To appear in *User Modeling and User-Adapted Interaction, 5,* 1996, in a special issue on Numerical Uncertainty Management in User and Student Modeling.

Numerical Uncertainty Management in User and Student Modeling: An Overview of Systems and Issues

ANTHONY JAMESON

Department of Computer Science, University of Saarbrücken
P.O. Box 151150, D-66041 Saarbrücken, Federal Republic of Germany
jameson@cs.uni-sb.de

Abstract. A rapidly growing number of user and student modeling systems have employed numerical techniques for uncertainty management. The three major paradigms are those of Bayesian networks, the Dempster-Shafer theory of evidence, and fuzzy logic. In this overview, each of the first three main sections focuses on one of these paradigms. It first introduces the basic concepts by showing how they can be applied to a relatively simple user modeling problem. It then surveys systems that have applied techniques from the paradigm to user or student modeling, characterizing each system within a common framework. The final main section discusses several aspects of the usability of these techniques for user and student modeling, such as their knowledge engineering requirements, their need for computational resources, and the communicability of their results.

Key words: numerical uncertainty management, Bayesian networks, Dempster-Shafer theory, fuzzy logic, user modeling, student modeling

1. Introduction

When we make inferences about the beliefs, abilities, motives, and future actions of other persons, we have to manage a good deal of uncertainty. Within social psychology, a large community of researchers has investigated the way people handle these challenges in everyday life and the nature of the errors that they make (see, e.g., Nisbett & Ross, 1980; Fiske & Taylor, 1991).

In the field of psychological assessment, for decades techniques have been explored and applied for making such inferences under controlled conditions (see, e.g., Wainer, 1990). Even when it is possible to process numerous observations of a person, carefully chosen and interpreted with reference to an extensive empirical database, the task of dealing with the uncertainty associated with the evidence is challenging.

For interactive software systems that attempt to model a user or a student, the gap between the nature of the available evidence and the conclusions that are to be drawn is often much greater. Such systems in general have more meager and/or more haphazardly collected data about their users than can be obtained by a person who is engaged in face-to-face interaction or by a tester who is in control of the situation. Moreover, the systems can less often fall back on a rich background of relevant experience.

Until the late 1980's, researchers interested in user or student modeling had available only a limited repertoire of techniques for uncertainty management. They mostly had to rely either on poorly understood ad hoc techniques or on general techniques—such as various forms of default reasoning—that were not really well suited for the treatment of most problems in this area.

Fortunately, the question of how to manage uncertainty has been a rapidly expanding and increasingly mainstream research topic in artificial intelligence during the past decade. In particular, numerical techniques are now widely used, whereas not so long ago such techniques were widely dismissed as impractical and/or incompatible with the basic nature of artificial intelligence.[1]

The present introductory overview, like this special issue as a whole, aims to give researchers in the areas of user and student modeling a picture of the prospects and difficulties associated with these numerical approaches and to show them where further ideas and information can be found. Sections 2, 3, and 4 consider in turn the three major uncertainty management paradigms and the user and student modeling systems that have employed them. Section 2 examines the paradigm that follows traditional probability theory most closely, the one in which Bayesian networks (*BN*s) are the central technique. Sections 3 and 4 then consider the successively less traditional approaches based on the Dempster-Shafer theory of evidence (*DST*) and fuzzy logic (*FL*), respectively; the use of these paradigms is often motivated by objections to the more traditional ones. Section 5 compares the three paradigms with respect to aspects of their usability in realistic contexts of research and application.

Although almost half of the systems covered by this overview fall into the category of student modeling, for convenience the term *user modeling* will be employed in a broad sense that subsumes student modeling. Similarly, the symbol $\mathcal{U}$ will denote a user of a system that does user or student modeling; $\mathcal{S}$ will denote the system in question. So that gender bias can be avoided, masculine pronouns will be used for references to the user in the context of instructional systems, and feminine pronouns will be used in all other cases.

2. Systems That Have Used Bayesian Networks

The most straightforward examples of Bayesian networks are those that involve a physical system that consists of several components some of which influence others causally. For example, a burglar alarm (see, e.g., Pearl, 1988, chap. 2) can be set off by at least two possible causes: a burglary or an earthquake. Or to take a largely physical example that lies a bit closer to user modeling: The probability that a high-jumper will be able to clear a given height can be seen as depending on two main factors: (a) the height of the bar and (b) the high-jumper's ability (expressed, e.g., in terms of the height she can jump successfully 50% of the time). Even if these two factors are known precisely, an observer will be uncertain about the outcome of a jump near the jumper's typical height, because of the operation of various other factors, some of which cannot be taken into account systematically. A specification of the relevant causal relationships makes it possible (a) to *predict* outcomes that depend on particular causes and (b) to *interpret* observed outcomes as evidence concerning the variables that caused them. For example, an unsuccessful jump at an apparently low height suggests a low level of high-jumping ability.

Relationships like these can often be represented naturally with a BN: a directed, acyclical graph in which the nodes correspond to (possibly multivalued) variables and the links correspond to probabilistic influence relationships.[2] As will be seen below, the relationships among variables do not have to be causal in nature, though the operation of a BN tends to be especially easy to understand when this is the case.

---

[1] Books that cover a variety of approaches include those of Kruse et al. (1991), Neapolitan (1990), Pearl (1988), and Shafer and Pearl (1990). The proceedings of the annual conferences on Uncertainty in Artificial Intelligence (e.g., the volumes edited by Lopez de Mantaras & Poole, 1994, and by Besnard & Hanks, 1995) present a broad range of research contributions. Works of more limited scope will be cited in later sections.

[2] The concepts used here are introduced in more detail by Mislevy and Gitomer (1995) in this issue. A longer tutorial exposition of BNs is provided by Charniak (1991) (see also the article by Henrion et al., 1991, in the same magazine issue). A more technical introduction is offered by Russell and Norvig (1995, chap. 15). Detailed technical background is given by Pearl (1988) and by Neapolitan (1990).

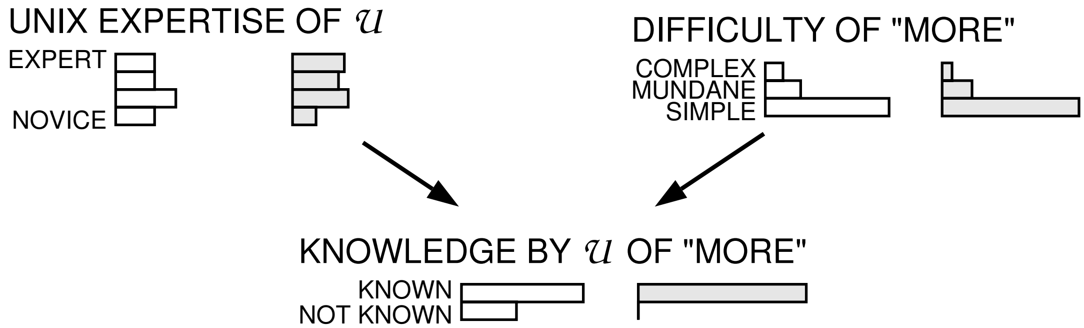

Figure 1. Prediction and interpretation of the user's knowledge of a UNIX concept with a small BN. Each arrow points from a parent node to its child node. The second histogram for each node represents the system's belief after the observation that $\mathcal{U}$ knows the concept "MORE".

2.1. THE IPSOMETER: SMALL-SCALE NETWORKS FOR KNOWLEDGE ASSESSMENT

The first system in this section will serve as a simple illustration of some of the basic concepts of BNs in a user modeling context. The example problem considered here will also serve as an introductory example of the use of DST (Section 3.1) and FL (Section 4.1). The example is situated in the domain of the system KNOME (Chin, 1989; see Section 4.1): An intelligent help system must incrementally update its assessment of a user $\mathcal{U}$'s expertise with respect to the operating system UNIX on the basis of information that reveals whether $\mathcal{U}$ is familiar with three UNIX concepts.

The example BN in Figure 1 represents an adaptation of the inference mechanism of the IPSOMETER (Jameson, 1990, 1992), which was initially developed as a standard of comparison for everyday human judgments of what other persons know. Consider for the moment only the first (white) histogram shown for each of the three *nodes* in the network. Each node represents the system's belief about one of three *variables*. The histogram for UNIX EXPERTISE OF $\mathcal{U}$ shows that four possible levels of this variable are distinguished. At any moment, the system's uncertain *belief* about $\mathcal{U}$'s expertise is represented by a probability distribution, which is depicted by the histogram. In the present example, $\mathcal{U}$ is at first entirely unknown to $\mathcal{S}$, so $\mathcal{S}$'s belief reflects simply the distribution of expertise levels within the population of users that $\mathcal{S}$ deals with.

Similarly, three possible levels for DIFFICULTY OF "MORE" are defined. It can be seen that $\mathcal{S}$ is not entirely certain about how difficult the concept "MORE" is. In spite of all this uncertainty, $\mathcal{S}$ can derive a belief about how likely $\mathcal{U}$ is to know "MORE" by allowing *downward propagation* to occur in the network. The resulting belief is shown in the node KNOWLEDGE BY $\mathcal{U}$ OF "MORE", which is the *child* of the two other *parent* nodes. To derive this belief, the network requires a *conditional probability table*. This table specifies, for each of the 24 combinations of possible values of the variables in the parent nodes and the child node, how likely the value of the child variable is, given the values of the parent variables. These conditional probabilities could be derived from empirical data, estimated by a domain expert, and/or based on a more general theory about the relationships among variables of these types. In the present example, the third possibility is realized: The probabilities are chosen to be consistent with a commonly used model within psychological test theory, the one-parameter logistic model of Item Response Theory (see, e.g., Hambleton & Swaminathan, 1985). This model treats a confrontation between a person and a knowledge item as analogous to an attempt by a high-jumper to

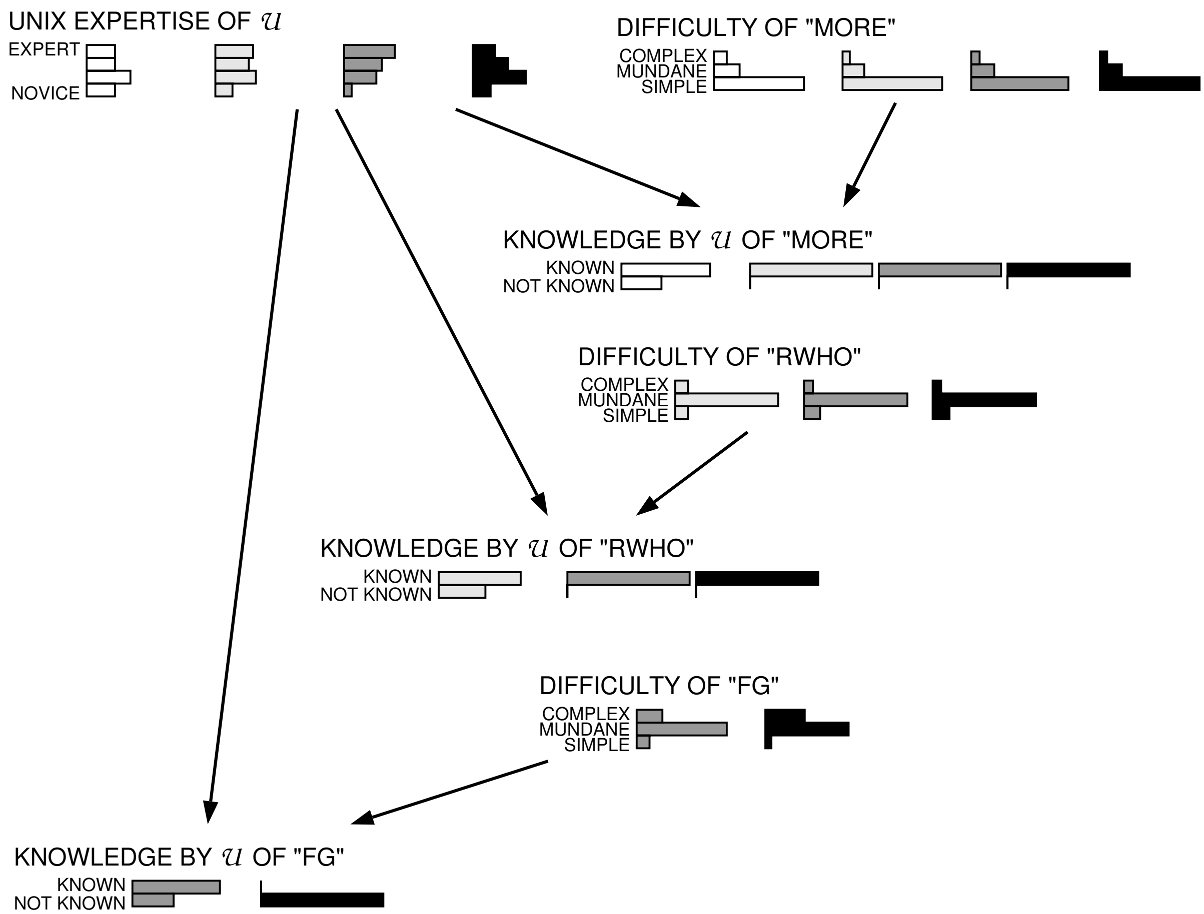

Figure 2. States of the BN of Figure 1 after a user's (lack of) knowledge of two further concepts has been observed. The histograms with a particular shade of gray represent the system's beliefs at a particular point in time.

jump a particular height, where the height of the bar corresponds to the difficulty of the knowledge item.

The belief for KNOWLEDGE BY $\mathcal{U}$ OF "MORE" shows that $\mathcal{S}$ expects $\mathcal{U}$ to know "MORE", even though $\mathcal{S}$ has no prior experience with $\mathcal{U}$, simply because "MORE" is apparently so easy. This type of inference from causes to effects can be called *predictive inference* (cf. Pearl, 1988, p. 6).

Diagnostic inference, on the other hand, moves from observed effects to possible causes. For example, the second (gray) histogram for each node shows $\mathcal{S}$'s beliefs after $\mathcal{S}$ has learned that $\mathcal{U}$ does in fact know "MORE". $\mathcal{S}$ now assigns a probability of 1.00 to the value KNOWN for KNOWLEDGE BY $\mathcal{U}$ OF "MORE". $\mathcal{S}$ then updates its belief about UNIX EXPERTISE OF $\mathcal{U}$ via *upward propagation*, which can be seen as a generalization of the application of Bayes' rule. The basic idea is that those combinations of values of the parent variables that are associated with the observed result by a high conditional probability become proportionally more likely, whereas the other combinations become less likely. In this case, $\mathcal{S}$ now assigns slightly higher probabilities to the higher levels of UNIX EXPERTISE OF $\mathcal{U}$; but on the whole $\mathcal{S}$ is not very "impressed" by $\mathcal{U}$'s familiarity with this simple concept.

Similar processing occurs when $\mathcal{S}$ learns about $\mathcal{U}$'s knowledge of further concepts, as illustrated in Figure 2, which extends Figure 1. For each new concept, a node representing $\mathcal{S}$'s belief about its difficulty is added to the network. A node for the belief about $\mathcal{U}$'s knowledge of the concept is also added and then updated to reflect $\mathcal{U}$'s observed (lack of) knowledge. After $\mathcal{U}$ is seen to know the MUNDANE concept "RWHO", $\mathcal{S}$'s assessment of her expertise again becomes more positive. Later, when she is seen not to know "FG", another MUNDANE concept, there is a change in the opposite direction. $\mathcal{S}$ begins to converge on the conclusion that $\mathcal{U}$'s level of expertise is neither very high nor very low.

The reader may have noticed that $\mathcal{S}$ also updates its beliefs about the difficulty of the various concepts on the basis of $\mathcal{U}$'s knowledge of them (compare, for example, the two histograms for DIFFICULTY OF "FG"). $\mathcal{S}$ even continues to update its belief about DIFFICULTY OF "MORE" after its initial processing of the fact that $\mathcal{U}$ knows the concept "MORE", because $\mathcal{S}$'s revised assessments of $\mathcal{U}$'s expertise cause $\mathcal{S}$ to rethink its original interpretation of this evidence. This updating of beliefs about variables other than those describing the current user is an automatic consequence of the standard BN propagation techniques; its consequences will be discussed in Section 2.12.

2.2. Categories of Systems That Have Used Bayesian Networks

Figure 3 and the later Figure 5 together give an overview of systems that have used BNs for user modeling.[3] Figures 8 and 11 will give similar overviews of systems that have used DST and FL, respectively. These figures characterize each system in terms of the types of variable about which it makes inferences. The five categories of variables distinguished are explained in Appendix A.

The systems will be discussed in the order of their appearance in the figures. The ones that have used BNs fall into four categories:

1. The main emphasis of the first four systems in Figure 3 is on the assessment of more or less general abilities. These systems mainly use the diagnostic inference capabilities of BNs, as was illustrated by the way in which the IPSOMETER (Section 2.1) interpreted evidence about $\mathcal{U}$'s knowledge of concepts.
2. The fifth and sixth systems in Figure 3 assess not general abilities but the possession of knowledge of individual concepts. The distinction between predictive and diagnostic inference is less clear here, because both upward and downward propagation can be used to infer that $\mathcal{U}$ knows a concept, given the fact that she knows certain other concepts.
3. The first three systems in Figure 5 are designed to recognize the plans of an agent (who is not actually a computer user, in the context of these systems). The emphasis is again mainly on diagnostic inference, although one purpose of plan recognition is to make possible the prediction of a person's future actions.
4. For the last three systems in Figure 5, predictive inference is at least as important as diagnosis. Predictions concerning a user's cognition and/or behavior form a basis for the system's decisions and actions.

Note that the arrows in Figures 3 and 5 have a different meaning than those used in graphical representations of BNs: In particular, an upward arrow in one of these figures may correspond to

[3] Three other uses of BNs have been described only briefly in the publications that have appeared to date. They are listed here for readers who might be especially interested in the domains involved: Sime (1993) uses a BN to express relationships among several types of knowledge that a student may have about a physical system. The BNs of PITAGORA 2.0 (Carbonaro et al., 1995, section 3.1; Roccetti & Salomoni, 1995) make inferences about a student's knowledge of problem solving procedures for Euclidean geometry. Draney et al. (1995) describe BNs that are designed for use within a LISP programming tutor. Their purpose is to assess students' mastery of items of procedural knowledge on the basis of their performance on programming tasks for which these knowledge items are required.

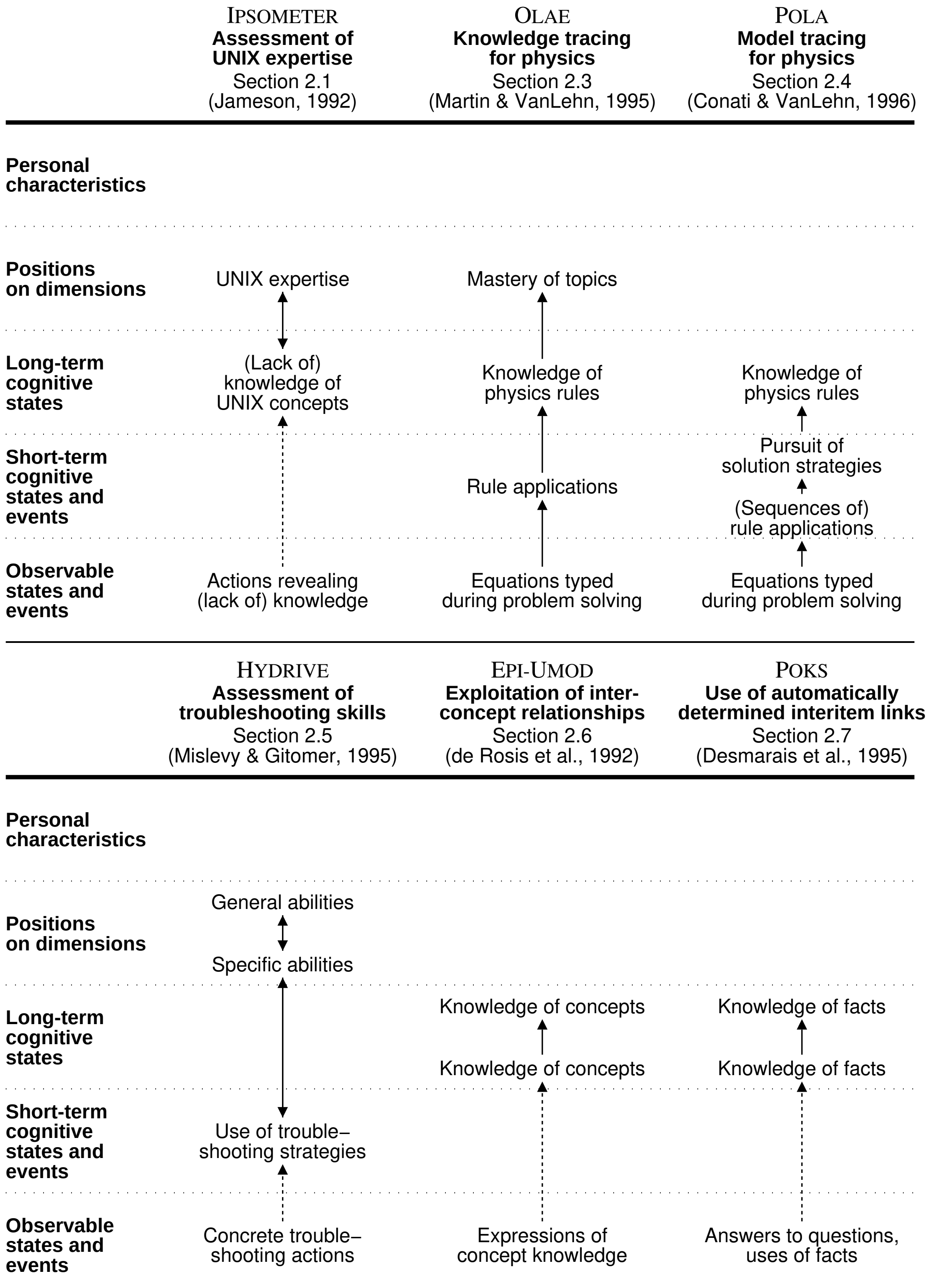

Figure 3. Systems that have used BNs for knowledge assessment.

The five levels of variables are explained in Appendix A. "$X \rightarrow Y$" means "The system makes inferences about variables of type Y on the basis of its beliefs about variables of type X". A dashed arrow means that the inferences are not made with the system's numerical uncertainty management techniques.

inferences that are made via upward propagation, which in a BN actually moves backward along links that point "downward", from parent to child.

2.3. OLAE: ASSESSING PHYSICS SKILL WITH FINE OR COARSE GRANULARITY

OLAE (Martin & VanLehn, 1993, 1995) is part of a physics tutoring system. But OLAE is not designed to support interactive tutoring. Its purpose is to yield a differentiated and reliable assessment of a student's knowledge of a subdomain of physics. The observations that its BNs use as input concern equations typed in by $\mathcal{U}$ while $\mathcal{U}$ solves a physics problem. The most important inferences concern $\mathcal{U}$'s (lack of) knowledge of particular physics rules (e.g., how to compute the mass of an object given its density and volume). Each such rule corresponds to a two-valued variable represented by a node in a BN. The system automatically constructs the BN that corresponds to a particular problem on the basis of a *problem solution graph*; this graph essentially represents the various ways in which $\mathcal{U}$ might try to solve the problem by applying rules to the facts of the problem and to the intermediate results generated by rule applications. The problem solution graph is itself automatically constructed on the basis of a description of the problem and a cognitive model of the problem-solving process.

The BN constructed for a problem therefore reflects, among other things, the likelihood that $\mathcal{U}$ would type in particular equations if he possessed particular rules. The observed behavior of $\mathcal{U}$ gives rise to upward propagation, which changes $\mathcal{S}$'s beliefs about $\mathcal{U}$'s possession of individual rules. These rule probabilities can be inspected by an assessor such as a teacher.

A unique feature of OLAE is its provision of a second type of BN specifically for the assessor, who consults the system after the processing sketched above has been completed. The assessor's network contains (a) the rule-possession nodes from the original BN which represent the result of that BN's processing and (b) dimensional nodes for more abstract variables that represent $\mathcal{U}$'s mastery of particular topics such as *Kinematics* or *Content of Chapter 5*. (The idea is that the assessor may be interested in a coarse-grained characterization of $\mathcal{U}$'s knowledge as well as in a fine-grained characterization in terms of individual rules.) Note that the dimensional nodes could in principle also be integrated into the original BN. If this were done, the dimensional nodes would be updated automatically during the interpretation of $\mathcal{U}$'s problem-solving behavior. In addition, they could in turn influence other aspects of that interpretation process, depending on exactly how they were defined (cf. Section 2.5 below). For example, if some aspects of $\mathcal{U}$'s behavior suggested a good overall mastery of kinematics, the probabilities associated with all kinematics rules would increase. $\mathcal{S}$ would then be less inclined to explain other aspects of $\mathcal{U}$'s behavior in terms of $\mathcal{U}$'s lack of knowledge of particular kinematics rules.

2.4. POLA: FROM KNOWLEDGE TRACING TO MODEL TRACING

Recently, Conati and VanLehn (1996) have presented the system POLA, which builds on the techniques of OLAE. Recall that OLAE is invoked only after the student has completed work on at least one physics problem; this kind of retrospective diagnosis is called *knowledge tracing*. By contrast, POLA is designed to perform *model tracing*: It can be invoked repeatedly during the student's problem solving, every time the student has performed an observable action. A primary task of POLA is to determine which of the various possible solution paths the student is pursuing and what rules he has applied so far. One problem that arises here is the need to distinguish between:

- rule applications that the student has already performed; and
- rule applications that belong to the student's chosen solution path but that the student has not yet performed.

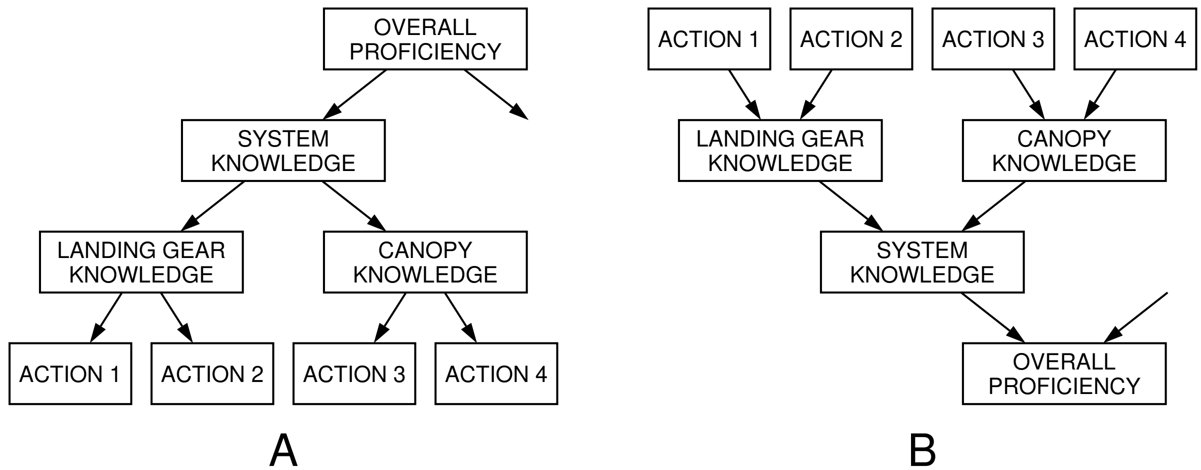

Figure 4. Two possible relationships between general and specific abilities.

It would in principle be possible to take this distinction into account with a more complex representation of hypotheses about rule applications than the straightforward representation used in OLAE's BNs. But Conati and VanLehn adopt an alternative strategy which allows the semantics of the network nodes to remain relatively simple: POLA constructs its BNs incrementally, in general adding nodes each time the student performs an observable action. At any moment, the nodes in a BN concern only rule applications for which there is evidence that the student has already performed them. In this way, possible future rule applications do not have to be taken into account in the BNs.

POLA's approach illustrates that the designer of a system that uses BNs sometimes has the choice between two ways of dealing with a necessary distinction:

- Represent the distinction explicitly in the semantics of the network nodes and in the conditional probability tables; or
- design a procedure for dynamic construction of the BN which deals with the distinction in a more procedural fashion.

As is usual with decisions about declarative vs. procedural solutions, there are arguments for and against each of these options. But POLA's domain does appear to be one in which a partly procedural approach deserves serious consideration.

2.5. HYDRIVE: MODELING A HIERARCHY OF ABILITIES

HYDRIVE (Mislevy & Gitomer, 1995, in this issue) models a student's competence at troubleshooting an aircraft hydraulics system. One salient characteristic of the system is its use of dimensional variables at three different levels of specificity, as depicted in the three upper levels of the partial network shown in Figure 4A. These variables illustrate the problem that it is not always clear what direction the links in a BN should go in, especially when the relationships are not causal relationships in an obvious sense. The two main options for this case are illustrated (with some simplification) in Figure 4. With the first option (Figure 4A), SYSTEM KNOWLEDGE is viewed as being in some sense a determinant of the more specific abilities LANDING GEAR KNOWLEDGE and CANOPY KNOWLEDGE. With the second option, SYSTEM KNOWLEDGE is, roughly speaking, the sum of the more specific abilities.

HYDRIVE adopts the first of these two options, whereas OLAE (Section 2.3) basically adopts the second one. Which option is better depends on exactly what the designer intends the dimensional variables to represent. The question can be brought into sharper focus if the consequences of each of the two options for the BN's processing are examined. Consider, for example, the case where $\mathcal{U}$ performs a successful action that leads $\mathcal{S}$ to modify upward its belief about $\mathcal{U}$'s LANDING GEAR KNOWLEDGE. What should the consequences for $\mathcal{S}$'s beliefs about SYSTEM KNOWLEDGE and CANOPY KNOWLEDGE be?

With the first solution (Figure 4A), $\mathcal{S}$ will now ascribe to $\mathcal{U}$ greater SYSTEM KNOWLEDGE (by upward propagation) and then in turn expect stronger CANOPY KNOWLEDGE (by downward propagation).

With the second solution (Figure 4B), $\mathcal{S}$ will likewise ascribe to $\mathcal{U}$ greater SYSTEM KNOWLEDGE (this time by downward propagation). But $\mathcal{S}$ will now have no reason to be more confident that $\mathcal{U}$ is strong in CANOPY KNOWLEDGE, as this is simply an independent factor that also influences SYSTEM KNOWLEDGE. In fact, if $\mathcal{S}$ had previously received some direct evidence to the effect that $\mathcal{U}$ had a given level of SYSTEM KNOWLEDGE, the increase in $\mathcal{S}$'s assessment of $\mathcal{U}$'s LANDING GEAR KNOWLEDGE would lead $\mathcal{S}$ to *lower* its assessment of $\mathcal{U}$'s CANOPY KNOWLEDGE, by upward propagation.

Another question concerning the relatively specific dimensional nodes used in HYDRIVE is whether they are optional, to be included in the BN only if the designer wishes to obtain an especially differentiated assessment of the user's abilities. One might at first think that nodes such as LANDING GEAR KNOWLEDGE could be omitted for simplicity and that nodes such as ACTION 1 could be linked directly to a global dimensional node like OVERALL PROFICIENCY. But this simplification could lead to serious distortions: For example, the successful performance of the pair of actions ACTION 1 and ACTION 2 would lead to basically the same inferences about $\mathcal{U}$'s OVERALL PROFICIENCY as the successful performance of the pair consisting of ACTION 1 and ACTION 4. But in reality, the former pair of actions constitutes weaker evidence of good OVERALL PROFICIENCY: It is possible that $\mathcal{U}$ simply happens to know a lot about landing gears. (This point becomes clearer in the case where $\mathcal{U}$ performs 10 successful actions, all of which require only landing gear knowledge.) The key consideration is that the outcomes of different landing-gear-related actions are not *conditionally independent*, even given a particular level of OVERALL PROFICIENCY—although they presumably are, to a sufficient degree, given a particular level of LANDING GEAR KNOWLEDGE. As explained by Mislevy and Gitomer (1995), the links in a BN have to be defined in such a way that all dependencies among variables are reflected.

2.6. EPI-UMOD: EXPLOITING DEPENDENCIES AMONG KNOWLEDGE ITEMS

Some approaches to knowledge assessment go one step further than introducing dimensional variables that are relatively specific: They attempt to avoid dealing with dimensional variables altogether.

The system EPI-UMOD (de Rosis et al., 1992) models the knowledge that various categories of medical personnel possess concerning concepts used in the analysis of epidemiological data. The BNs constructed contain no dimensional nodes like MASTERY OF EPIDEMIOLOGICAL DATA ANALYSIS. Instead, for each of a number of concrete user categories (e.g., “hospital doctor”), a separate BN is constructed which represents only specific probabilistic links among individual knowledge items (e.g., “How likely is it that $\mathcal{U}$ knows the concept “RELATIVE RISK” if she knows/does not know the concept “RISK FACTOR”?”). Given input information about some concepts that $\mathcal{U}$ does or does not know, the BN uses both downward and upward propagation to update its predictions about $\mathcal{U}$'s knowledge of other concepts.

A strong point of this approach is that it captures well the following type of close relationship between concepts: $\mathcal{U}$ can hardly know the concept “RELATIVE RISK” unless he has learned the concept “RISK FACTOR”. This relationship would be captured only partially by a belief on the part of $\mathcal{S}$ that “RISK FACTOR” was simply *easier* than “RELATIVE RISK” on some general difficulty dimension. Techniques of knowledge assessment that emphasize the exploitation of relationships among specific knowledge items have been explored extensively within the theory of *knowledge spaces*,[4] which promises to play an increasingly important role in user modeling.

On the other hand, there are many probabilistic relationships among knowledge items which are not described equally well in terms of prerequisite relationships. For example, a user who knows how to *file away* an e-mail message is relatively likely to know how to *forward* a message, yet neither of these two pieces of knowledge is a prerequisite for the other one; they are simply both reflections of e-mail handling proficiency. It is questionable whether the capability of handling such relationships in a straightforward fashion in a BN should be sacrificed in an effort to capture another type of relationship.

On the more practical side, the method of de Rosis et al. requires that a large number of conditional probabilities be specified for the links between the individual concepts. In fact the authors find it necessary to do this for each of several categories of user, in order to take into account differences in the ways in which they go about learning the concepts in question. This practicality problem is one of the main issues addressed by Desmarais et al. (1995) in this issue.

2.7. POKS: KNOWLEDGE ASSESSMENT WITHOUT KNOWLEDGE ENGINEERING

Desmarais et al. (1995) show how a network of the same general type as those of de Rosis et al. (1992) can be constructed fully automatically on the basis of a modest amount of empirical data. They then report on an investigation of the utility of such a network for the assessment of the knowledge of individual users.

The authors' goal is to determine how far one can get in practice with methods that make minimal demands in terms of empirical data and computation. Accordingly, they use simpler methods than the techniques that have been developed for BNs. Their work nonetheless deals with basically the same issues that would arise if BNs were used for the same purpose. It is likely to be followed in the future by applications of techniques that are currently being developed for the learning of BNs (Desmarais et al., 1995, section 4; see also, e.g., Heckerman et al., 1994; Russell & Norvig, 1995, section 19.6; Russell et al., 1995). These techniques allow the conditional probabilities of a BN—and in some cases its structure—to be acquired automatically from empirical data. This possibility may ultimately have important consequences for the practical usability of BNs for user modeling.

Another important general issue investigated by the authors (section 6.3.2) is that of biased sampling of user behavior. This problem can be explained with reference to the introductory example in Section 2.1 (Figure 2), in which the user $\mathcal{U}$ was observed to know the concepts "MORE" and "RWHO" but not the concept "FG". Suppose that the system $\mathcal{S}$ acquired information about $\mathcal{U}$ in such a way that it only learned about cases where $\mathcal{U}$ did in fact know a concept (e.g., because the only information available consisted of $\mathcal{U}$'s active use of a concept). Then negative observations like the third one in Figure 2 would never be obtained, and $\mathcal{S}$'s belief about $\mathcal{U}$'s UNIX expertise would keep becoming more optimistic. (The opposite effect could arise, for example, if the input data consisted only of questions by $\mathcal{U}$ to the help system which indicated that $\mathcal{U}$ did *not* know a given concept.) In other words, even the most valid BN can yield grossly incorrect conclusions if measures are not taken to ensure that acquisition of evidence about each variable is unbiased—that is, that the probability that information about the value of a variable V will be obtained is independent of the value that V actually has. More generally, designers of user modeling systems may take a hint from the fact that social psychologists (e.g., Nisbett & Ross, 1980, chap. 4) have identified biased data sampling as a major source of error in everyday social perception.

[4] See, for example, Falmagne et al. (1990), Villano (1992), and Kambouri et al. (1994), as well as the discussion by Desmarais et al. (1995, section 4) in this issue.

Desmarais et al. discuss two ways of avoiding data sampling bias:

1. $\mathcal{S}$ obtains through indirect inference the information that it cannot obtain directly because of bias.
2. $\mathcal{S}$ in effect administers a test to $\mathcal{U}$, deciding itself which variables to obtain information about.

A further possibility is for $\mathcal{S}$ simply to ignore particular types of information. For example, if biased information consisting of questions to the help system is obtained along with other, unbiased information, $\mathcal{S}$ can refrain from using the biased information to update its BN.

2.8. Using the Environment as Evidence for Plan Recognition

The next three systems (characterized in the top half of Figure 5) employ BNs in a quite different way than the ones discussed so far: for recognizing an agent's plans.

A good starting point is the system presented by Pynadath and Wellman (1995). The authors describe how the system handles a relatively simple example of a plan recognition problem: Suppose you are driving in the middle lane of a three-lane highway and you see the car directly behind you move into the right lane. You may wonder whether, during the next few moments, the driver $\mathcal{U}$ of the car will (a) move past you and then return to the middle lane, (b) stay behind you in the right lane, or (c) leave the highway via an exit. To make this prediction, you are likely to try to interpret the car's initial motion in terms of $\mathcal{U}$'s goals and plans—for example, the goal of attaining a higher velocity, which may require the execution of the plan of passing the car ahead. Pynadath and Wellman's method is intended ultimately to enable a computer-driven vehicle to perform this sort of plan recognition.[5]

The authors' BNs are based on a general causal model of the way in which an agent constructs and executes plans in a given environment. The causal planning model can be summarized in terms of the following phases, where the events in each phase are seen as causing those in the later phases:

1. $\mathcal{U}$ observes aspects of the physical environment.
2. $\mathcal{U}$ compares these observations with her goals, determining the extent to which her goals are fulfilled.
3. $\mathcal{U}$ selects a general plan to address an unfulfilled goal.
4. $\mathcal{U}$ refines the plan, taking into account details of the current situation.
5. $\mathcal{U}$ executes a sequence of actions, perhaps including communication actions which signal her intentions.
6. These actions have observable effects on the environment.

The BNs constructed within this framework have nodes that correspond to events within each phase.[6] In particular, the variables for the two planning phases have as possible values the various alternative plans that can be adopted with a view to achieving a given subgoal. The links emanating from the nodes for each phase mostly point to nodes for the next phase.

Using the network for plan recognition is quite straightforward: $\mathcal{S}$ first fixes the values of any observed variables—mainly variables for Phases 1 and 6. After upward and downward propagation have occurred, $\mathcal{S}$ can examine the beliefs in the network concerning any other variables that $\mathcal{S}$ is interested in. The most obvious variables of interest concern $\mathcal{U}$'s plans (Phases 3 and 4); but $\mathcal{S}$ can also simply derive a prediction of $\mathcal{U}$'s future behavior by examining nodes for Phases 5 and 6. In other words, $\mathcal{S}$ may want to know what $\mathcal{U}$ is going to do next without worrying much about what her reasons

[5] Forbes et al. (1995) use BNs for a similar purpose, but their system does not at present explicitly represent hypotheses about individual drivers.

[6] The nodes for Phase 1 represent the actual state of the environment, not $\mathcal{U}$'s observations of the environment. It is assumed in Pynadath and Wellman's (1995) example that $\mathcal{U}$ is able to observe the environment accurately. The authors note how this assumption could be avoided.

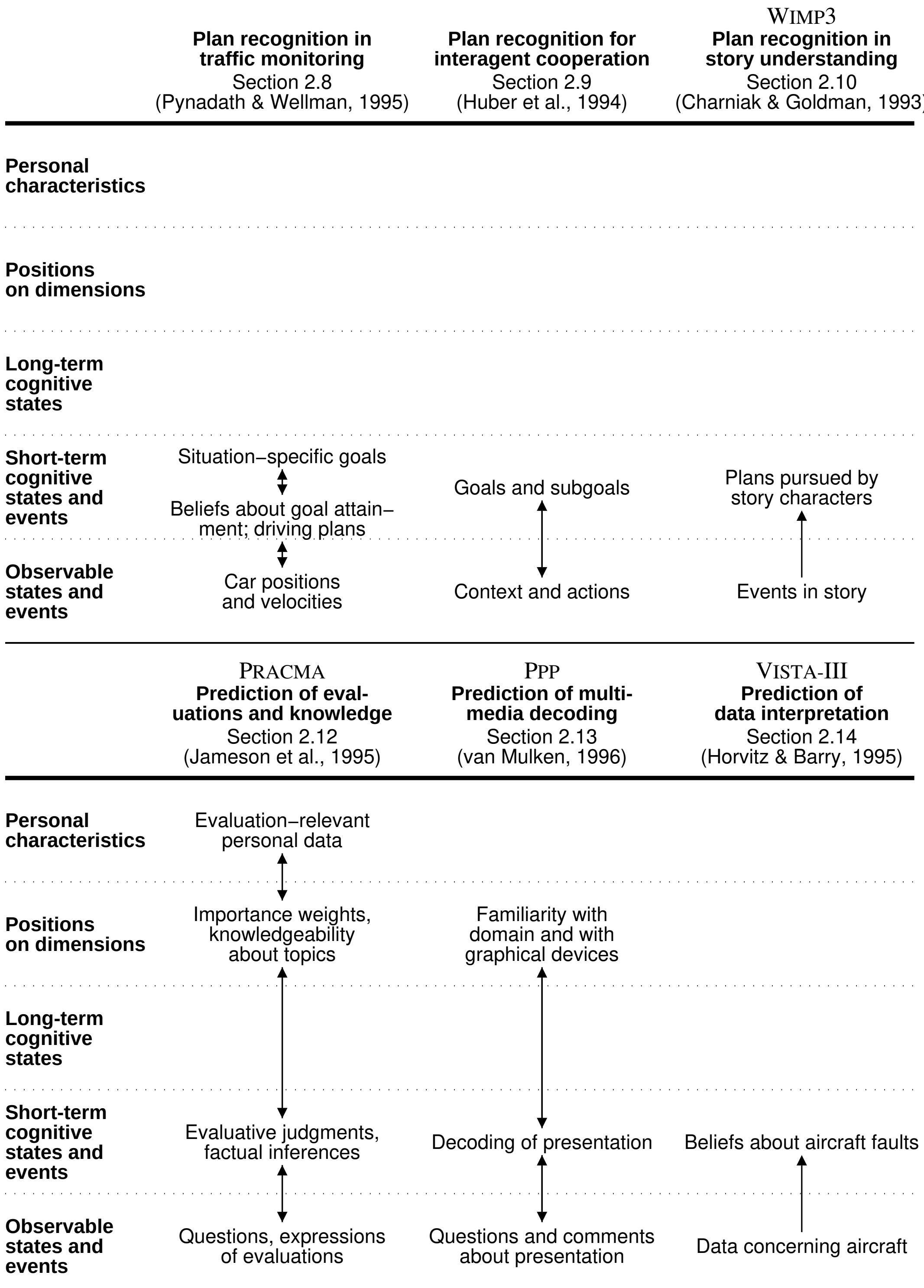

Figure 5. Systems that have used BNs for plan recognition (top half) or for prediction of users' responses (bottom half) (cf. Figure 3 and Appendix A).

for doing it might be—an attitude that makes sense if a given action can be performed as part of the execution of various different possible plans. This flexible use of the network's inferences is possible because of the integration of diagnostic and predictive inference in BNs.

Perhaps the most innovative aspect of this model is its inclusion of a number of variables that represent aspects of the physical context (Phase 1). For most user modeling systems that use BNs, the observable data concern only the user's behavior. These systems do not represent within a BN the dependence of $\mathcal{U}$'s cognition and behavior on $\mathcal{U}$'s perception of the current situation. The reason is presumably that the relevant aspects of the current situation are in general known to the system. For example, when a tutoring system $\mathcal{S}$ has presented a problem for $\mathcal{U}$ to solve, $\mathcal{S}$ in general knows exactly what problem has been presented and what other information $\mathcal{U}$ has available. $\mathcal{S}$ must take these facts into account when constructing a BN to analyze $\mathcal{U}$'s behavior; but these facts do not have to be represented in nodes of the BN, since there is no uncertainty about them.

The broader conceptualization presented by Pynadath and Wellman seems likely to become more important as computers move off of the desktop and out into the world. A user modeling system will then often have only uncertain hypotheses about the user's situation. But even designers of more conventional systems might consider including representations of relevant contextual events in their BNs. For example, a physics tutoring system could be uncertain as to whether the student has available a list of relevant formulas, either on paper or somewhere on the screen. And even information presented by the system itself may be partly unknown to $\mathcal{S}$. For example, if $\mathcal{S}$ presents video clips, $\mathcal{S}$ may not have a representation of all of the events shown in the video clip that might influence $\mathcal{U}$'s behavior.

2.9. Reusing Plan Generation Knowledge for Plan Recognition

One question raised by the system of Pynadath and Wellman (1995) is: Where does the network come from? In principle, a designer could design by hand a network like this for every plan recognition problem that the system might face. But this procedure may require unfeasible and unnecessary effort, especially if the system already has access to a plan library which can be used for the *generation* of the plans in question. In this case, it should be possible to use this library as a basis for the automatic generation of BNs that can be used for the *recognition* of the same plans.

This approach is introduced by Huber et al. (1994). Their example domain involves two agents cooperating to perform a military reconnaissance task. The authors present a general procedure for mapping planning knowledge onto BNs. For example, given a top-level goal that an agent might pursue, the procedure creates a BN with nodes corresponding to the following types of variables:

- top-level goals and subgoals;
- actions than can be performed to achieve subgoals;
- observable events and states that reflect the fact that a given action is being performed; and
- aspects of the context.

With this method, a user modeling system could generate a library of BNs before obtaining any observations of the current user $\mathcal{U}$. Each of these BNs would be specialized in the recognition of ways in which the user might pursue a particular high-level goal. This top-down approach is in some respects opposed to that of Charniak and Goldman (1993), which will be examined in the next subsection, in which plan recognition BNs are generated bottom-up on the basis of information about an agent's actions.

An incidental contribution of Huber et al.'s article is an example of a solution to a problem that frequently arises in user modeling: the problem of how best to handle alternative hypotheses which are (more or less) mutually exclusive and which have qualitatively different consequences. Consider, for example, the small plan-recognition BN depicted in Figure 6A (ignoring for the moment the dotted

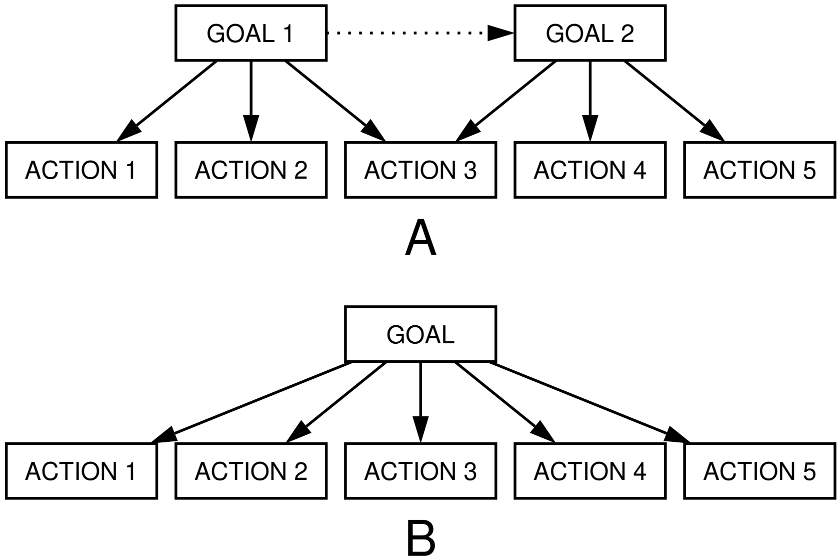

Figure 6. Two ways of representing hypotheses about mutually exclusive goals.

arrow from GOAL 1 to GOAL 2): GOAL 1 can be achieved through a combination of the actions ACTION 1, ACTION 2 and ACTION 3. Therefore, the left-hand half of the network can be used to recognize whether a user $\mathcal{U}$ is pursuing GOAL 1. A similar relationship holds for GOAL 2. But what if we know that $\mathcal{U}$ cannot pursue both of these goals at the same time? Then any evidence that $\mathcal{U}$ is pursuing GOAL 1 (e.g., an observation that $\mathcal{U}$ has performed ACTION 1) should count as evidence against the hypothesis that $\mathcal{U}$ is pursuing GOAL 2; yet this sort of inference is not provided for by the solid arrows in the network.

A straightforward way of representing the mutual exclusivity of a set of hypotheses in a BN is to include them all as possible values of a single variable (cf., e.g., the four possible levels of UNIX EXPERTISE OF $\mathcal{U}$ in Figure 1). This generally works well as long as the several hypotheses all have implications for the same set of child variables. But in the present example, the result would be the network shown in Figure 6B, in which the possible values for GOAL are GOAL 1, GOAL 2 and NEITHER GOAL 1 NOR GOAL 2. This solution has an obvious disadvantage: Inspection of the graph does not show which goal is associated with which actions; this information is hidden inside the conditional probability table. The problem becomes worse if there are more alternative goals, or if each one has a more complex set of associated actions and/or subgoals (cf., e.g., the example given by Mislevy, 1994, pp. 467–468).

The simpler solution used by Huber et al. is to add an *inhibitory link* from GOAL 1 to GOAL 2 (represented by the dashed arrow in Figure 6A). The conditional probability table for this link implies that if GOAL 1 is being pursued, GOAL 2 is not being pursued; by upward propagation, evidence in favor of GOAL 2 will likewise count against GOAL 1. The solution can be generalized to cases where GOAL 1 and GOAL 2 are not strictly mutually exclusive but rather GOAL 2 is simply less likely given GOAL 1 than it is without it. A somewhat unsatisfactory aspect of this solution is that the direction of the inhibitory link is in general largely arbitrary: You can say that pursuing GOAL 1 makes it impossible to pursue GOAL 2, but the opposite statement is equally justifiable. Yet only one of the two links can be included, because a BN cannot include cycles.

2.10. WIMP3: PUSHING BAYESIAN NETWORKS TO THE LIMIT FOR STORY UNDERSTANDING

Charniak and Goldman (1993) and (more briefly) Charniak and Goldman (1991) present an impassioned argument for the use of BNs for plan recognition; they argue, in particular, for the superiority of their approach to the DST-based approach of Carberry (1990; cf. Section 3.2 below). The input to their system WIMP3 does not come from a computer user but rather from passages of text that tell brief stories, such as “Jack went to the liquor store. He pointed a gun at the owner”.

The authors give an unusually explicit characterization of the properties of the nodes in their BNs and of the procedure by which WIMP3 dynamically and incrementally constructs such a network while processing a story. Of particular interest is a marker-passing technique that is used to prevent the explosive construction of large networks which contain mainly nodes associated with very low probabilities; similar problems can easily arise in other systems that generate BNs dynamically.

For user modeling researchers, Charniak and Goldman's BNs have a drawback, compared to those of Pynadath and Wellman (1995) and Huber et al. (1994): The nodes do not correspond to intuitively natural concepts like goals and actions. Instead, they correspond to fine-grained propositions like LS2 IS A LIQUOR STORE and LS2 IS THE FILLER OF THE “STORE” SLOT IN THE LIQUOR-SHOPPING EVENT LSS3. It can be difficult to assign probabilities to propositions like these in a natural way. For example, one of the numbers in the conditional probability table associated with the second proposition is $1/|\text{liquor-store}|$, where the denominator refers to the total number of liquor stores that exist. This fine-grained and partly unintuitive character of the BNs is due to the fact that BNs are used here for more than just plan recognition: They manage the entire process of story understanding, beginning with the interpretation of individual words in the input text.

This work also illustrates how the choice of an appropriate conceptualization can depend on the specific context in which a system is used. The assignment of probabilities in WIMP3 presupposes that the frequency of occurrence of events and objects in the stories corresponds to their frequency in some (real or fictional) world. But some authors like to violate readers' expectations intentionally. With such stories, the fact that a given event or plan seems likely as far as frequencies are concerned can in itself constitute a reason for believing that it will *not* occur in the story. The more general point is this: For the interpretation of intentional communicative acts, a probabilistic model must encompass variables that characterize aspects of the communication situation, such as the motivation of the communicators. The application of Bayesian methods (though not specifically BNs) to variables of this type is illustrated by contributions of Raskutti and Zukerman (1991) and Kipper and Jameson (1994).

2.11. OFF-LINE BAYESIAN ANALYSIS FOR PLAN RECOGNITION

Another approach to plan recognition that makes use of BNs is that of van Beek (1996). Unlike Pynadath and Wellman (1995), Huber et al. (1994), and Charniak and Goldman (1993), he does not show how BNs can be used on-line to perform plan recognition inferences in specific cases. Instead, he offers a theoretical analysis, in terms of Bayesian inference, of several general, nonprobabilistic plan recognition heuristics that have been proposed by previous researchers. (The analysis by van Beek is not represented in Figure 5, because it does not concern a specific system.)

For example, one heuristic is applicable to cases such as ones where the observed actions of a user $\mathcal{U}$ can be explained through either of two assumptions:

1. $\mathcal{U}$ is pursuing GOAL 1; or
2. $\mathcal{U}$ is pursuing the two independent goals GOAL 2 and GOAL 3.

The heuristic specifies that the former explanation is preferable, because it is more parsimonious.

The analysis by van Beek shows, however, that there are cases in which the second explanation is more probable—for example, when GOAL 1 is a goal that people very rarely pursue whereas GOAL 2 and GOAL 3 are both frequently pursued. A conclusion is that this heuristic should only be applied in domains in which these cases are unlikely to arise.

The reader may wonder why this type of Bayesian analysis shouldn't be performed on-line by the system itself, so that the restriction to particular domains can be avoided. The argument of van Beek is that there are situations in which this approach is impractical. For example, the required probabilities may be unavailable (cf. Section 5.1). In such situations, a useful next-best approach is to apply a set of carefully evaluated nonprobabilistic heuristics as a surrogate for on-line probabilistic analysis. Analogous strategies are worth considering in other systems in which the on-line use of BNs (or of other numerical approaches to uncertainty management) is for some reason impractical.

2.12. PRACMA: NEW USES FOR BAYESIAN NETWORKS IN DIALOG

The systems considered so far have made inferences either about users' knowledge or about their execution of plans. The system PRACMA (Jameson et al., 1995; Schäfer, 1994) illustrates that BNs are also applicable when the user's evaluation processes are of central interest. (FL-based systems that likewise reason about the user's evaluations are described in Sections 4.2 and 4.3.) The overall system's basic task is to present information to a user who wants to make an evaluative judgment about a given object (e.g., a used car). It presupposes, as an approximation, that the user will evaluate the car according to the principles of Multi-Attribute Utility Theory (von Winterfeldt & Edwards, 1986). The system has uncertainty about virtually all of the parameters that enter into such an evaluation. $\mathcal{S}$ is also uncertain about $\mathcal{U}$'s knowledgeability and about the prior beliefs (themselves mostly uncertain) that $\mathcal{U}$ has about the object under discussion. All of these types of uncertainty are managed within a dynamically constructed BN.

Jameson et al. (1995) discuss how this type of BN can be used to handle several general tasks faced by systems that provide evaluation-oriented information, including the following:

1. predicting how $\mathcal{U}$ will evaluate a given object;
2. predicting how $\mathcal{U}$ will react to information about particular aspects of an object;
3. interpreting $\mathcal{U}$'s behavior as evidence concerning her evaluation criteria and knowledge;
4. deciding what information to elicit explicitly from $\mathcal{U}$.

The third task requires diagnostic inference. The first two tasks require predictive inference, a type which has played a minor role in the systems mentioned so far but which will also be seen in the two systems described in the following subsections.

The fourth task requires a type of reasoning that has general importance for user modeling systems: The system reasons about the *value* that particular types of information about the user (e.g., about her personal characteristics) would be likely to have for the system; on the basis of this assessment, $\mathcal{S}$ can decide whether it seems worthwhile to take steps to acquire this information. Pearl (1988, section 6.3) gives an overview of various approaches to the assessment of the value of information. In the approach taken by PRACMA, the value is a function of the expected extent to which the *uncertainty* in the system's model of the user would be reduced by the information. Other approaches require a quantification of the *utility* associated with possible consequences of $\mathcal{S}$'s actions.

When interpreting evidence in the user's behavior (the third task listed above), PRACMA not only updates its beliefs about the current user but also learns about properties of users in general. For example, the system must inevitably begin with some assumption about the average importance assigned to the evaluation dimension of “Safety” by used-car customers in general; because otherwise $\mathcal{S}$ could make no predictions about how an unknown customer would evaluate a car's safety features.

But $\mathcal{S}$'s assumption is represented as a long-term node in its BN, so that it can be updated (gradually) on the basis of experience with users' reactions. This method represents a relatively simple way of handling one of the most frequently mentioned difficulties in applying BNs, namely the difficulty of specifying in advance all of the necessary probabilities (cf. Section 5.1). The method's applicability is much more limited than that of the general learning techniques mentioned in Section 2.7, but it may be useful when a system only has to update, during actual use, a small number of key parameters that it needs for its BNs.

Finally, PRACMA illustrates a further possible approach to the problem of how to generate BNs dynamically (cf. Sections 2.3, 2.4, 2.7, 2.9 and 2.10): To predict what inferences $\mathcal{U}$ might make on the basis of a given statement by $\mathcal{S}$, the system first uses a modal-logic-based representation system (Hustadt & Nonnengart, 1993) to generate possible inferences that $\mathcal{U}$ might *conceivably* make. $\mathcal{S}$ then uses the proof trees produced by the modal-logic-based system to construct BNs which predict whether this particular $\mathcal{U}$ actually has enough knowledge to make these inferences (Jameson, 1995). That is, BNs are used here in conjunction with a very different nonprobabilistic formalism for reasoning about the user's beliefs. Like some of the other examples of dynamic BN construction, this integration illustrates that user modeling researchers may not have to abandon their current favorite inference techniques if they want to make use of numerical techniques for uncertainty management.

2.13. PPP: ANTICIPATING DIFFICULTIES WITH MULTIMEDIA PRESENTATIONS

The next two systems both address the same basic problem: How can a system that presents technical information predict what difficulties the user might have in interpreting it? The two systems approach the problem in quite different ways.

PPP is an interactive successor to the knowledge-based multimedia presentation system WIP (Wahlster et al., 1993). The user modeling component being developed by van Mulken (1996) uses BNs to take into account the psychological factors that determine whether $\mathcal{U}$ will understand a technically oriented multimedia presentation. For example, graphical symbols such as arrows and radiating lines can have several different meanings, some of which may be known to only a minority of users. A number of factors interact to determine whether a user $\mathcal{U}$ will understand a particular use of such a symbol. These include $\mathcal{U}$'s familiarity with graphical presentations, $\mathcal{U}$'s knowledgeability about the domain, the frequency with which the symbol in question is used in the intended meaning, the extent to which the current context suggests some unintended interpretation, and the amount of time that $\mathcal{U}$ has available for the decoding of the presentation. Some indirect evidence about the ways in which these factors interact is available from experimental psychological research, but to construct corresponding BNs the designer has to fill in a number of gaps.

Because PPP is an interactive presentation system, the user's behavior serves as evidence about how $\mathcal{U}$ reacted to particular aspects of a presentation. This evidence is rather meager, consisting mainly of menu-selected comments and questions (e.g., "What does this mean?") in cases where $\mathcal{U}$ is not entirely satisfied with a presentation. Through upward propagation, the system can learn—gradually at first—about the user's graphical and domain knowledge; and also, in the long run, about the difficulty of the symbols that the system uses.

2.14. VISTA-III: USING BUGGY NETWORKS TO ANTICIPATE USERS' INFERENCES

Horvitz and Barry (1995) address a different type of difficulty that can arise with computer-based information presentation: The amount of information available for presentation can be immense, and it may consist largely of information that $\mathcal{U}$ does not need in order to make a decision. Especially if

$\mathcal{U}$ is under time pressure, $\mathcal{S}$ should try to find some small subset of the available information that will enable $\mathcal{U}$ to make an appropriate decision. Horvitz and Barry's approach to this problem was developed in the context of the system VISTA-III, which supports an operator who is monitoring the propulsion system of the Space Shuttle.

One of the many subtasks that is frequently performed by the system is that of predicting, given a particular subset of the available information, what inferences $\mathcal{U}$ would make if that subset of the information were displayed. The authors' approach presupposes that the system possesses a *gold-standard* BN which is capable of making expert-level inferences about the state of an engine. This BN represents links between (a) observable variables (largely data from sensors at various parts of the engine) and (b) states of the engine that are not directly observable.

If the current user herself happens to be an expert, the system has a straightforward way of predicting how she will interpret a given display: It feeds the information contained in the display into the BN and checks to see what inferences are made. For less knowledgeable users, BNs designed with the help of expert trainers are used which typically lack some of the more subtle nodes and links of the gold-standard BN. These nonexpert BNs are analogous to the incomplete or buggy models often used in intelligent tutoring systems to model student knowledge (cf., e.g., Section 3.4).

Note that whereas van Mulken (1996) uses BNs to manage uncertainty about $\mathcal{U}$, Horvitz and Barry use them in effect as simulation models of $\mathcal{U}$'s cognitive processes. This use presupposes that human inference with this type of problem is basically similar to causal inference in BNs. The question of the human-likeness of Bayesian reasoning will be raised again in Section 5.5.

2.15. FURTHER TYPES OF UNCERTAINTY

Mislevy (1994) discusses a number of subtle types of uncertainty that can arise in the context of educational assessment and demonstrates how they can be handled with BNs. Although these particular methods have apparently not yet been integrated into interactive tutoring systems, they are good candidates for such integration in that they deal with general, recurrent problems. The following are examples of the issues considered:

1. How can contextual factors, such as the student's chance familiarity with the topic of a passage being studied, be modeled explicitly so that they don't contribute excessive noise to the diagnostic process?
   Pynadath and Wellman (1995; cf. Section 2.8 above) showed how more concrete contextual factors can be modeled.
2. How can inferences be made about a learner's cognition in cases where two or more entirely different approaches to a problem are available and it is not initially known which one the learner is pursuing?
   One approach to this question was mentioned in Section 2.9.
3. How can higher-order uncertainty concerning the most appropriate initial beliefs about an unknown student be managed?
   The difficulty that BN designers often encounter in specifying prior probabilities constitutes a frequently mentioned motivation for employing DST instead (cf. Section 3.1.1). Mislevy shows how uncertainty about prior probabilities can be represented and managed explicitly within a BN.

2.16. CONCLUDING REMARKS ON BAYESIAN NETWORKS

The systems reviewed in this section have illustrated that BNs can be applied to user modeling in many different ways. Variables on all of the levels shown in Figures 3 and 5 have been incorporated;

predictive and diagnostic inference have been smoothly integrated; and various ways have been explored of constructing the networks dynamically.

Some of the drawbacks that have been attributed to the BN paradigm can best be discussed in the following sections on DST and FL, since these paradigms aim to address some of these issues. Questions concerning the practical usability of BNs will be considered in Section 5.

3. Systems That Have Used the Dempster-Shafer Theory of Evidence

A typical case where the application of DST is often judged to be appropriate is that of an unreliable witness (see, e.g., Shafer & Tversky, 1985). Suppose, for instance, that you ask a UNIX consultant about the expertise of a particular user $\mathcal{U}$. The consultant says "I think I know the person you mean; if she's the one I have in mind, all I can remember is that she's not a novice user". This evidence is of some use, but it is difficult to conceive of the statement as an event that had a particular probability of being *caused* by $\mathcal{U}$'s expertise, as in the most typical examples of BNs. Also, when dealing with this evidence, it is somewhat unnatural to update directly your assessment of the likelihood that $\mathcal{U}$ is, say, an intermediate-level user, as was done with the BN depicted in Figure 1, because the consultant said nothing about that specific level.

To continue the example, you might ask a second consultant who hedged his answer in the same way, expressing the judgment that $\mathcal{U}$ was "an intermediate or expert user"; and a third consultant might offer the contradictory recollection that $\mathcal{U}$ was "a novice or a beginner". It is not obvious what to make of this set of statements, and DST offers some subtle methods.

An unreliable witness in the literal sense may appear in a user modeling context when the user (or another person, such as a teacher) is asked to provide information about the user. This especially natural application of DST has apparently not yet been attempted, however. Instead, the evidence used by the four systems that will be reviewed in this section consists of observable user actions. The way in which such evidence can be handled like the reports of a witness can be illustrated with the first example, which is based on an adaptive testing system described by Petrushin and Sinitsa (1993) and by Petrushin et al. (1995).[7]

3.1. SOPHISTICATED PROCESSING OF SIMPLE OBSERVATIONS

Like the simple BN shown in Figure 1, Figure 7's system distinguishes four levels of expertise, labeled here for convenience with the integers 1 (NOVICE), 2, 3, and 4 (EXPERT). But evidence in the student's behavior is not linked only to these four hypotheses about $\mathcal{U}$; it is linked to all 10 *subsets* of contiguous levels, each of which is associated with a histogram in Figure 7A. Within DST, evidence could also be treated as supporting a noncontiguous set of hypotheses such as $\{1, 3\}$. But the nature of the evidence available to the example system ensures that this will never be necessary. Therefore, noncontiguous subsets are not taken into account by this system.

As before, the three difficulty levels SIMPLE, MUNDANE, and COMPLEX are distinguished. A piece of evidence (e.g., the fact that $\mathcal{U}$ knows the SIMPLE concept "MORE") is in effect treated like a statement by an unreliable witness: The witness is 70% sure that $\mathcal{U}$'s level belongs to the set $\{2, 3, 4\}$ of levels on which knowledge of a SIMPLE concept is likely; but there is a 30% chance that the witness has the wrong

[7] A more formal introduction of the central concepts of DST is given in this issue by Bauer (1995, section 5). A longer but still accessible introduction is provided by Gordon and Shortliffe (1984). Several of the papers collected by Shafer and Pearl (1990) provide further technical background and/or examples of applications. Pearl (1988, section 9.1) offers a challenging alternative perspective on DST. Yager et al. (1994) present a collection of articles that describe recent advances in research on DST.

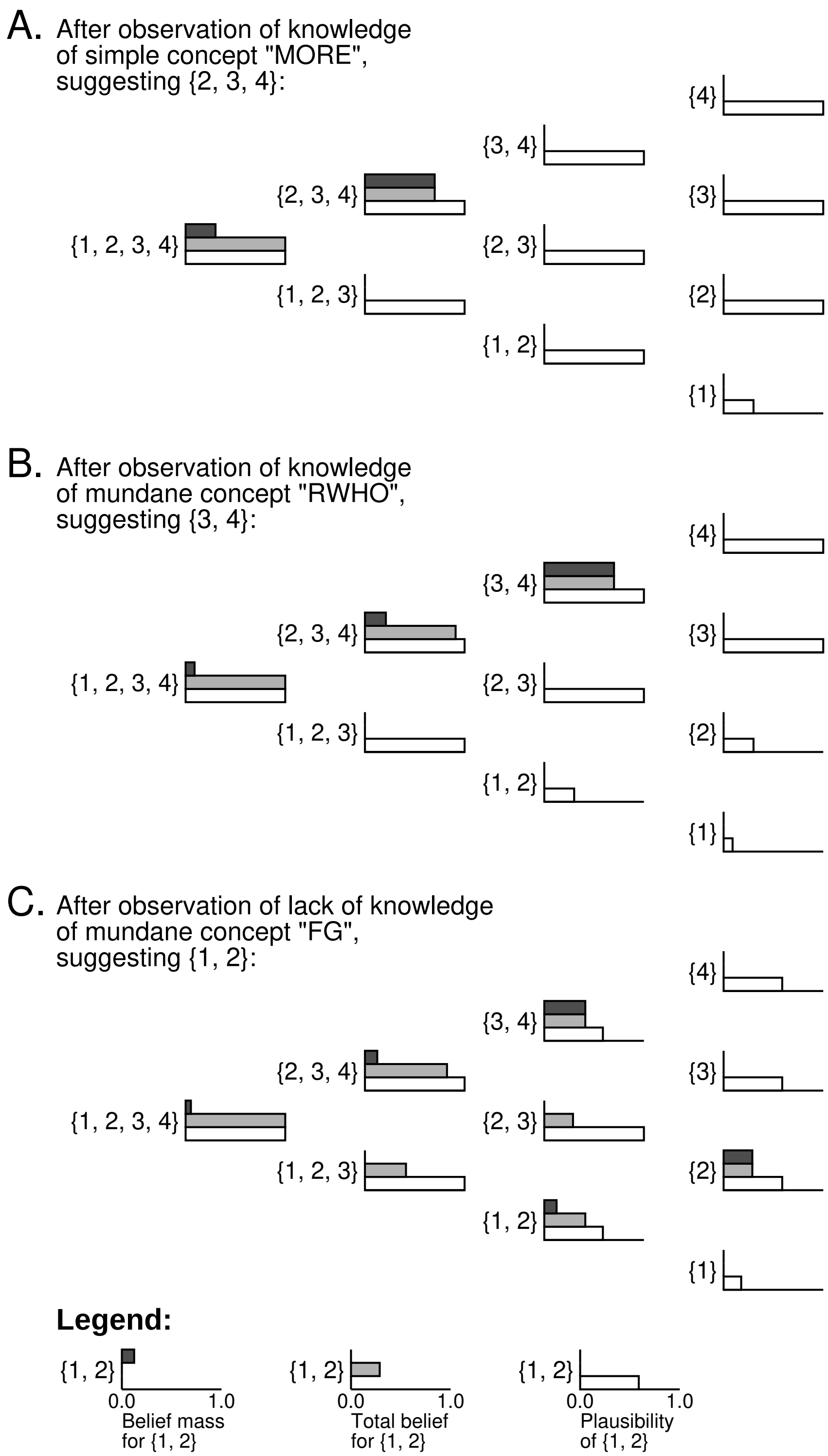

Figure 7. Processing with DST of evidence concerning a user's expertise level (cf. Section 3.1).

person in mind, in which case we know only that $\mathcal{U}$'s level belongs to $\{1, 2, 3, 4\}$. More precisely, each instance of knowledge or lack of knowledge is associated with an assignment of *belief mass*[8] to one or more of the hypothesis subsets, the total amount of belief mass for all subsets being 1.0. In the present example, a mass of .7 is assigned to the set of expertise levels with which the evidence is compatible, while .3 is assigned to the set $\{1, 2, 3, 4\}$ of all levels. The darkest bars in the histograms in Figure 7A illustrate this assignment of belief mass to the two relevant sets of hypotheses after the observation that $\mathcal{U}$ knows "MORE". The figure also illustrates two further basic DST concepts:

1. The total *belief* associated with a hypothesis set H (represented by the light gray bar for H) includes not only the belief mass specific to H but also the sum of the masses of all of the subsets of H. This index represents the extent to which $\mathcal{S}$ should believe that $\mathcal{U}$'s level is one of the levels in H.
2. The *plausibility* of a hypothesis set H (represented by the white bar associated with H) is the sum of the belief masses of all hypothesis sets that have at least one hypothesis in common with H. This index reflects the extent to which $\mathcal{U}$'s level *might* be within H according to the evidence processed so far.

In Figure 7B, $\mathcal{S}$ has just determined that $\mathcal{U}$ knows the MUNDANE concept "RWHO". This evidence is combined with the previous evidence through an application of *Dempster's rule of combination* (see, e.g., Bauer, 1995, section 5). This rule plays a role analogous to that of Bayes' rule within BNs. The result is intuitively plausible: It seems quite likely that $\mathcal{U}$ belongs to one of the levels in the set $\{3, 4\}$, but there is as yet no reason to believe specifically that she belongs to Level 3; and the same is true of Level 4 (cf. the absence of any belief associated with $\{3\}$ or with $\{4\}$, although both hypothesis sets are considered fully plausible). When the third piece of evidence has likewise been integrated (Figure 7C), the belief moves farther toward the specific hypothesis sets. $\mathcal{S}$ now for the first time assigns belief mass to a *singleton* hypothesis set, $\{2\}$.

3.1.1. *Comparison with Bayesian Networks*

Even this simple example illustrates some characteristic differences between DST and the BN approach.

1. Note that the system did not start with any a priori belief about $\mathcal{U}$'s expertise level: Figure 7 reflects solely the evidence obtained from observations. By contrast, in Figure 1 the node UNIX EXPERTISE OF $\mathcal{U}$ had to be initialized with some prior belief. Often this is unproblematic; but for example, if a tutoring system is being deployed for the first time in a new school with a different type of pupil than before, there may be no way of obtaining a meaningful prior distribution. In a BN, such cases are often handled through the assignment of equal prior probabilities to all hypotheses; but as advocates of DST point out, this method does not adequately distinguish between a state of *ignorance* about a variable and a genuine belief that all of its values are equally probable. It also means that valuable observational evidence may end up being combined with largely arbitrary prior assumptions.
2. Assigning a particular belief mass to a set of hypotheses is sometimes a natural way of expressing the fact that a piece of evidence in no way discriminates among the members of that set.
3. Basing a decision on the results of the analysis in this example is more complex than it is when a BN is involved; this feature can be viewed as either an advantage or a disadvantage. Suppose, for example, that the system at some point has to judge whether $\mathcal{U}$ belongs to Level 2. (Since the original system is an adaptive testing system, it has to make this type of decision in order to know when it can stop presenting items to the student.) With a BN, each hypothesis is associated with a single probability. With DST, for each hypothesis *set* there are three different measures of the

[8] This assignment is often called a *basic probability assignment*, but this term will not be used here, as it might lead to confusion with the probabilities used in BNs.

extent to which it is compatible with the evidence (two of which—the belief mass and the total belief—are always equal for singleton hypothesis sets). The designer can choose among various possible decision criteria, taking into account the context in which the decisions are to be made. For example, the criteria used by Petrushin et al. (1995) for deciding whether $\mathcal{U}$ belongs, say, to Level 2 actually do not refer to the measures associated with the singleton subset $\{2\}$ but rather to those associated with the subsets $\{2, 3, 4\}$ and $\{3, 4\}$. A more detailed discussion of the decision-making problem, with a worked example, is provided by Bauer (1995, section 8).

4. The example system does not provide a way to make *predictions* about $\mathcal{U}$'s knowledge or behavior; it is designed to support diagnostic inference. Yet the BNs reviewed in Section 2 have shown that predictive inference is often useful in a user modeling system.

3.1.2. *Sketch of an Alternative Treatment*

The fourth point just mentioned is symptomatic of a more general difference between DST and BNs: The basic DST theory provides a way of combining pieces of evidence about a single variable; it does not deal with the propagation of beliefs about one variable to beliefs about related variables. A number of researchers have, however, worked on extensions of DST to the processing of belief networks.[9] Within these extensions, the type of propagation used in BNs actually emerges as a special case. These methods deserve the attention of user modeling researchers who find DST appealing but who do not want to do without the ability to deal with networks such as those of the systems described in Section 2.

If this type of approach is applied to the first part of our example, the system maintains beliefs not only about the variable UNIX EXPERTISE OF $\mathcal{U}$ but also about DIFFICULTY OF "MORE" and KNOWLEDGE BY $\mathcal{U}$ OF "MORE". In addition, $\mathcal{S}$ has a belief about the *relationship* among these three variables, which corresponds to the conditional probability table underlying the inferences shown in Figure 1.[10] When $\mathcal{S}$ obtains information about one of the variables, its new belief about that variable in turn affects the other two beliefs; the propagation method makes use of Dempster's rule of combination, albeit in a more complex way than the procedure illustrated above. In short, basically the same predictive and diagnostic inferences can be made as were made in the BN of Figures 1 and 2. For instance, $\mathcal{S}$ can now predict whether $\mathcal{U}$ knows a concept, even if $\mathcal{S}$ is uncertain about that concept's difficulty.

Like the example system, the three other DST systems to be reviewed in the rest of this section do not use DST belief network techniques. This fact is reflected in the relatively small number of solid arrows in the overview in Figure 8.

3.2. JUDICIOUSLY SELECTING DEFAULT PLAN ASCRIPTIONS

Carberry (1990) introduced DST to user modeling with her method for the default ascription of goals in a plan recognition system. The example domain is student consulting. The problem is to determine, given a known goal of $\mathcal{U}$, which higher-level goal $\mathcal{U}$ is likely to be pursuing. DST is used to quantify the extent to which a given goal suggests a particular higher-level goal. For example, Table I shows the belief mass assignment for the piece of evidence that $\mathcal{U}$ wants to earn credit in the course M370. It expresses the belief that this mathematics course points quite strongly to a goal of majoring in Math and much less strongly to the goal set $\{$Math major, Computer Science major$\}$. Roughly similar

[9] Shenoy (1994) and Dempster and Kong (1988) offer largely theoretical expositions, while Zarley et al. (1988) describe an implementation that supports the interactive graphical specification of DST-based belief networks.

[10] Like the beliefs about the individual variables, this belief is represented by an assignment of belief mass; but each hypothesis now corresponds to one of the 24 possible combinations of values of the individual variables.

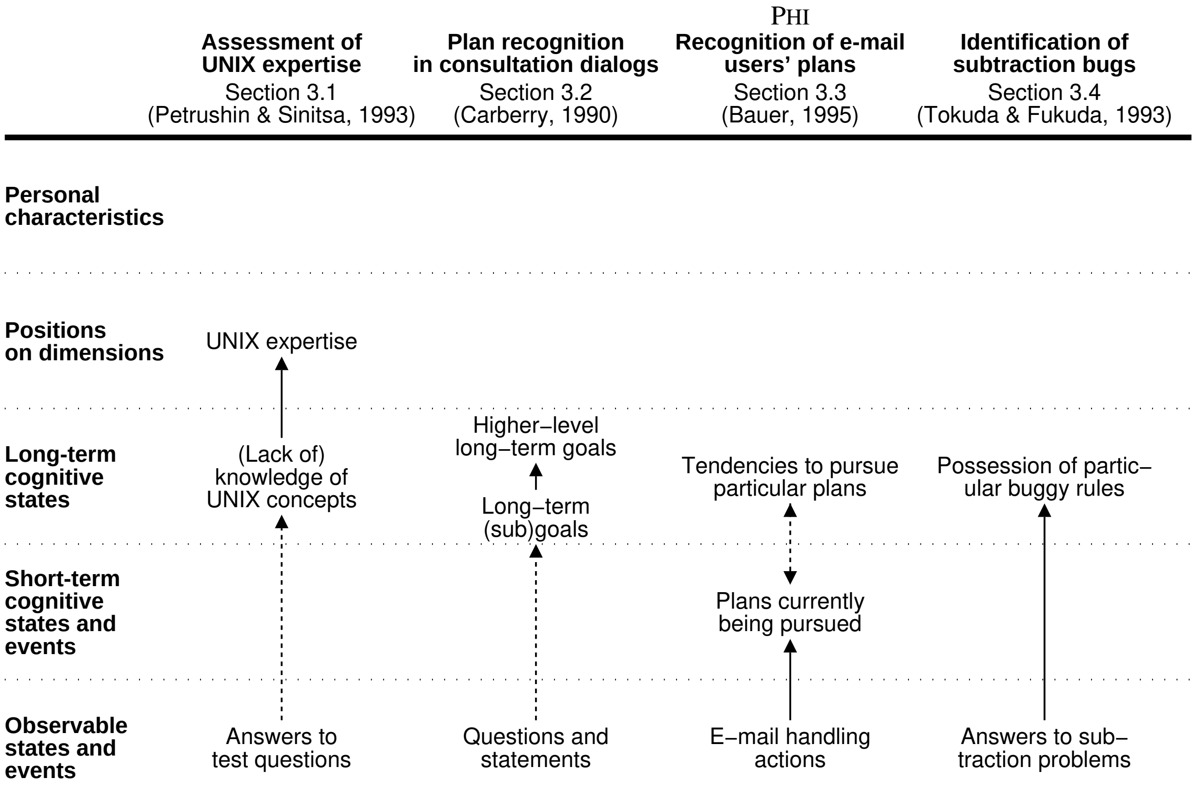

Figure 8. Systems that have used DST for user or student modeling (cf. Figure 3 and Appendix A).

links could be quantified within a Bayesian framework through the assignment of different conditional probabilities of taking M370 given different majors (high for Math majors, low for Computer Science majors, and very low for all other majors). The question arises, then, under what circumstances the DST belief mass assignment might be more natural or realistic. A case corresponding to the typical case of an unreliable witness occurs if the advisor cannot quite remember what majors M370 is open to: She is 85% sure it's open only to Math majors, but she allows a 12% chance that it's open to both Math and Computer Science majors and a 3% chance that any student can take the course. Carberry does not address the issue of the specific appropriateness of DST on this level.

When several observations are available that constitute evidence concerning $\mathcal{U}$'s goal, the evidence is integrated with Dempster's rule of combination, as in Figure 7.

Given the evidence for the ascription of a given goal, the system must decide whether the evidence is strong enough to warrant ascribing the goal to $\mathcal{U}$. The system's criterion is that

1. the goal must have a plausibility exceeding some threshold (here: .9); and
2. this plausibility must exceed that of the next-most-plausible goal by some threshold (here: .7).

After the single observation reflected in Table I, these criteria are fulfilled by the goal of majoring in Math.

The purpose of the second criterion is evidently to ensure that a goal is not ascribed simply because there is no evidence that speaks *against* it. A more straightforward way of formulating this criterion in

Table I. A Use of DST to Process Evidence About a User's Goals

Set of Possible Goals	Belief Mass	Total Belief	Plausi-bility
{Math major}	.85	.85	1.00
{Computer Science major}	.00	.00	.15
{Math major, Computer Science major}	.12	.97	1.00
{[All possible goals]}	.03	1.00	1.00

Note. Adapted from part of Figure 2 of Carberry (1990). The assignment of belief mass constitutes the system's interpretation of the fact that the user wants to earn credit in the course M370.

terms of DST concepts would be to require that the belief associated with the goal, which summarizes the strength of the evidence favoring that goal, must exceed some threshold (cf. the second column from the right in Table I). There may be different arguments for different criteria; but in any case, as was seen in the previous subsection, DST does supply a relatively differentiated description of the available evidence, which may make it easier for a designer to choose an appropriate criterion.

Carberry's system also departs from the straightforward use of DST in a more radical way: Once a goal $\mathcal{G}$ has been tentatively ascribed to $\mathcal{U}$, it is in turn used as evidence for the purpose of ascribing still higher-level goals; but the uncertainty underlying the ascription of $\mathcal{G}$ is not propagated upward, as it could be with the propagation methods sketched in Section 3.1.2. Instead, $\mathcal{G}$ is treated as if it could be ascribed with certainty. Carberry's justification for this procedure does not concern the computational cost that would be associated with propagation. Rather, it is based on psychological evidence that people, when performing multistage inferences, similarly refrain from propagating uncertainty from one stage to the next. The reason why this consideration is relevant is that the system can presumably explain its reasoning to the user more easily if this reasoning resembles human reasoning. The price to be paid for this possible increase in explainability is, of course, that the system's reasoning can only be justified locally in terms of DST; the confidence with which the system can ascribe a goal on the basis of a sequence of inferences involving at least two goals cannot be determined through an application of DST principles. This partial abandonment of a formal framework with a view to enhancing the human-likeness of reasoning will be seen to a greater extent in the systems that use some form of FL (Section 4). Its consequences will be considered in Sections 5.5 and 5.6.

3.3. PHI: Combining Evidence from the Past and the Present

The plan recognition system of Bauer (1995, in this issue) is a part of the intelligent help system PHI. It processes evidence about plans that an e-mail user may be pursuing. Bauer distinguishes between *basic plans* and *abstract plans*; each of the abstract plans may be realized by one of various basic plans. This abstraction hierarchy means that there exist natural subsets of plans—those which realize the same abstract plan—about which evidence can be obtained. Recall that in Carberry's (1990) system it was not obvious how an observation could suggest that $\mathcal{U}$ had *one of the two goals* {Math major, Computer Science major}; it is a bit easier to see how an observation might suggest that $\mathcal{U}$ is pursuing the abstract plan of storing messages without suggesting whether $\mathcal{U}$ intends to save them or to write them. (Similarly, naturally occurring hierarchies of diseases have constituted one of the motivations for applying DST to medical diagnosis—cf. Gordon & Shortliffe, 1984.)

3.3.1. *A DST Treatment of Information from Previous Sessions*

Because Bauer (1995) uses DST consistently throughout his plan recognition component, he faces the challenge of applying it to problems for which it is not obviously applicable. An example is the use of $\mathcal{U}$'s behavior in *previous* sessions as evidence about what plans she is pursuing in the current session. On the one hand, it makes sense to try to make use, for example, of the fact that in the sessions observed so far $\mathcal{U}$ was mainly occupied with the deleting of e-mail messages. If the number of sessions has been large, these observations can be assumed to reflect stable dispositions of $\mathcal{U}$, so it is reasonable that they should determine the system's initial belief about $\mathcal{U}$'s behavior in the next session, as they do in Bauer's system. But if the number of sessions so far has been small (e.g., 2), the system must be uncertain as to how well $\mathcal{U}$'s deleting tendency will extrapolate into the future (e.g., the extent to which it may have been due so far to situational factors such as having exceeded the disk quota).

It is not obvious how this type of uncertainty can best be handled with the basic DST principles for processing evidence. On the other hand, there is a fairly natural solution in terms of a belief network of related variables (realizable with BNs or with the extension of DST sketched in Section 3.1.2): The system can maintain beliefs about various dimensional variables such as INTEREST IN WRITING E-MAIL and TENDENCY TO DELETE E-MAIL MESSAGES. The system's beliefs about these variables can be updated on the basis of $\mathcal{U}$'s behavior from one session to the next, and they will become more and more definite. They will therefore have little impact on the interpretation of $\mathcal{U}$'s behavior in earlier sessions, but their influence will increase with each session.

Bauer essentially approximates the behavior of such a network by introducing as the first session of a new user a fictitious session in which the system was unable to make any inference at all as to what plans the user was executing. The effect of this fictitious session is to ensure that $\mathcal{S}$ will have less definite expectations about what $\mathcal{U}$ will do during the early real sessions. As more and more real sessions are observed, the impact of the fictitious session declines. In this way, the desirable inference behavior sketched above is approximated—but in a rather arbitrary way, one which is not justifiable in terms of the principles of DST. For example, why not introduce *two* such fictitious sessions instead of just one? That would make the system even more cautious about extrapolating on the basis of $\mathcal{U}$'s previous behavior; but how cautious should $\mathcal{S}$ be?

To be sure, there is also no automatic answer to this last question if a belief network is used as sketched above. In that context, the question concerns the exact relationship between the general dimensional variables and $\mathcal{U}$'s choice of plans in a concrete situation: $\mathcal{S}$ could presuppose either a high or a low degree of predictability of $\mathcal{U}$'s plans. But within the belief network conceptualization, the question at least concerns the details of a part of the model that is constructed within the theory.

3.3.2. *An Alternative Machine Learning Treatment*

In a recent extension of his system, Bauer (1996) introduces a more differentiated way of interpreting evidence from the user's behavior during previous sessions. The system no longer just records the overall frequencies with which particular plans were identified. Instead, $\mathcal{S}$ also notes the specific *situation* in which a plan was being executed on each occasion. In this way, $\mathcal{S}$ can form hypotheses about how the chosen plan depends on the situation. For the formation of these hypotheses, the system employs machine learning techniques. For example, it uses ID3 (Quinlan, 1983) to construct a decision tree which relates features of situations to typical behaviors of the current user in those situations. So now the system no longer has to use simple rules such as

▹ When $\mathcal{U}$ is reading a message in a context consistent with the hypothesis that she plans to store it, assign a belief mass of .8 to this hypothesis.

Instead, $\mathcal{S}$ has a much larger number of rules like the following one, which refer to attributes of the situation:

▹ When $\mathcal{U}$ is reading a message from a colleague and the message is at least 50 lines long, … assign a belief mass of .8 to the hypothesis that $\mathcal{U}$ plans to store the message.

One complication with the previous, simpler approach is inherited by this extension: The *reliability* of the evidence derived from previous sessions has to be assessed. For example, the number .8 might be based on the occurrence of 4 positive cases among 5 observations or on the occurrence of 80 positive cases among 100 observations. Bauer's method in effect treats the results of the machine learning procedure like the testimony of an unreliable witness, assuming a particular likelihood that its predictions are invalid. This likelihood decreases as the number of observations that the machine learning procedure has made use of increases; but it is unclear whether there is a theoretically justifiable way of determining this relationship. In other words, the original issue of how many fictitious sessions should be introduced now reappears in the form of the question of how the reliability of the machine learning results should be quantified.

The use of machine learning in this context does have advantages, however. It would be more difficult to discover highly specific regularities in the user's behavior with the normal evidence-evaluation methods of DST (or BNs or FL); machine learning techniques are in general better equipped to deal efficiently with the large number of potential regularities that have to be considered. So an important issue is: How can machine learning techniques be integrated with uncertainty management techniques in such a way that the uncertainty associated with the results of the machine learning is taken into account in a principled manner? This issue is especially important in the context of user modeling, because in this context the amount of data that a machine learning technique can make use of is often small. To date, successful applications of machine learning techniques to user modeling tasks (see, e.g., Maes, 1994; Orwant, 1995; Sheth & Maes, 1993) have typically involved contexts in which a large number of observations of user behavior have been available. The integration of machine learning and uncertainty management techniques has received more attention in connection with BNs and FL (see Sections 2.7, 5.1, and 5.3) than in connection with DST.

3.4. Combatting the Multiplication of Bug Subsets

DST deals with sets of hypotheses, which are of course more numerous than individual hypotheses. This property can lead to serious problems of computational complexity in cases such as the one addressed by Tokuda and Fukuda (1993): A student $\mathcal{U}$ solving subtraction problems is assumed to possess exactly one of a set of 36 known incorrect subtraction rules, or *bugs*. Given the answers of $\mathcal{U}$ to a series of problems, how can $\mathcal{S}$ determine which bug $\mathcal{U}$ possesses?

A straightforward application of DST methods to this problem can be sketched (with some simplification) as follows: When $\mathcal{U}$ gives an incorrect answer $\mathcal{A}$ to a problem $\mathcal{P}$, assign a certain amount of belief mass to the subset of bugs that would yield exactly the answer $\mathcal{A}$ to $\mathcal{P}$; when the next incorrect answer is observed, apply the same rule and combine the results by applying Dempster's rule of combination; continue until the belief associated with some singleton hypothesis set containing one bug has become sufficiently strong.[11]

A problem with this approach is that there are $2^{36} - 1$ nonempty subsets of the set of 36 bugs. To be sure, only a small proportion of these subsets would ever have any belief mass assigned to them by

[11] This sketch is simplified in that the system should also interpret $\mathcal{U}$'s correct and incorrect answers as evidence *against* $\mathcal{U}$'s possession of bugs which would lead to different answers.

the procedure just sketched; but even these could become impractical to process if more than a few observations were considered.

Tokuda and Fukuda adopt a computationally simpler procedure, following an approach applied in some previous applications of DST (e.g., by Gordon & Shortliffe, 1984). They divide the 36 bugs into 3 basic classes such that the bugs within each class produce incorrect answers on a particular type of problem (e.g., problems that require borrowing across more than one column). When $\mathcal{U}$ gives an incorrect answer $\mathcal{A}$ to a problem $\mathcal{P}$, belief mass is assigned to the following bug subsets, among others:

- the basic class of bugs that would produce incorrect answers on $\mathcal{P}$ (not necessarily the specific answer $\mathcal{A}$); and
- each of the singleton hypothesis subsets whose bug would produce exactly the answer $\mathcal{A}$ on $\mathcal{P}$.

The authors compare three variants of this procedure, all of which rate much better than the straightforward use of DST in terms of their computational complexity. Moreover, when the procedure is applied to artificial sets of answers generated by buggy models, it succeeds in identifying the underlying bugs. On the other hand, the belief mass assignments that the procedure prescribes are in general not the same as the ones that would be dictated by the underlying logic of DST. For instance, when an incorrect answer could have been caused by any of several bugs, there is no reason to assign belief mass to any singleton subset containing just one of these bugs. This departure from the basic logic of DST could be more problematic if the procedure were applied under less idealized circumstances—for instance, where the data came from real students who inconsistently applied varying numbers of buggy rules.

3.5. CONCLUDING REMARKS ON DEMPSTER-SHAFER THEORY

The systems examined in this section have illustrated several properties of a possible user modeling application which suggest that the use of DST should be considered. These properties do not, however, constitute necessary or sufficient conditions for the selection of DST, and in fact none of the systems described in this section exhibit all of them.

1. The total set of hypotheses has a structure that makes it possible to restrict attention to a limited number of hypothesis subsets.
2. For at least some of the variables about which inferences are to be made, it does not appear meaningful to specify prior probabilities.
3. The relationships between hypotheses and evidence cannot naturally be conceptualized as causal relationships.
4. When the system makes decisions on the basis of its inferences, its decision criteria can make good use of concepts such as those of the strength of belief in a hypothesis subset and the plausibility of a hypothesis subset.
5. The emphasis is to be on the accurate representation of subtle relationships between evidence and hypotheses rather than on the construction of large networks of variables which exhibit basically straightforward relationships.

Further considerations will be discussed in Section 5.

The number of user modeling systems that have used DST is much smaller than the number that have used BNs, and a roughly similar relationship is found in the total amounts of research devoted to the development of these two approaches to uncertainty management. Therefore, user modeling researchers who adopt DST in the near future will have to be relatively independent and enterprising.

4. Systems That Have Used Concepts from Fuzzy Logic

The term *fuzzy logic* is used in various senses, some broader than others (cf., e.g., Zadeh, 1994). It will be used here in a broad sense that encompasses any system that makes use of one or more typical concepts such as those of a *linguistic variable*, a *fuzzy set*, or a *fuzzy if-then rule*.[12]

The key concept of a *fuzzy set* was introduced as a way of dealing with the imprecision, or vagueness, that is typical of natural concepts. Although degrees of set membership are sometimes confused with measures of uncertainty, these are in fact quite different things. On the other hand, a variety of concepts and techniques developed within the tradition of FL have been used for the management of uncertainty, for example in expert systems; these include *possibility theory* (see, e.g., Zadeh, 1981) and the adaptation of probability theory and DST to the use of imprecisely specified probabilities and degrees of belief (e.g., by Lamata & Moral, 1994).

With regard to uncertainty management in user modeling, the appeal of FL appears to be based mainly on two quite different considerations:

1. People often reason in terms of vague concepts when dealing with situations in which they experience uncertainty. Consider, for instance, the statement: “This student is *quite advanced*, so he *ought* to be able to handle this task *fairly well*”. The vague concepts reflect the speaker's uncertainty about how advanced the student is and what his chances are of being able to handle the task with any particular degree of success. Many systems based on FL take advantage of FL's techniques for representing and reasoning with vague concepts to mimic this human style of reasoning. If a user modeling system adopts this approach, its reasoning may be especially easy for designers and users to understand and/or to modify.
2. When users supply explicit information about themselves to a system, they may express this information vaguely (e.g., “I don't know very much about the World-Wide Web”)—perhaps because they themselves do not have precise knowledge, or perhaps because they are for some reason not motivated or able to express their knowledge precisely. In any case, the user's vagueness leads to uncertainty in the system, as the system cannot in general infer an exact value. The concepts and techniques of FL do not provide the only possible answers to the question of how such uncertainty should be represented and processed, but they can form useful parts of a solution even when other uncertainty management techniques are used as well.

4.1. KNOME: SUBSTITUTING FUZZY RULES FOR THE LAWS OF PROBABILITY

The first example will concern the problem that was analyzed in Sections 2.1 and 3.1: that of updating an expertise estimate on the basis of evidence that particular concepts are known or not known. These previous analyses showed that this problem lends itself quite well to treatment with relatively traditional approaches—and therefore does not satisfy a criterion that is often applied to motivate the use of FL. But a basically fuzzy treatment was worked out in the mid-1980's by Chin (see, e.g., Chin, 1989) for KNOME, the user modeling component of the UNIX CONSULTANT.

As is illustrated in Figure 9A, KNOME represents likelihoods and likelihood changes in terms of a *linguistic variable* with 9 discrete values.[13] Four levels of user expertise are distinguished, as in the previous treatments of the example, as well as three levels of difficulty for concepts. There is also a

[12] Useful collections of papers in this area include those compiled by Dubois et al. (1993) and by Yager and Zadeh (1992). Widely used monographs include those by Kosko (1992) and by Cox (1994). The relationship between FL and other artificial intelligence techniques is the subject of lively debate in a special issue of *IEEE Expert* that includes a controversial paper by Elkan (1994) together with a number of critical replies.

[13] Chin uses the same 9 labels for likelihoods and likelihood changes. For clarity, in the present discussion the second set of labels shown in Figure 9A will be used for likelihood changes.

A. Scale of likelihoods and likelihood changes

False	Very unlikely	Unlikely	Somewhat unlikely	Uncertain	Somewhat likely	Likely	Very likely	True
–4	–3	–2	–1	0	+1	+2	+3	+4
Impos-sible	*Much more unlikely*	*More unlikely*	*Somewhat more unlikely*	*Equally likely*	*Somewhat more likely*	*More likely*	*Much more likely*	*Certain*

B. Prediction rules

UNIX EXPERTISE OF USER	DIFFICULTY OF CONCEPT			
	SIMPLE	MUNDANE	COMPLEX	ESOTERIC
EXPERT	True	True	Likely	Uncertain
INTERMEDIATE	True	Likely	Unlikely	Uncertain
BEGINNER	Likely	Unlikely	False	Uncertain
NOVICE	Unlikely	False	False	False

C. Updating rules for presence of knowledge

UNIX EXPERTISE OF USER	DIFFICULTY OF CONCEPT			
	SIMPLE	MUNDANE	COMPLEX	ESOTERIC
EXPERT	*More likely*	*More likely*	*Somewhat more likely*	*Equally likely*
INTERMEDIATE	*More likely*	*Somewhat more likely*	*Somewhat more unlikely*	*Equally likely*
BEGINNER	*Somewhat more likely*	*Somewhat more unlikely*	*Impossible*	*Equally likely*
NOVICE	*Somewhat more unlikely*	*Impossible*	*Impossible*	*Impossible*

D. Updating rules for lack of knowledge

UNIX EXPERTISE OF USER	DIFFICULTY OF CONCEPT			
	SIMPLE	MUNDANE	COMPLEX	ESOTERIC
EXPERT	*Impossible*	*Impossible*	*Somewhat more unlikely*	*Equally likely*
INTERMEDIATE	*Impossible*	*Somewhat more unlikely*	*Somewhat more likely*	*Equally likely*
BEGINNER	*Somewhat more unlikely*	*Somewhat more likely*	*More likely*	*Equally likely*
NOVICE	*Somewhat more likely*	*More likely*	*More likely*	*Somewhat more likely*

Figure 9. KNOME's rules for predicting and interpreting a user's (lack of) knowledge of UNIX concepts (cf. Section 4.1).

fourth category, ESOTERIC, which does not really represent a difficulty level, in that knowledge of these concepts is by definition hard to predict on the basis of the user's expertise. Figure 9B summarizes what are actually 16 fuzzy if-then rules like the following one:

▷ IF $\mathcal{U}$ is a BEGINNER AND the concept $\mathcal{C}$ is SIMPLE,
THEN it is LIKELY that $\mathcal{U}$ knows $\mathcal{C}$.

These rules express the same type of assumption as the conditional probability table of the BN shown in Figures 1 and 2. One argument for using the fuzzy versions is that precise, reliable conditional probabilities are unlikely to be available and that it is best to represent explicitly the system's imprecise knowledge of the relationship among these three variables.

The system is at most times uncertain about $\mathcal{U}$'s expertise level. The top of Figure 10 shows how, even before $\mathcal{S}$ makes any observations, each level is associated with some value of the linguistic likelihood variable. This representation of a belief is similar to the BN representation in terms of a probability distribution (cf. the histogram for the node UNIX EXPERTISE OF u in Figure 1).

4.1.1. *Predictive Inference*

Predicting whether $\mathcal{U}$ knows a given concept is straightforward if $\mathcal{S}$ has definite beliefs about both $\mathcal{U}$'s expertise and the concept's difficulty; in this case, only one of the rules in the prediction table (Figure 9B) is applicable. If, on the other hand, KNOME still has an indefinite belief about $\mathcal{U}$ like the one shown at the top of Figure 10, $\mathcal{S}$ bases its prediction on the level with the highest likelihood (here: BEGINNER). That is, it does not take into account the likelihoods associated with the other levels; this would require some way of combining the predictions made for the levels to which $\mathcal{U}$ might belong, as is done with downward propagation in BNs. Many systems based on FL do in fact apply interpolation techniques in such cases (cf. Section 4.1.3 below). KNOME's method could be extended in this way, but this extension would require the introduction of explicit *membership functions* for the various fuzzy concepts involved, as well as a more complex mechanism for applying the fuzzy rules.

4.1.2. *Diagnostic Inference*

To interpret observations concerning particular concepts that $\mathcal{U}$ knows or does not know, KNOME in effect uses the two further tables of fuzzy rules labeled C and D in Figure 9. The first entry in Table C corresponds to the following if-then rule:

▷ IF the concept $\mathcal{C}$ is SIMPLE AND $\mathcal{U}$ knows $\mathcal{C}$,
THEN it now seems MORE LIKELY that $\mathcal{U}$ is an EXPERT.

The term *more likely* refers to a likelihood change relative to the currently assumed likelihood that $\mathcal{U}$ is an expert. The current likelihood and the likelihood change are to be combined additively according to the scale shown in Figure 9A. For example, if $\mathcal{S}$ currently considers it to be UNCERTAIN (0) whether $\mathcal{U}$ is an expert, the fact that this classification has just become MORE LIKELY (+2) means that $\mathcal{S}$ should now consider it LIKELY (+2) that $\mathcal{U}$ is an expert.

In a BN, the likelihoods shown in Tables C and D would not be represented separately; instead, the appropriate adjustments to $\mathcal{S}$'s belief would be computed essentially with an application of Bayes' rule together with the conditional probabilities corresponding to Table B. In this sense, Tables C and D represent fuzzy approximations of Bayes' rule as it would be applied in this particular situation. The question arises, how well can such an approximation work? Figure 10 gives an example. The first updating occurs after $\mathcal{S}$ has observed that $\mathcal{U}$ knows the concept "MORE". As in Figure 2, $\mathcal{S}$ becomes slightly more optimistic about $\mathcal{U}$'s expertise. Similarly, after $\mathcal{U}$ is observed to know a second concept as

Expertise Level	Likelihood Change	Likelihood
Before observations:		
EXPERT		Uncertain
INTERMEDIATE		Uncertain
BEGINNER		Somewhat likely
NOVICE		Uncertain
After observation of knowledge of simple concept "MORE":		
EXPERT	*More likely*	Likely
INTERMEDIATE	*More likely*	Likely
BEGINNER	*Somewhat more likely*	Likely
NOVICE	*Somewhat more unlikely*	Somewhat unlikely
After observation of knowledge of mundane concept "RWHO":		
EXPERT	*More likely*	True
INTERMEDIATE	*Somewhat more likely*	Very likely $\rightarrow$ False
BEGINNER	*Somewhat more unlikely*	Somewhat likely $\rightarrow$ False
NOVICE	*Impossible*	False
After observation of lack of knowledge of mundane concept "FG":		
EXPERT	*Impossible*	True
INTERMEDIATE	*Somewhat more unlikely*	False
BEGINNER	*Somewhat more likely*	False
NOVICE	*More likely*	False

Figure 10. Updating of an assessment of user expertise by KNOME (cf. Section 4.1.2).

well, KNOME's assessment becomes more positive again. According to the updating rules, the likelihood of the level EXPERT now reaches the maximum value of True. KNOME's procedure specifies that when this happens, the hypothesis in question must be accepted and the likelihood False must be assigned to the other three hypotheses. Accordingly, even the two hypotheses INTERMEDIATE and BEGINNER, which would otherwise be associated with the likelihoods Very likely and Somewhat likely, respectively, are from now on regarded as False.

Now that one hypothesis has been accepted, no subsequent observations can change KNOME's belief. So even the information that $\mathcal{U}$ does not know the MUNDANE concept "FG" can no longer have any effect, even though this information would have been sufficient in itself to falsify the hypothesis that $\mathcal{U}$ is an EXPERT.

A system that employed BNs or DST would never completely reject a hypothesis—such as INTERMEDIATE in this example—which had been quite consistent with all of the available evidence. Such a system would therefore be able to take into account the relatively surprising result of the third observation.[14]

4.1.3. *Sketch of an Alternative Treatment*

This example problem could be handled in another way that would make more use of typical FL concepts. Note that Chin's rules do not take into account the fact that imprecise concepts such as EXPERT and INTERMEDIATE may overlap considerably. For instance, an intermediate user is an expert to a limited degree, so a statement that applies to an expert may also apply to an intermediate user to a limited degree. FL systems often exploit such relationships among concepts to reduce the number of rules that they use, thereby achieving a type of *data compression* (see, e.g., Zadeh, 1994). If this approach is taken to an extreme, the 12 prediction rules summarized in the middle three columns of Table B in Figure 9 can be replaced by the following two rules:

▷ IF the concept $\mathcal{C}$ is EASY AND $\mathcal{U}$ is an EXPERT
THEN $\mathcal{U}$ KNOWS $\mathcal{C}$.

▷ IF $\mathcal{C}$ is NOT EASY AND $\mathcal{U}$ is NOT an EXPERT
THEN $\mathcal{U}$ does NOT KNOW $\mathcal{C}$.

Similarly, except for the treatment of ESOTERIC concepts, each of the Tables C and D in Figure 9 could be replaced by a pair of rules. The first rule for Table C might read:

▷ IF $\mathcal{C}$ is DIFFICULT AND $\mathcal{U}$ KNOWS $\mathcal{C}$
THEN $\mathcal{U}$ is an EXPERT.

At first glance, each of these rules appears to apply to only one specific case. But when the rules are defined and interpreted in terms of imprecise concepts, together they cover a number of cases. The question of course arises, how well can rules like this approximate the relationship that the designer has in mind? The particular results obtained will depend in part on how the concepts that occur in the rules—including the operators AND and NOT—are defined (cf. Section 4.2 below). The results will also depend on which of the various procedures for processing fuzzy if-then rules is applied (see, e.g., Cox, 1994, chap. 6; Kruse et al., 1991, section 10.3; Yager, 1992).

Given the central role within KNOME of the rules in Figure 9, it may well be worthwhile to use a larger number of rules so as to approximate the intended relationship as well as possible. But vague rules such as the ones just mentioned could be useful in cases where there is little information about the true relationship and where effort cannot be expended to obtain more information. For example, a system for recommending movies might try to represent and make use of a database of expert opinions such as the following one:

▷ If you like a fast-moving plot and you don't mind a bit of violence, this movie may be right for you.

4.1.4. *Comments on the Treatments of the Example Problem*

This introductory example illustrates two general points about FL that are relevant to user modeling:

[14] Chin (1989, p. 106) notes that the inability of KNOME to adjust its belief about a user's expertise once it has accepted a particular hypothesis could be a drawback—especially if a single user's interaction with the system were to last longer than is usually the case with KNOME, long enough for $\mathcal{U}$ to learn enough about UNIX to advance to a higher expertise level.

1. Fuzzy rules such as those summarized in Figure 9 are in some ways easy for both users and system designers to understand. The vague concepts correspond to the way people naturally talk and think about things like likelihoods, user categories, and concept difficulties. In fact, the naturalness of KNOME's concepts and rules may be one reason why its method is one of the most widely cited and reproduced proposals in the user modeling literature.
2. Such procedures are relatively easy to fine-tune in arbitrary ways on the basis of experience. For example, rules for dealing with special cases such as ESOTERIC concepts (cf. the right-hand column in the tables in Figure 9) can be added if cases are noticed where the other rules do not appear to work well. More basic aspects of the procedure can also be fine-tuned. For example, if the rather strange result shown in Figure 10 were found disturbing, a designer could reduce the frequency of such results by changing the rules for updating probabilities in such a way that the value True was derived only under more extreme conditions. (Note that designers of systems based on BNs or DST cannot easily eliminate undesired results by modifying the laws of probability or the rules for the combination of evidence.) To be sure, such a change might lead to undesirable results of other types; it is simply not easy to approximate a principle such as Bayes' rule or Dempster's rule of combination satisfactorily with a set of fuzzy rules.

The following subsections will examine several user modeling systems, more recent than KNOME, that use at least some concepts from FL (Figure 11).

4.2. THE SALES ASSISTANT: MAXIMAL USE OF MINIMAL USER INPUT

If system designers and knowledge engineers are often more comfortable with fuzzy concepts than with precise numbers, the same is even more true of ordinary system users. This consideration is taken into account in the SALES ASSISTANT (Popp & Lödel, 1995, in this issue). Part of this system is responsible for predicting how a user will evaluate particular products. Consider, for example, the problem of asking $\mathcal{U}$ how important the attribute “Screen size” is in her evaluation of a personal computer. $\mathcal{U}$ is unlikely to want to specify a numerical *importance weight* (e.g., .17) for each such attribute, but she might be willing to state that the attribute is “quite important”. Fuzzy logic provides means for representing such vague input—in this case, means for viewing “quite important” as a fuzzy set of numbers—and for processing it in combination with other imprecise inputs.

The SALES ASSISTANT also illustrates a slightly less obvious way of using fuzzy concepts to process vague user input when it predicts how $\mathcal{U}$ would evaluate various possible specific screen sizes. Strictly speaking, $\mathcal{U}$ should be asked to specify some sort of *value function* (cf., e.g., von Winterfeldt & Edwards, 1986)—that is, a mapping of possible screen sizes onto their evaluations. Yet a user is not likely to want to specify such a function with any precision, especially given that a product also has a number of other relevant attributes for which the same problem arises. At most, $\mathcal{U}$ is likely to specify, for example, that the screen “should be 19 inches across”—whereby she presumably does *not* mean simply that any size under 19 inches is out of the question. One way of conceptualizing this problem is to view $\mathcal{U}$ as defining a fuzzy concept SUITABLE SCREEN SIZE which is clearly applicable at 19 inches and less applicable at some other values. Fuzzy logic provides a repertoire of membership functions for capturing such concepts (see, e.g., Cox, 1994, chap. 3).

The problem of determining the overall suitability of a product $\mathcal{X}$ for $\mathcal{U}$ can then be viewed as the problem of interpreting a fuzzy rule like the following one:

▷ IF $\mathcal{X}$ has a SUITABLE SCREEN SIZE
AND $\mathcal{X}$ has a SUITABLE PROCESSOR SPEED
AND ...
THEN $\mathcal{X}$ is a SUITABLE COMPUTER.

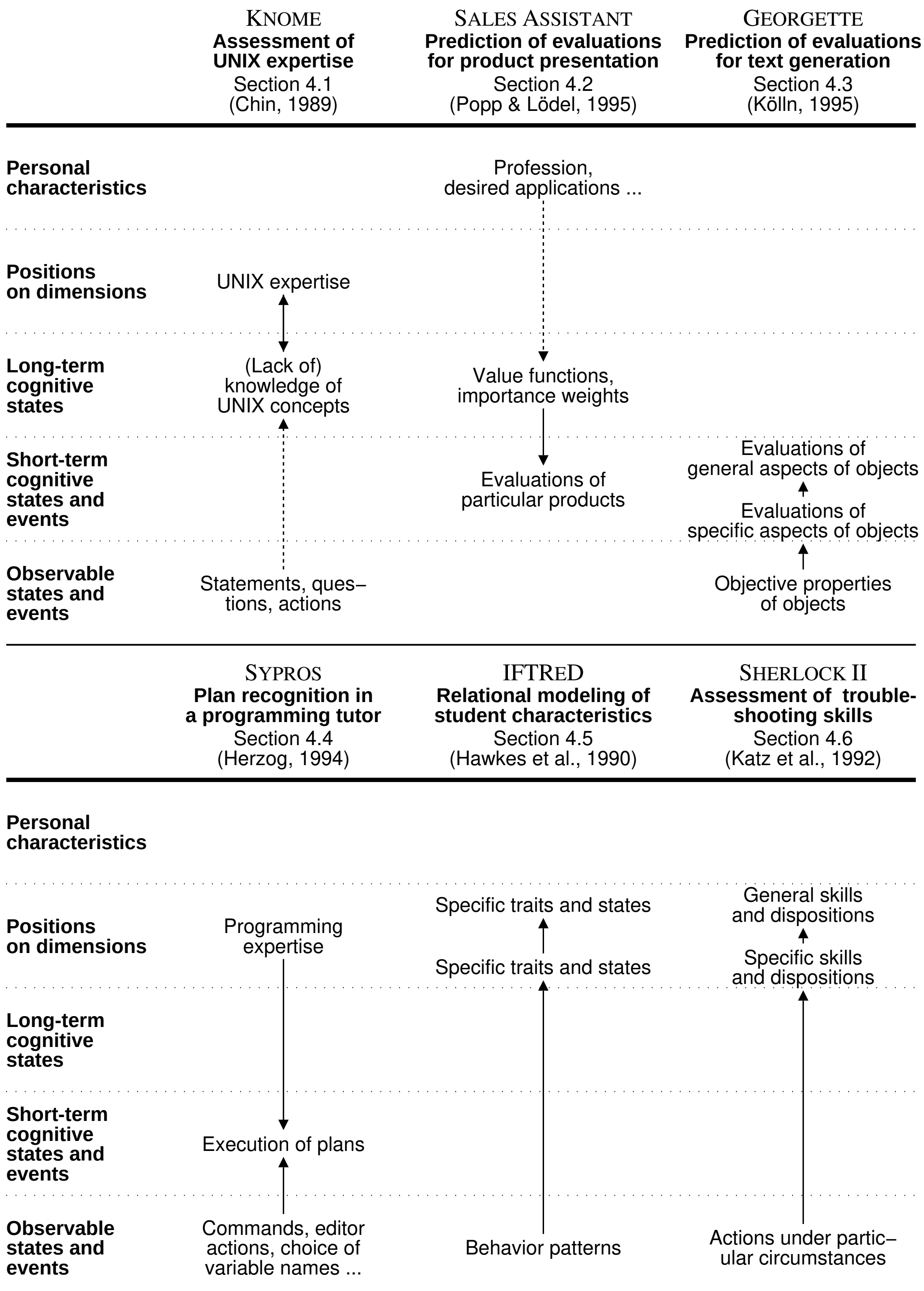

Figure 11. Systems that have used some concepts of FL for user or student modeling (cf. Figure 3 and Appendix A).

This is a type of formulation that both system designers and users are likely to understand and to accept, but its exact interpretation is not obvious. For example, it must be determined what the conjunction AND means in this context—in particular, the extent to which a single unsuitable attribute should disqualify a product. There is no universally valid answer, but fuzzy logic provides a library of functions for the interpretation of operators like this one (see, e.g., Cox, 1994, chap. 4).

In sum, the verbal formulations that people tend to prefer to use to express quantitative information in general have meanings which are vague and context-dependent (see Wallsten et al., 1993; Wallsten & Budescu, 1995). The repertoire of membership functions, operators, and other techniques that FL provides does not yield easy solutions to this problem, as the task of determining the appropriate representations remains—a task which may require considerable empirical testing and/or knowledge engineering (cf. Sections 5.1 and 5.3).

4.3. GEORGETTE: FUZZY RULES FOR USER SIMULATION

Kölln (1995) is developing the text generation system GEORGETTE, which, like the SALES ASSISTANT, uses fuzzy concepts to represent the user's assumed evaluation criteria. In the example domain, the objects to be evaluated are houses and apartments that are being offered for sale. As the purpose of this system is to generate real-estate advertisements noninteractively, the motivation for using FL is not to enable the system to handle vague user inputs. Rather, Kölln argues that fuzzy concepts allow the reader's evaluation processes to be modeled in a realistic fashion. A realistic model is required here: The generator consults the model not only to predict how $\mathcal{U}$ will evaluate an object as a whole but also to determine the relevance of particular attributes and the evaluation of the object with respect to various evaluation subdimensions such as “Location”.

In both GEORGETTE and the SALES ASSISTANT, the FL-based user model is not just an instrument for making predictions about the user; it can also be viewed as a (more or less simplified) cognitive simulation model of the user's evaluation processes (cf. the use of BNs by Horvitz & Barry, 1995, described in Section 2.14). When this is the intention, arguments concerning the greater human-likeness of fuzzy systems (cf. Section 5.5) acquire special weight.

Another characteristic aspect of Kölln's system is the way it represents complex quantitative relationships with combinations of rules formulated in terms of fuzzy concepts. Compared with the SALES ASSISTANT, GEORGETTE uses a much larger number of fuzzy concepts and rules to characterize the evaluation criteria that the system ascribes to members of a particular user group. In this way, the system can represent a vast number of possible relationships between a property of an object and its evaluation by a user; some of these would be hard to capture within a traditional framework such as Multi-Attribute Utility Theory (cf. Section 2.12 and von Winterfeldt & Edwards, 1986). This prospect can be attractive if the system's knowledge engineer thinks that it is both *necessary* to capture a functional relationship in such a differentiated fashion and *feasible* to do so by querying relevant persons and/or fine-tuning the rules on the basis of experience.

4.4. SYPROS: STUDENT MODELING WITH A FUZZY EXPERT SYSTEM

Whereas each of the FL systems reviewed so far exhibits some typical fuzzy features, more of these features are combined within the diagnostic component of SYPROS (Herzog, 1994). SYPROS is an intelligent tutoring system for parallel programming in a specially designed programming language. The key task of the user modeling component is the following one: Given that $\mathcal{U}$ has written a particular command within his program, determine which of several possible plans $P_1 \ldots P_N$ the command was intended to help realize.

The system has available 10 types of evidence on which to base such a decision, including the following:

- the relative difficulty of the goals that correspond to the plans $P_1 \ldots P_N$;
- the expertise level of $\mathcal{U}$;
- the nature of the explanations that have been offered so far by the tutoring system; and
- the extent to which a particular plan P_i would help to achieve a higher-level goal that $\mathcal{U}$ is known to be pursuing.

It would be possible, though by no means straightforward, to try to use BNs or DST to integrate these diverse types of evidence. Instead, SYPROS uses a fuzzy expert system with rules that can be expressed roughly as in the following example:

▷ IF P_i has the RIGHT HISTORY
AND the goal associated with P_i has the RIGHT DEGREE OF DIFFICULTY
OR P_i is NOT associated with the WRONG TYPICAL MISTAKES
THEN P_i is the CORRECT HYPOTHESIS.

As in the rules listed in Sections 4.1.3 and 4.2, each of the concepts and operators printed in upper-case letters is fuzzy. The procedure for determining the fuzzy membership value for a given P_i and a fuzzy concept $\mathcal{C}$ is as follows: All N candidate plans are rank-ordered with respect to the extent to which the evidence relevant to $\mathcal{C}$ speaks for them. The highest-ranking plan always receives a fuzzy membership value of 1.0; for the other plans, the rank order (essentially an integer between 2 and N) is mapped onto the half-open interval [0,1) with a monotonic function that takes various forms, depending on the judged importance of $\mathcal{C}$.

As with the evaluation systems discussed in the two previous subsections, the logical operators AND, OR, and NOT are given definitions taken from the repertoire of fuzzy operators. For OR and NOT, the most common definitions are used; for AND, it is not the minimum of the membership values that is used but rather the arithmetic mean. The justification is that a single largely unfulfilled condition (e.g., the fact that a plan ranks low with respect to having the RIGHT HISTORY) should not completely block the application of a rule.

In sum, a set of rules like this takes into account a variety of types of information that appear relevant, and it combines them in ways which appear plausible to the designer and/or knowledge engineer. The rules are not based on an explicit model of the relationships among the variables that they take into account, and many of the details of the rules are hard to justify theoretically. These facts have consequences for the way in which such a system is tested and adapted, as will be discussed in Section 5.3.

4.5. IFTRED: REPLACING FUZZY RULES WITH LINEAR EQUATIONS

For completeness, two approaches should be mentioned to which the term *fuzzy* has been applied but which in fact exhibit relatively few characteristics of fuzzy systems.

Hawkes et al. (1990) present an approach in which a student model is maintained in a relational database which stores hypotheses mainly about patterns in $\mathcal{U}$'s observable behavior and the positions of $\mathcal{U}$ on a variety of specific dimensions. Some of the variables are fuzzy in the sense that they characterize a student with terms like VERY LOW instead of with Boolean predicates or numbers. But these terms are internally represented and processed as integers, and no use is reported of fuzzy membership functions or related concepts.

The system allows the designer to formulate equations that permit inferences on the basis of the student model, such as the following equation (p. 423):

$$Y = .3X_1 + .2X_2 + .1X_3 + .3X_4 + .1X_5.$$

Here, Y represents $\mathcal{U}$'s current motivational state, which the system might want to predict, for instance, in order to judge the appropriateness of presenting a given type of problem. X_1 represents $\mathcal{U}$'s general trait motivation for mathematics, and X_2 through X_5 summarize several aspects of $\mathcal{U}$'s behavior in the current session which seem likely to reflect (or perhaps to influence) his current motivational state. The system evaluates an equation like this simply by substituting in the integer values for X_1 through X_N and rounding off the result to obtain an integer value for Y.

These equations are like the fuzzy rules of Herzog (1994) in that each equation expresses the way in which various pieces of evidence support a conclusion according to the judgment of the domain expert who formulated them. They are more constrained, however, in that they do not allow evidence to be combined through the use of various logical operators, which could in turn be given various definitions. It is not clear how successfully equations of this form can in fact predict student behavior and update a student model.

4.6. SHERLOCK II: INVESTIGATING THE UTILITY OF ALTERNATIVE PROPAGATION TECHNIQUES

Katz et al. (1992)[15] present a student modeling component for a troubleshooting tutoring system that uses $\mathcal{U}$'s actions as evidence concerning $\mathcal{U}$'s positions on various skill dimensions. An example of a rule that links observable behavior to a specific dimension is the following rule:

▷ IF $\mathcal{U}$ uses the handheld meter to measure DC voltage when the voltage is in fact AC,
THEN downgrade moderately the assessment of $\mathcal{U}$'S ABILITY TO USE THE HANDHELD METER.

Rules like this could be represented and processed in a variety of ways with the techniques described elsewhere in this article. Katz et al. argue that problems of knowledge engineering and computational complexity might make the use of techniques such as BNs impractical (cf. Sections 5.1 and 5.4 below). They therefore explore a novel technique in order to determine whether it represents a useful alternative: Each variable such as $\mathcal{U}$'S ABILITY TO USE THE HANDHELD METER has five possible values, ranging from NO KNOWLEDGE to FULLY DEVELOPED KNOWLEDGE. For each variable, $\mathcal{S}$ maintains a probability distribution like the ones used in BNs. These probabilities are not updated according to any rules of probability theory, however. For example, the formula for downgrading the assessment of a variable prescribes that some proportion of the probability assigned to each value of the variable be transferred to the value one step below. In this respect, the method is similar to Chin's (1989) nonstandard method for updating probabilities. A consequence of the use of tailor-made updating rules is that the updating process itself can be fine-tuned so that particular effects are obtained. For example, the updating procedure provides explicitly for slower upgrading when $\mathcal{U}$ is judged to be a near-expert, because the designers consider it to be especially undesirable for a student to be classified as an expert erroneously.

In addition to assessing specific abilities, SHERLOCK II assesses more global ones, relating them to the specific abilities via linear equations, as in IFTRED. For example, the ability to use test equipment (Y) is related to the abilities to use the handheld meter (X_1), the digital multimeter (X_2), and the oscilloscope (X_3) as follows:

$$Y = .2X_1 + .2X_2 + .6X_3.$$

[15] A slightly shorter account of this research is given by Katz and Lesgold (1992).

Applying an equation like this one is less straightforward than applying the corresponding equations in IFTRED, because the system's beliefs about the four variables are represented by probability distributions rather than by specific numbers. Here again, instead of invoking the principles of probability theory, the authors introduce a simpler scheme: The linear equation is applied directly to the probabilities associated with the various possible values of the variables. Assume, for example, the following probability distributions:

- X_1 and X_2: (1 0 0 0 0); that is, definitely NO KNOWLEDGE;
- X_3 : (0 0 0 0 1); that is, definitely FULLY DEVELOPED KNOWLEDGE.

Then $\mathcal{U}$ ought to be rated about average overall with respect to Y, his ability to use test equipment. But the procedure applied yields a probability distribution for this variable of (.4 0 0 0 .6). That is, $\mathcal{U}$ may have NO KNOWLEDGE of test equipment usage, or he may have FULLY DEVELOPED KNOWLEDGE; but he definitely does not have a knowledge level between these extremes. Anomalies like this illustrate the perils of developing new methods of handling evidence as alternatives to methods that have already benefited from extensive thought and scrutiny. Katz et al. acknowledge that their method is less reliable than standard uncertainty management techniques. In fact, they conclude their article with a discussion of reasons why BNs might be better suited for a system like SHERLOCK II.

4.7. CONCLUDING REMARKS ON FUZZY LOGIC

The systems reviewed illustrate two general user modeling situations in which the use of FL techniques should be considered:

1. Reasoning is involved that can be described and explained naturally in terms of imprecise concepts, operators, and rules, as opposed to mathematical principles or rules involving precise concepts.
   This reasoning may be that of the user, whose inferences or evaluations are being anticipated; or it may be that of an expert whose knowledge constitutes the basis for the system's reasoning.
2. Imprecise verbal input from the user has to be processed.
   The membership functions of FL are in general well suited to the representation of such input, even if the subsequent processing does not use FL techniques.

With regard to the first situation, note that even in cases where the system's inferences can be realized straightforwardly with a precise model, it will also be possible to realize them in a more approximate way with FL techniques. But some of the examples considered above in Sections 4.1 and 4.6 have raised doubts as to whether this is an appropriate way to use FL techniques. The arguments in favor of doing so would mainly concern usability issues such as those examined in the next section.

5. The Usability of Numerical Uncertainty Management for User Modeling

Some of the most frequently advanced arguments against the use of numerical uncertainty management techniques for user modeling do not concern the question of whether they could, in principle, yield accurate and/or useful results. Rather, the arguments address the question of the usability of these techniques: whether the techniques can be used satisfactorily in the conditions in which research and application typically take place. More specifically, a designer who is considering such techniques for the user modeling system $\mathcal{S}$ may wonder about the following questions:

1. Where will the numbers needed by $\mathcal{S}$ come from?
2. How much effort will it require to implement $\mathcal{S}$?
3. To what extent will $\mathcal{S}$ have to be improved through trial and error?
4. Will $\mathcal{S}$'s inference methods be efficient enough to permit acceptably fast system responses?

5. To what extent will the inferences made by $\mathcal{S}$ be similar to those made by people faced with analogous tasks?
6. To what extent will it be possible to explain $\mathcal{S}$'s inferences to users and other persons who want to understand them?
7. To what extent will it be possible to justify the conclusions drawn by $\mathcal{S}$ if they are called into question?
8. How effectively will it be possible to communicate the lessons learned in the design and testing of $\mathcal{S}$ to other designers of user modeling systems?

These questions will be discussed in turn in the following subsections. It would be convenient if we could give each of the three uncertainty management paradigms a rating with respect to each of these questions—and maybe even a rating for overall usability in user modeling systems. Unfortunately, the answers to these questions depend in part on the particular way in which a technique is employed. So the aim of the following subsections will be more modest: to mention the major factors that should be taken into account during an attempt to answer each of these questions with regard to a specific system.

5.1. Knowledge Engineering Requirements

The question of where the numbers come from appears to be the most difficult one of all. In most of the systems described in this overview, most or all of the required numbers were apparently entered by the designers on the basis of intuitive judgment. Even in cases where systematically collected empirical data were used, the designers themselves warn against optimistic assumptions about the accuracy of the numbers (see, e.g., de Rosis et al., 1992, pp. 386–387; Desmarais et al., 1995, section 4). This widespread lack of a solid quantitative knowledge base is in keeping with the fact that these systems are research prototypes. But this research does not in most cases indicate how accurate numbers could be obtained in practice.

The development of FL was motivated largely by a desire to make the arbitrary specification of precise numbers unnecessary. In particular, it is sometimes noted that it is easier to elicit fuzzy rules than exact numbers from domain experts (see, e.g., Section 4.2 and Vadiee & Jamshidi, 1994, p. 37). Still, FL-based systems do require the specification of numbers on other levels. As the systems reviewed in Section 4 illustrate, at some point even the fuzziest inputs have to be translated into some sort of internal numerical representation: A vague concept has to have a membership function (if it is not just to be mapped onto an integer), and the various pieces of input data for a complex rule have to be combined according to some operators. These internal representations can in principle be just as unrealistic as arbitrarily chosen conditional probabilities or belief mass assignments. So the problem remains of how to choose the right ones. This problem is fully recognized within FL, and experimentation with a variety of alternative numerical representations is often viewed as a central part of the process of developing an FL-based system (cf., e.g., the application-dependence of the fuzzy modeling in the systems described by Popp & Lödel, 1995).

The development of improved techniques for acquiring the necessary numbers is a widely pursued research goal (see, e.g., Druzdzel et al., 1995, for examples of efforts in connection with BNs). One general approach is to improve methods of eliciting the necessary judgments from experts. For example, Druzdzel and van der Gaag (1995) present a general method for deriving exact probabilities from a variety of types of input provided by experts, including purely qualitative probability assessments. A complementary strategy is to develop appropriate machine learning techniques (see, e.g., Section 2.7 with regard to BNs and Kosko, 1992, with regard to FL-based systems).

In the near future, however, precise numbers that are output by a user modeling system should not be taken too literally, unless the system's accuracy has somehow been convincingly demonstrated. In other words, for many systems precise and accurate results do not represent the chief benefit of employing numerical uncertainty management techniques. There are several advantages to the use of well-understood numerical techniques which can be exploited even if the system's quantitative knowledge base leaves much room for improvement (cf. Pearl, 1988, pp. 20–21):

1. The attempt to specify all of the necessary quantities forces the system developer to deal with many complications that would otherwise not even be noticed; and a system that embodies at least an educated guess about such a complication is likely to be preferable to one that doesn't take it into account at all.
2. It is convenient to know that any unexpected or incorrect behavior of the system cannot be due simply to poorly understood aspects of the basic inference mechanism; rather, there must be something wrong with the assumptions that were entered into the model.
3. Although the conclusions drawn by the system may be inaccurate, they will not be inconsistent or bizarre. They will be conclusions that would be valid in some plausible world, namely one in which all of the assumptions made in the system were accurate.
4. Even inaccurate numbers in the system can serve as place-holders for more accurate numbers that can be acquired in the future, in one or more of the ways mentioned above. Some alternative approaches to uncertainty management do not support this sort of incremental improvement.

5.2. Programming Effort

The effort required to program an implementation is likewise often cited as a reason for avoiding the use of the more sophisticated numerical techniques covered in this overview. Apart from its lack of scientific weight, this argument is rapidly becoming anachronistic as the amount of relevant public-domain and commercial software increases. For example, Mislevy and Gitomer (1995) use a commercially available BN shell, and the *Fuzzy Systems Handbook* by Cox (1994) comes with a diskful of numerical routines programmed in C++.[16]

5.3. Empirical Model Adjustment

Section 5.1 noted the difficulty of specifying a valid model on the basis of empirical data, expert assessments, or theoretical considerations. An alternative approach is to tune the parameters of a user modeling system on the basis of feedback from system performance. This general approach has long been familiar in the field of rule-based expert systems. As was noted in Section 5.1, this sort of tuning is especially characteristic of the FL paradigm. The FL systems described in Section 4 in fact show more possibilities and examples than do those of the other two paradigms.

As a reviewer of one of the manuscripts for this special issue has emphasized, revising a model on the basis of empirical feedback should not be viewed as necessarily just a matter of adjusting parameters by trial and error until the results are more or less satisfactory. Instead, it is desirable, where possible, that the initial design of the model should be based on explicit assumptions about the exact meanings of the variables involved and about the nature of their relationships. Unexpected behavior of the system may then suggest revisions of these assumptions rather than just adjustment of specific parameters. An advantage of this approach is that experience with the system enhances general understanding of the user modeling problem being investigated instead of merely improving the performance of a specific

[16] At the time of this writing, one source of information about software for BNs and DST is the World-Wide Web page `http://bayes.stat.washington.edu/almond/belief.html` maintained by Russell Almond.

system in a particular domain. This approach appears to come more naturally with BNs than with DST or especially FL: BNs tend more strongly to force the designer to make explicit, falsifiable claims about relationships among variables in the world, instead of specifying explicitly how particular types of evidence are to be processed. By the same token, a BN designer is perhaps the one who is most likely to be forced to the conclusion that his or her entire conceptualization of a problem is inappropriate.

5.4. COMPUTATIONAL COMPLEXITY

One reason that is often given for not using BNs or DST for user modeling is the computational complexity of these techniques. This problem is mentioned less frequently in connection with FL. In fact, proponents of FL often point to the fast execution times of fuzzy systems, in particular fuzzy controllers. This does not mean that problems of computational complexity never arise within FL, but it does suggest that they needn't be a primary concern for researchers who are considering the use of FL for user modeling.

For BNs and DST, however, it has been proven that the exact application of the inference techniques is in general NP-hard—as are some types of approximate application (see, e.g., Wellman & Liu, 1995, and Kreinovich et al., 1994, for recent discussions). Moreover, intractability is not only a theoretical possibility. It often plagues users of these formalisms in practice, though publications within the user modeling field by researchers who employ these techniques have seldom mentioned problems of this sort.[17] But before a researcher decides, because of computational complexity considerations, to refrain from using these techniques for a given application $\mathcal{A}$, he or she should consider the following questions:

1. Are the problems within $\mathcal{A}$ small?
   Even computationally complex methods can be used if the problems are small enough. And in user modeling systems, there may be specific places where a differentiated and reliable treatment of uncertainty is desirable even if most of the system uses other inference techniques (cf., e.g., the system of Carberry, 1990, discussed in Section 3.2).
2. Do the problems in $\mathcal{A}$ have a structure that happens to be favorable with regard to the uncertainty management technique being considered—or can they be reformulated in such a way that they have such a structure?
   Even large problems can be handled without difficulty if they represent special cases for which the inference techniques can work efficiently. In BNs that are *singly connected* (i.e., that have at most one undirected path between any two nodes), propagation time is linear or better in the size of the network (see, e.g., Delcher et al., 1995). It also helps if a typical node has a small number of parents and if a typical variable has a small number of possible values (Wellman & Liu, 1995). With DST, it is helpful if only a small number of possible subsets of each set of hypotheses has to be taken into account (cf. Sections 3.1, 3.3, and 3.4). When DST is used to realize a belief network (as was discussed in Section 3.1.2), some of the same considerations apply as with BNs.
3. Are approximative techniques applicable within $\mathcal{A}$?
   A good deal of research is being devoted to ways of making the use of BNs and DST computationally more feasible. Much of this research investigates approximative techniques. The use of such techniques makes good sense in an area like user modeling, where the numbers that are entered into the system will themselves often be only approximate. But even approximative techniques may be effective only under particular conditions. For example, the use of a restricted set of *infinitesimal*

[17] On the other hand, about half of Eugene Charniak's invited talk at the Fourteenth International Joint Conference on Artificial Intelligence, a talk which dealt with the research program touched on in Section 2.10, was devoted to a report on his research group's efforts to find tolerably fast ways of evaluating their especially fine-grained BNs.

probabilities in a BN appears to work well only when the prior probability of the condition to be diagnosed is very small (Henrion et al., 1994). And a Monte Carlo technique for DST proposed by Kreinovich et al. (1994) is inapplicable if Dempster's rule of combination is used; the designer must be willing to accept an alternative rule for integrating pieces of evidence.

If the answers to all three of these questions are negative, the natural response might be to adopt some technique $\mathcal{X}$ for managing (or ignoring) uncertainty that poses fewer computational problems but that has a less sound theoretical foundation than BNs or DST. But before taking this course, the researcher should consider the following (interrelated) questions:

1. What class of problems can be solved satisfactorily with $\mathcal{X}$?
   It may turn out that method $\mathcal{X}$ works adequately only on a limited subset of the possible problems; and that for this subset, the use of BNs or DST would be feasible according to the criteria listed above.
2. How accurate are the solutions yielded by $\mathcal{X}$?
   It may turn out that these solutions have only a limited degree of accuracy; and that the same degree of accuracy can be obtained by feasible approximative versions of BNs or DST.

When these two questions are considered, it may turn out that BNs and DST are the worst possible techniques, until you consider the alternatives.

5.5. Human-Likeness

The human-likeness of the inferences performed with an uncertainty management technique is especially important if the technique is being used to simulate the user's reasoning rather than to manage uncertainty about it (cf. Sections 2.14 and 4.3).

As was noted several times in Section 4, FL can claim at least a certain degree of human-likeness because of the way in which it captures human reasoning with vague concepts. On the other hand, FL was not developed for the purpose of cognitive simulation, and it cannot be taken for granted that an FL treatment of a given problem corresponds to the way in which people would deal with it. Chandrasekaran (1994) and Freksa (1994) offer balanced discussions of the extent to which FL can—and should—faithfully mirror human reasoning.

At the other extreme, Bayesian inference has often been viewed as antithetical to human reasoning. One basis for this view has been the large collection of empirical results which document the errors that experimental subjects make when confronted with tasks that require Bayesian reasoning. But the gap between human and Bayesian inference is not quite as wide as is commonly believed. In particular, relatively recent studies have shown that people can actually perform well on basically the same experimental tasks under certain circumstances—for instance when the tasks are reformulated in terms of frequencies as opposed to subjective probabilities (see, e.g., the recent review by Ayton & Pascoe, 1995).

In sum, a judgment of the human-likeness of the inferences supported by a particular uncertainty management framework should take into account the nature and context of the particular inferences being made.

5.6. Explainability

With BNs and DST, the role of numerical calculations is more prominent than it is in FL-based systems. The designer of a user modeling system may therefore be concerned about the need to explain such calculations to users. But in fact, explaining BN- and DST-based inferences is not primarily a matter of explaining specific numerical calculations (cf. Shafer's remark to Pearl that "probability is not really about numbers; it is about the structure of reasoning"—Pearl, 1988, p. 15). For instance, when BNs are being explained, the essential goal is for the user to grasp the nature of the causal (or other) relationships between the variables represented in the network. The user should also be able to recognize certain typical patterns of reasoning, such as the pattern of *explaining away*: the idea that to the extent to which an effect has been explained adequately by one cause, it no longer constitutes evidence for the operation of another possible cause (cf. Pearl, 1988, p. 7).

A fair amount of research has been devoted to the development of techniques for explaining the reasoning of systems that are based on BNs and DST (see, e.g., Henrion & Druzdzel, 1991, and Xu & Smets, 1995, respectively). Comparable research on other normatively oriented reasoning techniques (see, e.g., Klein & Shortliffe, 1994) is also relevant on a more general level. The results suggest that, with ingenuity and careful attention to users' reasoning processes, researchers and designers can find ways to bridge the gap between formally rigorous methods and users' natural ways of thinking.

5.7. Justifiability

Because of the desirability of making user modeling procedures human-like and explainable, it is easy to forget that *justifiability* is also a desirable—and somewhat different—property. An extreme case is one in which a user model provides the basis for decisions about users that have important consequences for them (e.g., a recommendation of an expensive product as suitable for purchase by the user, or a judgment that a student is unfit to participate in a given course). If such a decision is called into question and has to be justified, it may be useful to be able to argue that the system employs the most reliable techniques currently available, even if these techniques are less intuitively natural than the alternatives (cf. Martin & VanLehn, 1995, p. 144).

5.8. Communicability of Research Advances

When the solution to a problem in user modeling is formulated in terms of a well-known uncertainty management paradigm, it is relatively easy for other researchers to understand and evaluate what has been accomplished and to adopt the solution in their own work. When a solution is formulated within some system-specific framework, other researchers often have to invest considerable effort simply to discover that the solution is essentially equivalent (or perhaps inferior) to one that has been proposed before.

6. Conclusion

This overview supports an integrative view of the role of numerical uncertainty management techniques in user and student modeling. We have seen that these techniques can be used to address a broad variety of problems in this area, often in conjunction with more familiar qualitative techniques. Researchers and designers therefore do not have to commit themselves to the use of a technique from a particular uncertainty management paradigm as the primary inference technique for their system. And there is no compelling reason why different uncertainty management paradigms cannot be applied within a single system.

Researchers who begin applying these techniques are likely to encounter problems that they have not previously had to deal with. But many of these problems do not concern technical drawbacks of the techniques in question but rather difficulties inherent in the enterprise of user and student modeling. These problems will at some point have to be dealt with head-on. Researchers who choose to do so will find that the paradigms covered here offer a large, varied, and rapidly expanding collection of conceptual and implementational tools.

Acknowledgements

Preparation of this article was supported by the German Science Foundation (DFG) in its Special Collaborative Research Program on Artificial Intelligence and Knowledge-Based Systems (SFB 314), project N1, PRACMA. Valuable suggestions on earlier versions were made by Matthias Kölln, Ralph Schäfer, Katherine Sinitsa, Susanne van Mulken, and the four anonymous reviewers. Matthias Bauer, Cristina Conati, Kurt VanLehn, and Peter van Beek allowed their recent research to be discussed in this overview even before they had finished writing it up for publication.

Appendix

A. A Framework for the Comparison of Systems

In Figures 3, 5, 8, and 11, each system is characterized in terms of a classification of variables into several *levels*. The levels will be explained here with reference to the example situation analyzed in Sections 2.1, 3.1, and 4.1.

Personal characteristics. These variables correspond to theory-independent, objectively verifiable facts about $\mathcal{U}$, such as her area of specialization and her amount of experience with a given system. If the example systems had this type of information about a UNIX user, they could treat it as evidence concerning $\mathcal{U}$'s UNIX expertise.

Positions on dimensions. Each variable on this level refers to the position of $\mathcal{U}$ on some general or specific dimension such as expertise in UNIX. Dimensions may also concern preferences rather than knowledge—such as $\mathcal{U}$'s tendency to make backup copies of files. A variable of this type corresponds to a theoretical construct: Whether the variable is meaningful depends on whether it plays a useful role in inferences about variables on other levels.

Long-term cognitive states. These states include the user's possession of knowledge, beliefs, and goals whose relevance extends beyond the performance of a particular task. Examples are variables that represent the propositions that (a) $\mathcal{U}$ believes that files named `README` are usually worth reading; and (b) $\mathcal{U}$ knows the meaning of the command `more` (which displays the contents of a file on the screen). The main difference from variables on the previous level, positions on dimensions, is that the content of long-term cognitive states is more specific. The distinction is not sharp, however. If, for example, knowing the meaning of `more` is considered to be a matter of degree, then the degree of understanding of this command can be treated as a very specific knowledgeability dimension which is analogous to the dimension of expertise in UNIX.

Short-term cognitive states and events. These are events and temporary states that arise in connection with the particular task being performed by $\mathcal{U}$. For example, variables might refer to the current status for $\mathcal{U}$ of the goal "Read the contents of `README`"; or to $\mathcal{U}$'s belief that `README` contains currently relevant information.

Observable states and events. These are states and events that a person or system could in principle observe with a negligible chance of error. The most important events are user actions—for example $\mathcal{U}$'s typing of the command `more README`. But this level also encompasses variables corresponding to observable causes and consequences of user behavior, for example whether the name of the file `README` is currently being displayed on the screen (cf. Section 2.8).

References

Ayton, P., & Pascoe, E. (1995). Bias in human judgement under uncertainty? *The Knowledge Engineering Review, 10*(1), 21–41.

Bauer, M. (1995). A Dempster-Shafer approach to modeling agent preferences for plan recognition. *User Modeling and User-Adapted Interaction*.

Bauer, M. (1996). Acquisition of user preferences for plan recognition. In S. Carberry & I. Zukerman (Eds.), *Proceedings of the Fifth International Conference on User Modeling* (pp. 105–112). Boston, MA: User Modeling, Inc.

Besnard, P., & Hanks, S. (Eds.) (1995). *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. San Francisco: Morgan Kaufmann.

Carberry, S. (1990). Incorporating default inferences into plan recognition. *Proceedings of the Eighth National Conference on Artificial Intelligence*, Boston, MA, 471–478.

Carbonaro, A., Maniezzo, V., Roccetti, M., & Salomoni, P. (1995). Modelling the student in Pitagora 2.0. *User Modeling and User-Adapted Interaction, 4,* 233–251.

Chandrasekaran, B. (1994). Broader issues at stake: A response to Elkan. *IEEE Expert, 9* (4), 10–13.

Charniak, E. (1991). Bayesian networks without tears. *AI Magazine, 12* (4), 50–63.

Charniak, E., & Goldman, R. (1991). A probabilistic model of plan recognition. *Proceedings of the Ninth National Conference on Artificial Intelligence*, Anaheim, CA, 160–165.

Charniak, E., & Goldman, R. P. (1993). A Bayesian model of plan recognition. *Artificial Intelligence, 64,* 53–79.

Chin, D. N. (1989). KNOME: Modeling what the user knows in UC. In A. Kobsa & W. Wahlster (Eds.), *User models in dialog systems* (pp. 74–107). Berlin: Springer.

Conati, C., & VanLehn, K. (1996). POLA: A student modeling framework for Probabilistic On-Line Assessment of problem solving performance. In S. Carberry & I. Zukerman (Eds.), *Proceedings of the Fifth International Conference on User Modeling* (pp. 75–82). Boston, MA: User Modeling, Inc.

Cox, E. (1994). *The fuzzy systems handbook: A practitioner's guide to building, using, and maintaining fuzzy systems*. Boston: AP Professional.

de Rosis, F., Pizzutilo, S., Russo, A., Berry, D. C., & Molina, F. J. N. (1992). Modeling the user knowledge by belief networks. *User Modeling and User-Adapted Interaction, 2,* 367–388.

Delcher, A. L., Grove, A., Kasif, S., & Pearl, J. (1995). Logarithmic-time updates and queries in probabilistic networks. In P. Besnard & S. Hanks (Eds.), *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 116–124). San Francisco: Morgan Kaufmann.

Dempster, A. P., & Kong, A. (1988). Uncertain evidence and artificial analysis. *Journal of Statistical Planning and Inference, 20,* 355–368. Reprinted in collection by Shafer and Pearl (1990).

Desmarais, M. C., Maluf, A., & Liu, J. (1995). User-expertise modeling with empirically derived probabilistic implication networks. *User Modeling and User-Adapted Interaction*.

Draney, K. L., Pirolli, P., & Wilson, M. (1995). A measurement model for a complex cognitive skill. In P. D. Nichols, S. F. Chipman, & R. L. Brennan (Eds.), *Cognitively diagnostic assessment* (pp. 103–125). Hillsdale, NJ: Erlbaum.

Druzdzel, M., van der Gaag, L., Henrion, M., & Jensen, F. (1995). *Building probabilistic networks: Where do the numbers come from?* Working notes of a workshop held in conjunction with the Fourteenth International Joint Conference on Artificial Intelligence, available from `http://www2.lis.pitt.edu/~uai/ijcai95.html`.

Druzdzel, M. J., & van der Gaag, L. C. (1995). Elicitation of probabilities for belief networks: Combining qualitative and quantitative information. In P. Besnard & S. Hanks (Eds.), *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 141–148). San Francisco: Morgan Kaufmann.

Dubois, D., Prade, H., & Yager, R. R. (Eds.) (1993). *Readings in fuzzy sets for intelligent systems*. San Mateo, CA: Morgan Kaufmann.

Elkan, C. (1994). The paradoxical success of fuzzy logic. *IEEE Expert, 9* (4), 3–8.

Falmagne, J.-C., Koppen, M., Villano, M., Doignon, J.-P., & Johannesen, L. (1990). Introduction to knowledge spaces: How to build, test, and search them. *Psychological Review, 97,* 201–224.

Fiske, S. T., & Taylor, S. E. (1991). *Social cognition* (2nd ed.). New York: McGraw-Hill.

Forbes, J., Huang, T., Kanazawa, K., & Russell, S. (1995). The BATmobile: Towards a Bayesian Automated Taxi. In C. S. Mellish (Ed.), *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 1878–1885). San Mateo, CA: Morgan Kaufmann.

Freksa, C. (1994). Fuzzy logic: An interface between logic and human reasoning. *IEEE Expert, 9* (4), 20–21.

Gordon, J., & Shortliffe, E. H. (1984). The Dempster-Shafer theory of evidence. In B. G. Buchanan & E. H. Shortliffe (Eds.), *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project*. Reading: MA: Addison-Wesley.

Hambleton, R. K., & Swaminathan, H. (1985). *Item response theory: Principles and applications*. Boston: Kluwer-Nijhoff.

Hawkes, L. W., Derry, S. J., & Rundensteiner, E. A. (1990). Individualized tutoring using an intelligent fuzzy temporal relational database. *International Journal of Man-Machine Studies, 33,* 409–429.

Heckerman, D., Geiger, D., & Chickering, D. M. (1994). Learning Bayesian networks: The combination of knowledge and statistical data. In R. Lopez de Mantaras & D. Poole (Eds.), *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence* (pp. 293–301). San Francisco: Morgan Kaufmann.

Henrion, M., Breese, J. S., & Horvitz, E. J. (1991). Decision analysis and expert systems. *AI Magazine, 12* (4), 64–91.

Henrion, M., & Druzdzel, M. J. (1991). Qualitative propagation and scenario-based schemes for explaining probabilistic reasoning. In P. P. Bonissone, M. Henrion, L. N. Kanal, & J. F. Lemmer (Eds.), *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence* (pp. 17–32). Amsterdam: Elsevier.

Henrion, M., Provan, G., Favero, B. D., & Sanders, G. (1994). An experimental comparison of numerical and qualitative probabilistic reasoning. In R. Lopez de Mantaras & D. Poole (Eds.), *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence* (pp. 319–326). San Francisco: Morgan Kaufmann.

Herzog, C. (1994). Fuzzy-Techniken für das Verstehen von Studentenlösungen in intelligenten Lehrsystemen [Fuzzy techniques for understanding student solutions in intelligent tutoring systems]. In R. Gunzenhäuser, C. Möbus, & D. Rösner (Eds.), *Beiträge zum 7. Arbeitstreffen der GI-Fachgruppe 1.1.5/7.0.1, "Intelligente Lehr-/Lernsysteme" [Papers for the Seventh Meeting of GI Section 1.1.5/7.0.1, "Intelligent Tutoring Systems"].* Ulm, Germany: Research Institute for Application-Oriented Knowledge Processing (FAW).

Horvitz, E., & Barry, M. (1995). Display of information for time-critical decision making. In P. Besnard & S. Hanks (Eds.), *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 296–314). San Francisco: Morgan Kaufmann. Available from `http://www.research.microsoft.com/research/dtg/horvitz/`.

Huber, M. J., Durfee, E. H., & Wellman, M. P. (1994). The automated mapping of plans for plan recognition. In R. Lopez de Mantaras & D. Poole (Eds.), *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence* (pp. 344–351). San Francisco: Morgan Kaufmann.

Hustadt, U., & Nonnengart, A. (1993). Modalities in knowledge representation. In *Proceedings of the Sixth Australian Joint Conference on Artificial Intelligence* (pp. 249–254). Singapore: World Scientific Publishing.

Jameson, A. (1990). *Knowing what others know: Studies in intuitive psychometrics*. Doctoral dissertation, University of Amsterdam, The Netherlands.

Jameson, A. (1992). Generalizing the double-stereotype approach: A psychological perspective. In E. André, R. Cohen, W. Graf, B. Kass, C. Paris, & W. Wahlster (Eds.), *Proceedings of the Third International Conference on User Modeling* (pp. 69–83). Saarbrücken, Germany: German Research Center for Artificial Intelligence (DFKI).

Jameson, A. (1995). Logic is not enough: Why reasoning about another person's beliefs is reasoning under uncertainty. In A. Laux & H. Wansing (Eds.), *Knowledge and belief in philosophy and artificial intelligence* (pp. 199–229). Berlin: Akademie Verlag. Available from `http://zaphod.cs.uni-sb.de/KI/`.

Jameson, A., Schäfer, R., Simons, J., & Weis, T. (1995). Adaptive provision of evaluation-oriented information: Tasks and techniques. In C. S. Mellish (Ed.), *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 1886–1893). San Mateo, CA: Morgan Kaufmann. Available from `http://zaphod.cs.uni-sb.de/KI/`.

Kambouri, M., Koppen, M., Villano, M., & Falmagne, J.-C. (1994). Knowledge assessment: Tapping human expertise by the QUERY routine. *International Journal of Human-Computer Studies, 40,* 119–151.

Katz, S., & Lesgold, A. (1992). Approaches to student modeling in the Sherlock tutors. In E. André, R. Cohen, W. Graf, B. Kass, C. Paris, & W. Wahlster (Eds.), *Proceedings of the Third International Conference on User Modeling* (pp. 205–230). Saarbrücken, Germany: German Research Center for Artificial Intelligence (DFKI).

Katz, S., Lesgold, A., Eggan, G., & Gordin, M. (1992). Modeling the student in Sherlock II. *Journal of Artificial Intelligence in Education, 3,* 495–518.

Kipper, B., & Jameson, A. (1994). Semantics and pragmatics of vague probability expressions. *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, Atlanta, 496–501. Available from `http://zaphod.cs.uni-sb.de/KI/`.

Klein, D. A., & Shortliffe, E. H. (1994). A framework for explaining decision-theoretic advice. *Artificial Intelligence, 67,* 201–243.

Kölln, M. E. (1995). Employing user attitudes in text planning. *Proceedings of the Fifth European Workshop on Natural Language Generation*, Leiden, The Netherlands, 163–179.

Kosko, B. (1992). *Neural networks and fuzzy systems: A dynamical systems approach to machine intelligence*. London: Prentice-Hall.

Kreinovich, V. Y., Bernat, A., Borrett, W., Mariscal, Y., & Villa, E. (1994). Monte-Carlo methods make Dempster-

Shafer formalism feasible. In R. R. Yager, J. Kacprzyk, & M. Fedrizzi (Eds.), *Advances in the Dempster-Shafer theory of evidence* (pp. 175–191). New York: Wiley.
Kruse, R., Schwecke, E., & Heinsohn, J. (1991). *Uncertainty and vagueness in knowledge based systems: Numerical methods*. Berlin: Springer.
Lamata, M. T., & Moral, S. (1994). Calculus with linguistic probabilities and beliefs. In R. R. Yager, J. Kacprzyk, & M. Fedrizzi (Eds.), *Advances in the Dempster-Shafer theory of evidence* (pp. 133–152). New York: Wiley.
Lopez de Mantaras, R., & Poole, D. (Eds.) (1994). *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*. San Francisco: Morgan Kaufmann.
Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM, 37* (7), 30–40.
Martin, J. D., & VanLehn, K. (1993). OLAE: Progress toward a multi-activity, Bayesian student modeler. In P. Brna, S. Ohlsson, & H. Pain (Eds.), *Artificial intelligence in education: Proceedings of AI-ED 93* (pp. 410–417). Charlottesville, VA: Association for the Advancement of Computing in Education.
Martin, J. D., & VanLehn, K. (1995). A Bayesian approach to cognitive assessment. In P. D. Nichols, S. F. Chipman, & R. L. Brennan (Eds.), *Cognitively diagnostic assessment* (pp. 141–165). Hillsdale, NJ: Erlbaum.
Mislevy, R. J. (1994). Evidence and inference in educational assessment. *Psychometrika, 59,* 439–483.
Mislevy, R. J., & Gitomer, D. H. (1995). The role of probability-based inference in an intelligent tutoring system. *User Modeling and User-Adapted Interaction*.
Neapolitan, R. E. (1990). *Probabilistic reasoning in expert systems: Theory and algorithms*. New York: Wiley.
Nisbett, R., & Ross, L. (1980). *Human inference: Strategies and shortcomings of social judgment*. Englewood Cliffs, NJ: Prentice-Hall.
Orwant, J. (1995). Heterogeneous learning in the Doppelgänger user modeling system. *User Modeling and User-Adapted Interaction, 4* (2), 107–130.
Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Mateo, CA: Morgan Kaufmann.
Petrushin, V. A., & Sinitsa, K. M. (1993). Using probabilistic reasoning techniques for learner modeling. In P. Brna, S. Ohlsson, & H. Pain (Eds.), *Artificial intelligence in education: Proceedings of AI-ED 93* (pp. 418–425). Charlottesville, VA: Association for the Advancement of Computing in Education.
Petrushin, V. A., Sinitsa, K. M., & Zherdienko, V. (1995). Probabilistic approach to adaptive students' knowledge assessment: Methodology and experiment. In J. Greer (Ed.), *Artificial intelligence in education: Proceedings of AI-ED 95* (pp. 51–58). Charlottesville VA: Association for the Advancement of Computing in Education.
Popp, H., & Lödel, D. (1995). Fuzzy techniques and user modeling in sales assistants. *User Modeling and User-Adapted Interaction*. In this special issue.
Pynadath, D. V., & Wellman, M. P. (1995). Accounting for context in plan recognition, with application to traffic monitoring. In P. Besnard & S. Hanks (Eds.), *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 472–481). San Francisco: Morgan Kaufmann.
Quinlan, J. R. (1983). Learning efficient classification procedures and their application to chess end games. In R. Michalski, J. Carbonell, & T. Mitchell (Eds.), *Machine learning: An artificial intelligence approach* (pp. 463–482). Berlin: Springer.
Raskutti, B., & Zukerman, I. (1991). Handling uncertainty during plan recognition in task-oriented consultation systems. In B. D. D'Ambrosio, P. Smets, & P. P. Bonissone (Eds.), *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence* (pp. 308–315). San Mateo, CA: Morgan Kaufmann.
Roccetti, M., & Salomoni, P. (1995). *Using Bayesian belief networks for the automated assessment of students' knowledge of geometry solving procedures*. Manuscript submitted for publication.
Russell, S., Binder, J., Koller, D., & Kanazawa, K. (1995). Local learning in probabilistic networks with hidden variables. In C. S. Mellish (Ed.), *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 1146–1152). San Mateo, CA: Morgan Kaufmann.
Russell, S. J., & Norvig, P. (1995). *Artificial intelligence: A modern approach*. Englewood Cliffs, NJ: Prentice-Hall.
Schäfer, R. (1994). Multidimensional probabilistic assessment of interest and knowledge in a noncooperative dialog situation. In C. G. Thomas (Ed.), *ABIS 94 — GI Workshop: Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen [Adaptivity and user modeling in interactive software systems]* (pp. 46–62). Sankt Augustin, Germany: Gesellschaft für Mathematik und Datenverarbeitung.
Shafer, G., & Pearl, J. (Eds.) (1990). *Readings in uncertain reasoning*. San Mateo, CA: Morgan Kaufmann.
Shafer, G., & Tversky, A. (1985). Languages and designs for probability judgment. *Cognitive Science, 9,* 177–210.
Shenoy, P. P. (1994). Using Dempster-Shafer's belief function theory in expert systems. In R. R. Yager,

J. Kacprzyk, & M. Fedrizzi (Eds.), *Advances in the Dempster-Shafer theory of evidence* (pp. 395–414). New York: Wiley.

Sheth, B., & Maes, P. (1993). Evolving agents for personalized information filtering. *Proceedings of the Ninth Conference on Artificial Intelligence for Applications, CAIA-93*, Orlando, FL, 345–352.

Sime, J.-A. (1993). Modelling a learner's multiple models with Bayesian belief networks. In P. Brna, S. Ohlsson, & H. Pain (Eds.), *Artificial intelligence in education: Proceedings of AI-ED 93* (pp. 426–432). Charlottesville, VA: Association for the Advancement of Computing in Education.

Tokuda, N., & Fukuda, A. (1993). A probabilistic inference scheme for hierarchical buggy models. *International Journal of Man-Machine Studies, 38,* 857–872.

Vadiee, N., & Jamshidi, M. (1994). The promising future of fuzzy logic. *IEEE Expert, 9* (4), 36–38.

van Beek, P. (1996). An investigation of probabilistic interpretations of heuristics in plan recognition. In S. Carberry & I. Zukerman (Eds.), *Proceedings of the Fifth International Conference on User Modeling* (pp. 113–120). Boston, MA: User Modeling, Inc.

van Mulken, S. (1996). Reasoning about the user's decoding of presentations in an intelligent multimedia presentation system. In S. Carberry & I. Zukerman (Eds.), *Proceedings of the Fifth International Conference on User Modeling* (pp. 67–74). Boston, MA: User Modeling, Inc. Available from `http://www.dfki.uni-sb.de/~mulken`.

Villano, M. (1992). Probabilistic student models: Bayesian Belief Networks and Knowledge Space Theory. In C. Frasson, G. Gauthier, & G. I. McCalla (Eds.), *Proceedings of the Second International Conference on Intelligent Tutoring Systems* (pp. 491–498). Berlin: Springer.

von Winterfeldt, D., & Edwards, W. (1986). *Decision analysis and behavioral research*. Cambridge, England: Cambridge University Press.

Wahlster, W., André, E., Finkler, W., Profitlich, H.-J., & Rist, T. (1993). Plan-based integration of natural language and graphics generation. *Artificial Intelligence, 63,* 387–427.

Wainer, H. (Ed.) (1990). *Computerized adaptive testing: A primer*. Hillsdale, NJ: Erlbaum.

Wallsten, T. S., & Budescu, D. V. (1995). A review of human linguistic probability processing: General principles and empirical evidence. *The Knowledge Engineering Review, 10* (1), 43–62.

Wallsten, T. S., Budescu, D. V., Zwick, R., & Kemp, S. M. (1993). Preferences and reasons for communicating probabilistic information in verbal or numerical terms. *Bulletin of the Psychonomic Society, 31,* 135–138.

Wellman, M. P., & Liu, C.-L. (1995). State-space abstraction for anytime evaluation of probabilistic networks. In P. Besnard & S. Hanks (Eds.), *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 567–574). San Francisco: Morgan Kaufmann.

Xu, H., & Smets, P. (1995). Generating explanations for evidential reasoning. In P. Besnard & S. Hanks (Eds.), *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 574–581). San Francisco: Morgan Kaufmann.

Yager, R. R. (1992). Expert systems using fuzzy logic. In R. R. Yager & L. A. Zadeh (Eds.), *An introduction to fuzzy logic applications in intelligent systems* (pp. 27–44). Boston: Kluwer.

Yager, R. R., Kacprzyk, J., & Fedrizzi, M. (Eds.) (1994). *Advances in the Dempster-Shafer theory of evidence*. New York: Wiley.

Yager, R. R., & Zadeh, L. A. (Eds.) (1992). *An introduction to fuzzy logic applications in intelligent systems*. Boston: Kluwer.

Zadeh, L. A. (1981). Possibility theory and soft data analysis. In L. Cobb & R. Thrall (Eds.), *Mathematical frontiers of social and policy sciences* (pp. 69–129). Boulder, CO: Westview Press. Reprinted in collection by Dubois et al. (1993).

Zadeh, L. A. (1994). Fuzzy logic, neural networks, and soft computing. *Communications of the ACM, 37* (3), 77–84.

Zarley, D., Hsia, Y.-T., & Shafer, G. (1988). Evidential reasoning using DELIEF. *Proceedings of the Seventh National Conference on Artificial Intelligence*, St. Paul, MN, 205–209. Reprinted in collection by Shafer and Pearl (1990).

Author's vita

Anthony Jameson

Anthony Jameson is a senior researcher and project director in the Department of Computer Science at the University of Saarbrücken, Germany. He received a B.A. in philosophy at Harvard College, a master's degree in psychology at the University of Hamburg, and a Ph.D. in psychology at the University of Amsterdam. Until 1993 he was a lecturer in human-computer interaction at the University of Nijmegen in the Netherlands.

His psychological research has focused on the ways in which people assess the knowledge of others in everyday situations. His user modeling research integrates the results of this work into the design of user modeling components for dialog systems.