model is just the categorization stage and does not include the Prediction Model.

User Agents: An *agent* may be considered as just one module of a complex system (e.g., one of the experts in Figure 4). However, a more interesting sense of the term "agent" is as modular system which acts on behalf of the user. A user agent might simply provide information about the user. For instance, it might search a database of the user's writings to answer a question on behalf of the user. An active user agent may negotiate on behalf of the user. This mean that user agents must have a model of the relative values of alternative outcomes for the user.

3.3.4 Adaptive Models, Machine Learning, and Feedback

User models are often said to *adapt* to users. However, there are different senses in which a model may be adaptive. In the simplest sense, a model is adaptive if it gives different responses to different categories of users. A more interesting sense is that a model adapts as it gains experience with an individual user. Adaptation may be within connected sequences of events or across different events. Figure 5 illustrates a model which keeps state across a sequences of inputs such as a conversation, information retrieval session, or task.

As indicated in Figure 6, *feedback* uses output from the model to refine it. This would be most common for models in which adjustments are made across a sequence of events for future trials. Most often feedback is an automatic process, however, having users refine their profiles could also be considered a type of feedback. The idea of feedback originated with control theory in which only a few parameters in the model would typically change; however, in computer models, the structure (i.e., the program) of the model itself may change.

User models which change and improve their performance as a result of greater experience are said to show *machine learning*. Some examples of machine learning algorithms are neural networks, genetic algorithms, clustering, and case-based reasoning. An important distinction is whether all training examples are maintained by the model (as in case-based reasoning) or whether other data are compressed to form a *representation* of the original training set.

3.3.5 Tasks and Planning

User models are often distinguish according to whether they apply to "short-term" or "long- term" interactions, however, this blurs several issues of adaptation and feedback. Typically, short-term models describe the user performance on specific tasks.

Tasks and Task Models: The task in which a person is engaged and plausible strategies for completing it greatly constrain the behavior of that person. Indeed, the task in which a person is engaged is often more important than individual differences for predicting user behavior (see Mischel, 1968). The interaction of tasks with behavior is so strong that there is a tendency to identify tasks or even to define new tasks to explain ambiguous behaviors. On the other hand, there are many cases in which individual differences have a major effect (Egan, 1988).

A *task model* (Sleeman, 1982) is a description of a task as well as strategies for completing that task. In some cases, the task model may include incorrect methods for completing the task. In situations when tasks are clearly defined, mental models and user models reflect the task structure. For instance, student models (Section 3.3.11) are typically focused on a student completing specific tasks.

Planning and Plan Recognition: Planning often proceeds as a cycle of determining goals or subgoals and finding methods for completing those goals. GOMS (Section 3.2.1; Kieras, this volume) is a planning model of how people complete command sequences. There are several different types of planning mechanisms and the nature of planning has been widely discussed recently. Some important dimensions for planning are whether the plans are fixed once they have been established or whether they may be modified depending on the situation.

When users complete a task they, presumably, have strategy for what they are doing. However, a human being or computer with whom they are interacting many not know what strategy the users have or even what task they are trying to accomplish. The process of *plan recognition* would involve identifying the task and the strategy. Although Figure 5 illustrated a user model in which the model is known, the figure might also be applied to plan recognition in which an observer must infer the model (i.e., the plan) and maybe the inputs (i.e., the task) given the outputs and a few contextual clues. For instance, a person who comes to a train ticket window is likely to be asking for information about trains but could, instead, be asking for other types of information. As suggested by J. Allen (1983, p. 108), "[People] are often capable of inferring the plans of other agents that perform some action."

A wide variety of techniques has been proposed for plan recognition. Most often some type of grammar is fit to responses although statistical methods such as neural networks could also be applied. Plan recognition may benefit from information about the user. For instance, knowing accents and idiosyncratic expressions may be useful for speech utterance understanding systems. The closely related problem of interpreting a student's strategy in solving a problem is an essential component of intelligent tutoring systems (Section 3.3.11).

Another tradition of inferring information about people from the context of their actions is *attribution theory* (Kelley, 1963). This describes the processes by which people make inferences about internal state of other people. For instance, *social norms* might be employed in assessing how a person might be expected to be behave in a given situation. A person who behaves differently from the norm is likely to be judged as showing an *intention* for that action.

3.3.6 Evaluation of User Models

Evaluation Criteria: The main criterion for the *effectiveness* of a user model is in predicting important behavior which facilitates the user's activities. Among the components contributing to this are:

- *Relevance* requires that models make predictions that apply to the target behavior or user goals.

- *Accuracy* requires that the models make correct predictions for at least one task and situation.

- *Generality* of the model requires robustness despite changes in tasks, situations, and users. In addition, the model should be *scaleable* with increased numbers of tasks, situations and users. Too many existing systems have been tested only on "toy" data sets.

- *Adaptability* of the model requires that it not be *brittle* in response to changes in user behavior.

- *Ease of development and maintenance* is whether the effort in maintaining the user model is worthwhile for the user. A typical factor is whether the inputs are collected unobtrusively. For instance, if the model has to be maintained by an individual, is it clear to the user how to do that?

- *Utility*: The model should improve the user's behavior. For instance, adding new commands to an interface based on a models might confuse the user.

Evaluation Techniques: While the ultimate test of a user model is how well it satisfies the criteria listed above, it is often difficult to test a model as a whole in field conditions. However, to test parts of it in limited environments. For a given set of inputs it is worth choosing the model which give the best predictions. Another standard for measuring the performance of a user model is to compare it to the performance of a human being who is given the same inputs. This is a control condition for the model. It may be called a *Turing control* condition because it is a limited type of Turing Test (Allen, 1990).

The extensive literature in psychology on techniques for assessing individual differences and personality types can be applied to the evaluation of user models. Standard assessment criteria such as *reliability* and *validity* can be adopted. Reliability is measured as the consistency by which different samples of a person's behavior are classified the same. Validity is the agreement of a model's classification with evidence from other aspects of behavior. *Incremental validity* (Mischel, 1968) is the use of the simplest set of independent variables and inference mechanisms to generate classifications.

Social Implications of User Models: Beyond the evaluation of the performance of user models, they may be evaluated in terms of their effect on individuals in society and on society itself.

Privacy may be compromised by having large amounts of detailed information about individuals stored on-line. Private information may be breached and used in unauthorized ways. An alternative is not to store sensitive information about a person but only categorical information and demographics. Another possibility is to have the most personal details of these systems stored locally and under user's control. For instance, an agent acting on behalf of the user could have access to the data and respond to inquires.

If agents (Section 3.3.3) make decisions on behalf of human beings based on user models, then the users may be removed from those actions and may not feel *responsible* for the actions of their agents. Eventually laws are likely to be developed to clarify the extent of responsibility a person has for agents acting on their behalf.

Effective user modeling may greatly affect the behavior of individuals and their *relationship to society*. For instance, information services can be tailored for individuals. Paradoxically, however, the ability to personalize highlights the similarity among people.

3.3.7 User Models for Supporting Completion of Routine Tasks

Streamlining Routine Command Sequences: It is widely recognized that the completion of tasks in HCI follows regular patterns. Several proposals have been made for systems which learn patterns of computer commands by purely statistical regulation, or inference underlying tasks (e.g., Hanson et al., 1984). From a different approach, Desmarais et al. (1991) apply plan recognition to parsing user actions in order to recognize inefficient user behavior and to suggest shortcuts. "Eager" is a programming-by-example system (Cypher, 1991) which attempted to anticipate and support the placement of widgets by GUI designers by observing patterns of their responses.

Agents and Routine Office Tasks: User agents have been applied to routine office tasks such as scheduling appointments on an electronic calendar and prioritizing email. Typically, these agents do not model goals and complex chains of events. Because the tasks being modeled are repetitive, there is sufficient data to apply machine learning to find salient attributes. The agents also often adapt across users and across tasks for a given user.

Dent et al. (1992) describe a learning apprentice and apply it to personal calendar management. The system uses decision trees to characterize previous calendar scheduling. For example, it created a rule that theory seminars were held on Mondays. After training, the model performed slightly better than rules hand coded by the researchers. Maes and Kozerick (1994) describe two interfaces with agents: an email assistant and a calendar scheduling assistant. The agents predicted meeting schedules with an inference mechanism known as case-based reasoning. Small improvements in the confidence level reported by the agents' when their predictions when those predictions were correct.

3.3.8 User Models for Information Access

Information Retrieval and Information Filtering: The most widely studied approach to information needs is *information retrieval* (IR) (Dumais, 1988). The usual paradigm in information retrieval is to find documents which match a user query. This requires representations of the document collection and the query as well as algorithms for comparing the two representations. The algorithm for comparing queries to documentation may be simple keyword matches or complex derivation of verbal associations. In terms of user models, the IR algorithms integrate input from users and produces a set of relevant documents which are, ideally, those the users would have selected for themselves.

An IR model may be tuned to improve its selections by *relevance feedback*. The user identifies the most relevant documents and terms from those are used as inputs for a revised model (see Figure 6) to use in retrieving documents from the document representation. While relevance feedback is usually thought of as refining a query this may also be seen as developing a user model (Figure 5). For instance, Foltz and Dumais (1992) represented users' interests as a points in latent-semantic indexing (LSI) space.

Although there are parallels between IR and generalized user modeling, it seems that user modeling could be used even more in IR systems. The IR task is often implicit (e.g., a research topic the user is exploring). Demographic information about the user has, typically, been little used in the IR literature (Korfhage, 1984). There have been suggestions for long-term user models as part of IR systems. For instance, a user's interest across several sessions could be recorded and used to improve responses to queries but little work has been done.

Information filtering is a variation of IR; specifically, it is a type of text categorization. Filters identify new information which matches long-term, relatively stable interests. Selective dissemination of information (SDI, Luhn, 1958) usually employs an explicit profile of keywords. In some systems, these word lists are maintained by the users and it not clear if people are good at maintaining these profiles. Feedback in SDI is accomplished by the users making changes in their profiles.

Agents (see also Sections 3.3.3 and 3.3.7) were applied to adaptive information filtering by Sheth (1994). Specifically, genetic algorithms were employed as a highly adaptive model of NetNews preferences. This agent was able to acquire initial models for several subjects. It was also able to adapt to changes in simulated preferences ratings.

Hypertext Linking: Hypertext allows users to browse text by linking information objects together. Links may be selected that would be relevant to a specific task of use or interest to a given user. One example of adaptive hypertext is from KN-AHS (Kobsa et al., 1994) in which, for instance, the fact that a user checks a term in a glossary is used to infer that the user is not familiar with that term and a full explanation of it is included if it is used in later text. Thus, tailoring links in a hypertext may be related to community ratings on web pages (Section 3.3.8), to language generation (Section 3.3.10) and interfaces for training (Section 3.3.11).

Graphical views of objects and links which have been frequently accessed have been described as "history-enriched digital objects" (Hill et al., 1993, also see Lesk and Geller, 1983). Typically, these show total counts (i.e., baserates).

On the Web, there is often a delay in accessing information, and it is helpful to predict what topics the user will browse next. The anticipated material can be *prefetched* so it will be available locally if the user decides to access it. Clearly, prefetching would be more efficient with accurate models of what the user is likely to access.

Information and Aesthetic Preferences: Information preferences may include movies a person would like to view or books or news articles they would like to read. Information and aesthetic preferences may be distinguished from information needs (previous sections) in not being associated with a particular task. Because the are based on easily-described tasks, it is difficult to model users' processes involved in aesthetic preferences.

Probably the most successful technique for predicting information and aesthetic preferences is known as *collaborative filtering* (Section 3.3.9) although other techniques for predicting information preferences have been examined. These include the stereotypes of the "Grundy" system described earlier (Section 3.3.2).
A content-based system which modeled music preferences was proposed by Loeb (1992). Songs were categorized and models developed for each category of song based on ratings made by the user just after the song had finished. This is, essentially, an IR relevance feedback mechanism.

Table 1. Hypothetical ratings of seven items by four users.

	1	2	3	4	5	6	Target
1	9	1	4	8	3	0	2
2	3	0	9	2	3	8	1
3	2	8	7	9	3	1	7
4	8	3	3	7	8	2	?

Table 2. Correlations of ratings among four users from Table 1.

	1	2	3	4
1	-			
2	-0.14	-		
3	+0.10	-0.40	-	
4	+0.74	-0.49	-0.11	-

Yet another proposal for predicting information preferences is based on "interestingness" (e.g., Hidi and Baird, 1986). This is, essentially, a baserate prediction and might work if there is consensus over what is interesting. However, where there are substantial individual differences, the problem of determining what is interesting seems no easier than the original task of predicting preferences.

3.3.9 Collaborative Filtering for Information Preferences

While most user models attempt to model the user's thought process, other mechanisms are possible. Recently, statistical mechanisms based on identifying individuals with similar preferences have been proposed (Allen, 1990). With collaborative filtering (also known as "community ratings" or "social learning"), there is no attempt to model the underlying cognitive process. Rather, the preferences of other individuals are employed, the other individuals integrate the inputs.

Example: While users may differ greatly in their preference ratings for different items, subgroups of users may be quite similar to each other. Table 1 shows hypothetical ratings by four users for seven documents. Table 2 shows the correlation coefficients of the ratings on a 10-point scale for the users. The ratings of Users 1 and 4 are similar. Thus, User 1's ratings could be a predictor for User 4's ratings (specifically, predicting a rating of 2 for the target item). Beyond the simple correlations, multivariate statistics, such as *multiple linear regression,* combine evidence from several different users into a single predictive model. For the data in Table 1, a multiple linear regression gives an $\mathbf{R}^2 = +\ 0.77$ which is better than any of the simple correlations for that user in Table 2.

News, Videos, Movies, and Music: Allen (1990) used collaborative filtering to make predictions about preferences for news articles, however, few of those correlations were substantial. This may have been because of the small number of participants or because news preferences are not stable and are difficult to predict.

Collaborative filtering has been more successful predicting preferences for movies. Hill et al. (1995) report studies of video preference predictions with data collected from the Internet. 291 respondents made 55,000 ratings of 1700 movies. Three of four video recommendations made by the service were highly evaluated by the users of the system. In Figure 7, individualized predictions show a much stronger correlation with the user's preferences than do the predictions of the movie critics.

Collaborative filtering has also been applied to recommend music preferences. Shardanad and Maes (1995) developed the "Ringo" system (also known as "Firefly") in which ratings were collected from users by email. Modified pairwise correlation coefficients were calculated for 1000 of the people who made ratings. Predictions of preferences based on these correlations were somewhat better than the baserate predictions.

Limitations and Extensions of Collaborative Filtering: A number of constraints determine whether collaborative filtering will be effective in real-world applications. Ratings are often used to build the filtering models. Ratings are not direct measures of the target behavior and may be affected by many incidental factors. The underlying preferences must also be fairly stable. For instance, musical tastes may change depending on time of day, mood, and what is going on around a person. The ratings a person provides must not change very much while predictions are being made. If the community is heterogeneous and if there is a wide variety of opinions about the material, a relatively large number of other people must make ratings.

Beyond the basic correlational techniques described above, a number of extensions for collaborative filtering techniques may be considered. While collabo-

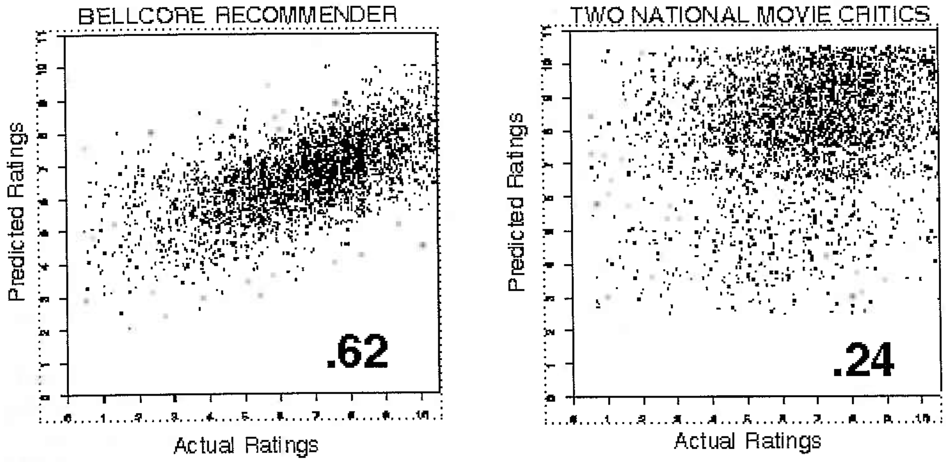

Figure 7. Video recommender scatterplots.

rative filtering is an effective technique, it seems likely that predictions with a mixture of community ratings and a limited amount of content (categorization) could be better still. Preferences in one domain (e.g., movies) could be used to make predictions across domains (e.g., books). Indeed, a generalized profile might be developed in which a wide range of a user's preferences were combined and compared with other people's preferences. In addition to matching a person with an information object, user models can also be used to match one person with another. For instance, people with similar interests might be identified based on correlations their aesthetic preference ratings. In some cases, information about users may be used to *initiate* the presentation of information to them. This would often be *targeted advertising* (i.e., personalized commercial information); however, other types of information such as health messages and community information could also be triggered.

Predictions may be made by friends of the user (Allen, 1990). In the previous section the subjects in the control condition had relatively little information about the preferences of the person for whom the predictions were to be made. Collaborative filtering might also be used to make predictions about the preferences of a group such as a family or friends attempting to find a movie which they would enjoy. If the group provides ratings as a unit, the predictions should be similar to those made for individuals. Alternatively, group prediction could be synthesized from predictions for the individuals in that group. In that case, a simple technique would be to find the item which was highest, on the average, across users. However, group dynamics such as the dominance among members of the group would be difficult to model.

3.3.10 User Models and Natural Language

Natural language processing is complex and involves many user modeling issues. This section is divided into discussion of low-level modeling issues in the speech chain and higher-level issues affecting understanding and generation. Related issues are also considered in the following section on training (Section 3.3.11) and in Section 3.3.5 under subsection on plan recognition.

Speech Chain: The activities associated with the production of speech may be said to form a *speech chain.* There are individual differences at many steps in the speech chain. These include speaker characteristics in speech recognition, word senses, complexity of vocabulary and grammar, and dialect. These differences apply to both language understanding and generation.

Speech recognition attempts to identify the words in user utterances. Because individuals differ substantially from each other in their speech, it is helpful to have a model of their characteristic speech patterns. These individualized models, known as *speaker-dependent* models, can be used to transform responses for further processing. However, a detailed review of the literature on user models for speech recognition is beyond the scope of this paper.

Neural networks were used classify gestures of a person wearing a data glove and then to produce speech based on those gestures (Fels and Hinton, 1995). The neural networks adapted to the idiosyncratic responses of the user. After 100 hours of training, a person gesturing with the data glove was able to use the system to produce speech somewhat better than a speech synthesizer.

Language Generation and Understanding: Communication between people is much more than the transmission of literal messages which can be easily deciphered with a dictionary and simple parser. Rather, communication is highly context and task dependent and both sender and receiver have complex expectations about the for the interaction. These expectations are described as *conversational maxims* (Grice, 1980). They are: Quality (truthful and accurate information), Quantity (neither more nor less information than is required), Relation (information appropriate to the task), and Manner (clear and unambiguous information).

The emphasis on the *function* of messages instead of on their literal meaning has led Austin (1962) to categorize them in terms of *speech acts*. Speech acts may be *indirect* (Searle, 1980) as in irony where is intended message is the opposite of the literal message. Purposeful conversations have a regular structure or pattern of speech acts (Winograd and Flores, 1986). For instance, we would expect a offer of help to be followed by an acceptance or rejection of that offer.

In order to engage in communication human beings may be said to have mental models of the task, the context, and the other participants. If the communication is between a human being and a computer (whether as a task-oriented dialog or as natural language dialog) then the computer may be said to have a user model of the human being. In order to fulfill conversational maxims and to initiate speech acts, in language generation, a speaker or agent must consider long-term characteristics of the listeners such as their knowledge level. In addition, the speaker may monitor the reactions of the listeners during the conversation to ensure the intended communication is received. Likewise, the listener could expect that the communicator was following conversational maxims. Recognition of speech acts by a listener may be viewed as a type of plan recognition (see Section 3.3.5). As noted earlier, plan recognition would include many aspects of user models such as the situation, the speaker's appearance, and previous experiences with the speaker.

3.3.11 User Models for Tutoring, Training, and Help Systems

Instructional interaction between a computer and a human being may be viewed as a specialized conversation. For workers using computers to complete routine tasks, the actions in which the user may be engaged are generally highly constrained. Thus, for help systems which try to give advice to those workers, little inference is about the task is needed. However, the task and the user's interpretation of it are generally less well known for training or tutoring contexts than for help system. Thus, plan recognition is an important part of training and tutoring systems.

Tutoring and training are closely related to each other although training is focused on teaching skills needed for a particular task while tutoring is applied to learning general skills such as reading and mathematics. Help systems support users who are attempting to complete tasks with a computer system. These systems differ in their style of interaction with the student. For instance, the system may interrupt the user in the middle of a task to provide advice, or it may wait for the user to request information. Similarly, the system might follow a prescribed set of tasks or it might schedule tasks for the student depending on the model of the student's knowledge.

Personalization in tutoring may be modeled by observing the conversations between tutors and students. There have been several studies of how human tutors adapt their interests to a given student and to assess the human tutors' strategies for modeling the user (e.g., Grosz 1983; Stevens, et al., 1982).

The models of users or students in training, tutoring, and help systems combine aspects of mental models, user models, and task models. *Student models* usually describe the student on a given task but generally do not include demographic information about the student. In the style of Figure 4 separate "user models" with long-term information are often included in these systems.

The key for training systems based on student models is in recognizing inefficiencies and errors in the student's performance and determining where they came from. The simplest approach to recognizing errors is to make a catalog of errors for a given task. However this is not always so easy, there may be several correct ways to solve a task.

A more general approach to recognizing errors is known as *differential modeling* (Burton and Brown, 1979). This compares a student's performance to that of an expert engaged in a similar task. The models a student has of the task may not be like a expert's models of that same task (Clancey and Lestinger, 1981). Because they are novices, students' models are generally simplified or incomplete (see Section 3.3). Indeed, the student's model may have inconsistent and even illogical strategies for completing tasks.

Determining the cause of an error can be very tricky; an error may be caused by a combination of factors. Errors may be modeled as the result of specific processes. Sleeman (1982) has proposed modeling a student's problem solving with "mal-rules" which are rules that an expert would say are false, but which reflect a student's incorrect beliefs. Similarly, Brown and Burton (1978) proposed a "theory of bugs". The "EdCoach" system (Desmarais et al's, 1993), models the task as a grammar for executing sequences of knowledge units. It then infers the student's confusions about these knowledge units by examining the methods the students actually use.

3.4 Discussion

Focusing on mental and user models in human-computer interaction highlights the intentionality of the interaction between the person and the machine. Because mental models are inside a person's head, they are not accessible to direct inspection and it is difficult to have confidence about how a mental model is constructed or how it can be modified. Indeed, a *reductionist* might assert that there are no mental models per se, but only "generalizations" from behaviorally conditioned expectations. By contrast to mental models, user models can be directly inspected and modified. Although for user models many difficult questions remain about the best way to capture and represent information about the user and the task.

Because computing is relatively cheap and widely deployed, complex networked services can increasingly be personalized Yet, few of the techniques described here are in regular use; simpler techniques non-adaptive techniques are generally preferred as being more robust. Thus, we may expect mental models and user models to expand as an area of research and innovation.

3.5 References

Allen, J. (1983). Recognizing intentions from natural language utterances. In: *Computational Models of Discourse*, M. Brady and R.C. Berwick (eds.), Cambridge, MA: MIT Press, 107-166.

Allen, R. B. (1990) User models: Theory, method, and practice. *International Journal of Man-Machine Studies,* 32, 511-543.

Anderson, J. R. (1983). *The Architecture of Cognition*, Cambridge, MA: Harvard.

Austin, J.L. (1962). *How to do Things with Words,* Cambridge, MA: Harvard.

Borgman, C. L. (1986). The user's mental model of an information retrieval system: An experiment on a prototype online catalog. *International Journal of Man-Machine Studies,* 24, 47-64.

Brown, J. S. and Burton, R. R. (1978) Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science,* 2, 155-191.

Burton, R. R., and Brown, J. S. (1979). An investigation of computer coaching for informal learning activities. *International Journal of Man-Machine Studies,* 11, 5-24.

Card, S. K., Moran, T. P., and Newell, A. (1983). *The Psychology of Human-Computer Interaction,* Hillsdale NJ: Erlbaum.

Carroll, J. M. (1990), *The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill.* Cambridge, MA: MIT Press.

Clancey, W. J. and Lestinger, R. (1981). NEOMYCIN: Reconfiguring a rule-based expert system for application teaching. *Proceedings International Joint Conference on Artificial Intelligence,* 829-836.

Croft, W. B. and Thompson, R. H. (1984). The user of adaptive mechanisms for selection of search strategies in document retrieval systems. In: *Research and Development in Information Retrieval*, C.J. vanRijsbergen (ed.), Cambridge: Cambridge University Press, 111-122.

Cypher, A. (1991). EAGER, Programming repetitive tasks by example. *Proceedings ACM SIGCHI* (New Orleans) 33-39.

Dent, L., Boticario, J., McDermott, J., Mitchell, T., and Zabowski, D. (1992). A personal learning apprentice. *Proceedings of the 10th National Conference on Artificial Intelligence*, Menlo Park, CA, AAAI Press, 96-103.

Desmarais, M. C., Giroux, L., and Larochelle, S. (1991). Plan recognition in HCI: The parsing of user actions. In: *Mental Models and Human Computer Interaction (vol.2),* M.J. Tauber and D. Ackermann (eds.), Amsterdam: North-Holland, 291-314.

Desmarais, M. C., Giroux, L., and Larochelle, S. (1993). An advice giving interface based on plan-recognition and user-knowledge assessment. *International Journal of Man-Machine Systems,* 39, 901-924.

Douglas, S. and Moran., T. P. (1983). Learning text editor semantics by analogy. *Proceedings ACM SIGCHI* (Boston), 207-211.

Dumais, S. T. (1988). Information retrieval. In: *Handbook of Human Computer Interaction,* 1st edition. M. Helander (ed.), Amsterdam: North-Holland, pp. 673-694.

Egan, D. E. (1988). Individual differences. In: *Handbook of Human Computer Interaction,* 1st edition. M. Helander (ed.), Amsterdam: North-Holland, pp. 543-568.

Fels, S. and Hinton, G. (1995). Glove-TalkII: An adaptive gesture-to-format interface. *Proceedings ACM SIGCH,* (Denver, May) 456-463.

Fischer, G. (1991). The importance of models in making complex systems comprehensible. In: *Mental Models and Human Computer Interaction* (Vol. 2), M.J. Tauber and D. Ackermann (eds.), Amsterdam: North-Holland, 3-36.

Foltz, P. W., and Dumais, S. T. (1992). Personalized information delivery: An analysis of information filtering methods, *Communications of the ACM*, 35 (12) 51-60.

Geller, V., and Lesk, M. E. (1983). User interfaces to information systems, *Proceedings ACM SIGIR* (Bethesda, June), 130-135.

Gentner, D. and Gentner, D. R. (1983). Flowing water of teeming crowds of electricity. In: *Mental Models*, D. Gentner and A. Stevens (eds.), Hillsdale NJ: Erlbaum, 99-129.

Goldstein, I. P. (1982). The genetic graph: A representation for the evolution of procedural knowledge. In: *Intelligent Tutoring Systems.* D. Sleeman and J.S. Brown (eds.), London: Academic, 59-78.

Gonzalez, C. (1996) Does animation in user interfaces improve decision making. In *Proceedings ACM SIGCHI* (Vancouver, April), 27-34.

Grice, H. P. (1975). Logic and conversation. In: *Syntax and Semantics: Vol. 3, Speech Acts*, P. Cole and J.L. Morgan (eds.), New York: Academic, 41-58.

Halasz, F. G. (1983). Analogy considered harmful, *Proceedings ACM SIGCHI*, Boston, 383-386.

Halasz, F. G. and Moran, T. P. (1982). Mental models and problem solving in using a calculator. *Proceedings ACM SIGCHI,* 212-216.

Hanson, S. J., Kraut, R. E., and Farber, J. M. (1984). Interface design and multivariate analysis of UNIX command use. *ACM Transactions on Office Information Systems,* 2, 42-57.

Hegarty, M. and Just, M. A. (1993). Constructing mental models of machines from text and diagrams. *Journal of Memory and Language,* 32, 717-742.

Hidi, S. and Baird, W. (1986). Interestingness: A neglected variable in discourse processing. *Cognitive Science,* 10, 179-194.

Hill, W. C., Hollan, J., Wroblewski, D., and McCandles, T. (1992) History-enriched digital objects. *Proceedings ACM SIGCHI* (Monterey, May), 3-8.

Hill, W. C., Stead, L., Rosenstein, M., and Furnas, G. W. (1995). Video recommender. In *Proceedings ACM SIGCHI* (Denver, May), 194-201.

Johnson-Laird, P. N. (1983). *Mental Models*, Cambridge, MA: Harvard.

Kass, R. and Finin, T. (1991). General user modeling A facility to support intelligent interaction. In: *Intelligent User Interfaces,* J. W. Sullivan and S. W. Tyle (eds.), New York: ACM Press, 111-128.

Kelley, H. H. (1971). Attribution in social interaction In: *Attribution: Perceiving the Causes of Behavior,* E E. Jones., et al. (eds.), Morristown NJ: General Learning Press, 1-26.

Kieras, D. E. (1988). What mental model should b taught: Choosing instructional content for comple engineered systems, In: *Intelligent Tutoring Systems Lessons Learned,* Psotka, J., Massey, L.D., and Mutte S. A. (eds.), Hillsdale NJ: Erlbaum, 85-111.

Kieras, D. (this volume). GOMS Model. In: *Handboo of Human Computer Interaction,* 2nd edition.M Helander, T. K. Landauer and P. Prabhu (eds.), Nort Holland: Amsterdam.

Kobsa, A., Muller, D., and Nill, A. (1994). KN-ANS An adaptive hypertext client of the user modeling sy tem BGP-MS. *Proceedings User Modeling Confe ence,* (Hyanis, MA), 99-105.

Korfhage, R. R. (1984). Query enhancement by us profiles. In: *Research and Development in Informati Retrieval,* C.J. vanRijsbergen (ed.), (Cambridge: Can bridge University Press), 122-132.

Littman, D. C., Pinto, J., Letovsky, S., and Soloway, (1986). Mental models of software maintenance. *Pr ceedings of Empirical Studies of Programme* (Washington), 80-98.

Loeb, S. (1992). Architecting personal delivery multimedia. *Communications of the ACM,* 35 (12) 3 48.

Lokuge, I., Gilbert, S. A., and Richards, W. (199 Structuring information with mental models: A tour

Boston. In: *Proceedings ACM SIGCH* (Vancouver, April), 413-419.

Luhn, H. P. (1958). A business intelligence system. *IBM Journal of Research and Development, 2,* 314-319.

Maes, P. and Kozierck, R. (1994). Agents that reduce workload and information overload. *Communications of the ACM,* 37, 31-40.

Mayer, R. E. (1988). From novice to expert. In: *Handbook of Human Computer Interaction,* 1st edition. Landauer, Helander, M. (ed.), Amsterdam: North-Holland, pp. 569-580.

Morita, M., and Shinoda, Y. (1994). Information filtering based on user behavior analysis and best match text retrieval. In *Proceedings SIGIR*, Dublin, 272-281.

Mischel, W. (1968). *Personality Assessment.* New York: Wiley.

Norman, D. (1983). Some observations on mental models, In: *Mental Models*, D. Gentner and A. Stevens (eds.), Hillsdale NJ: Erlbaum,, 7-14.

Payne, S. J., and Green, T. R. G. (1986). Task-action grammars: A model of mental representations of task languages. *Human-Computer Interaction,* 2, 93-113.

Rich, E. (1989), Stereotypes and user modeling. In: *User Models in Dialog Systems*, A. Kobsa and W. Wahlster (eds.), Berlin: Springer-Verlag, 35-51.

Searle, J. Indirect speech acts. In: *Syntax and Semantics. Speech Acts*, vol. 3. P. Cole and J. L. Morgan (eds.). New York: Academic, pp.59-82.

Self, J. (1974). Student models in CAI. *International Journal of Man-Machine Studies,* 6, 261-276.

Shardanand, U. and Maes, P. (1995). Social information filtering: Algorithms for automating "word of mouth". In *Proceedings ACM SIGCHI*, (Denver, May), 210-217.

Sheth, B. D. (1994). *A Learning Approach to Personalized Information Filtering*. MIT Media Laboratory Technical Report, 94-01.

Simon, H. A. (1969). *The Sciences of the Artificial,* Cambridge, MA: MIT Press.

Singley, M. K., and Anderson, J. R., (1985). The transfer of text-editing skill. *International Journal of Man-Machine Studies,* 22, 403-423.

Sleeman, D. (1982). Assessing aspects of competence in basic algebra. In: *Intelligent Tutoring Systems.* D. Sleeman and J. S. Brown (eds.) London: Academic, 185-199.

Stevens, A., Collins, A., and Goldin, S. (1982). Misconceptions in students' understanding. In: *Intelligent Tutoring Systems*, Sleeman, D. and Brown, J.S. (eds.), New York: Academic Press, 13-24.

Wahlster, W. and Kobsa, A. (1989). User models in dialog systems. In: *User Models in Dialog Systems*, A. Kobsa and W. Wahlster (eds.), Berlin: Springer-Verlag, 4-34.

Winograd, T., and Flores, F. (1986), *Understanding Computers and Cognition.* Norwood, NJ: Ablex.

Young, R. M. (1983). Surrogates and mappings: Two kinds of conceptual models for interactive devices. In: *Mental Models*, D. Gentner and A. Stevens (eds.), Hillsdale NJ: Erlbaum, 35-52.