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:

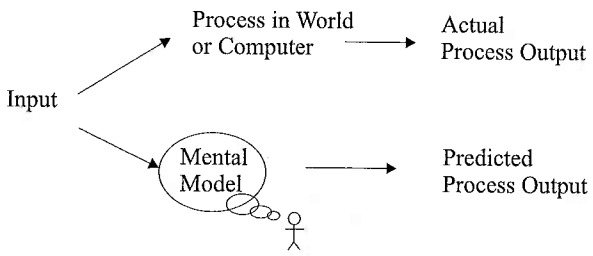

Figure 1. Process and mental model of that process.

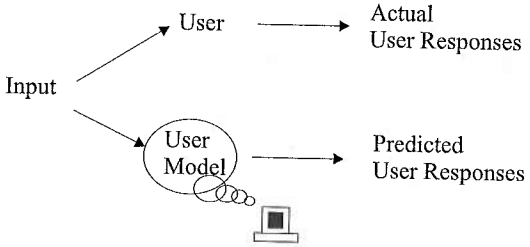

Figure 2. User responses and user model predicting those responses.

- **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

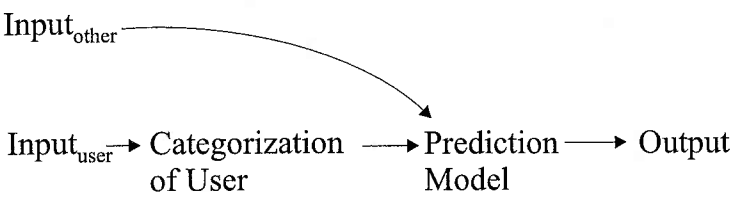

Figure 3. Categorization of user input.

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

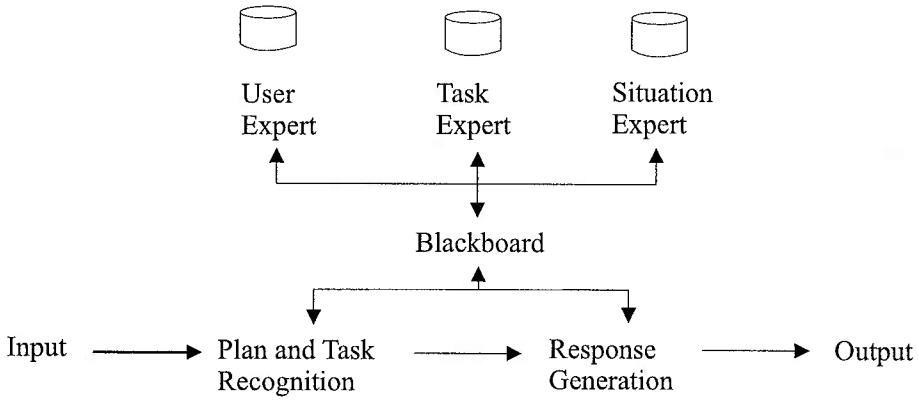

Figure 4. Typical components of a blackboard system.

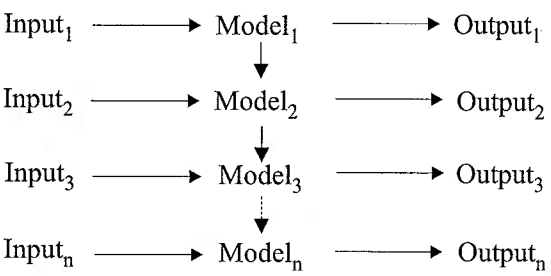

Figure 5. Adaptation of model across sequential events.

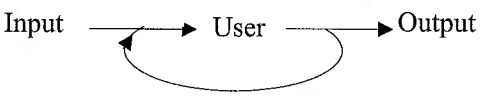

Figure 6. Model with feedback.

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 per-

formance 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.