User-expertise modeling with empirically derived probabilistic implication networks

MICHEL C. DESMARAIS and AMEEN MALUF
Centre de recherche informatique de Montréal
1801 McGill College bureau 800, Montréal, Qc, Canada H3A 2N4 tel: 514-398-1789
fax: 514-398-1244 e-mail: desmarais@crim.ca, amaluf@crim.ca

and

JIMING LIU
Department of Computing Studies, Hong Kong Baptist University

Abstract. The application of user-expertise modeling for adaptive interfaces is confronted with a number of difficult challenges, namely, efficiency and reliability, the cost-benefit ratio, and the practical usability of user modeling techniques. We argue that many of these obstacles can be overcome by standard, automatic means of performing knowledge assessment. Within this perspective, we present the basis of a probabilistic user modeling approach, the POKS technique, which could serve as a standard user-expertise modeling tool.
The POKS technique is based on the cognitive theory of knowledge structures: a formalism for the representation of the order in which we learn knowledge units (KU). The technique permits the induction of knowledge structures from a small number of empirical data cases. It uses an evidence propagation scheme within these structures to infer an individual's knowledge state from a sample of KU. The empirical induction technique is based, in part, on statistical hypothesis testing over conditional probabilities that are determined by the KUs' learning order.
Experiments with this approach show that the technique is successful in partially inferring an individual's knowledge state, either through the monitoring of a user's behavior, or through a selective questioning process. However, the selective process, based on entropy minimization, is shown to be much more effective in reducing the standard error score of knowledge assessment than random sampling.

Key words: knowledge assessment, knowledge spaces, automatic knowledge structure induction, Bayesian network inferences, computer assisted testing, user modeling

1. Introduction

Among the different attributes that can be included in a user model, the level of knowledge, or the expertise in a given domain, is of fundamental importance. It allows the use of an adequate level of detail and the appropriate choice of vocabulary in the dialogue with the user. In particular, help given to the user should adapt to the user's knowledge level, whether it corresponds to standard on-line help text, or to something more advanced as coaching, tutoring, or cues given to the user.

However, in spite of these apparent benefits that can be achieved with a user-expertise model, and in spite of a number of research prototypes, no adaptive interfaces based on user-expertise modeling have been deployed so far in large-scale, real-life applications. Of course, it takes a long time to go from a research prototype to an operational application, but given that some user-expertise techniques have been around for more than a decade, such as the simple novice/expert stereotyping techniques, it is legitimate to ask ourselves why the apparent benefits of adapting the user interface to user-expertise are not being exploited in real-life applications. We will consider potential obstacles to the application of user-expertise models in adaptive interfaces before presenting the POKS technique and discussing how it addresses some of these obstacles.

1.1. Challenges to user-expertise modeling

Efficiency and reliability

The need for assessing expertise *efficiently and reliably* is an essential requirement of a user-expertise model. It is a task that humans are particularly good at, as we can rapidly assess someone's state of knowledge in a domain with only a few questions, or after a short discussion, if we know this domain well enough ourselves. This same efficiency is what we expect from a user-expertise model. Failure to assess effectively and efficiently the user knowledge state would generally result in unacceptable behaviors in the user interface, such as going through a long series of redundant questions[⋆] with the user prior to building a reliable model, or taking inappropriate actions based on an inexact model. Moreover, we cannot rely upon asking the user directly whether he belongs to the expert/novice category in a domain of knowledge, because self-assessment of this type is known to be unreliable, and the domain may involve different sub-domains in which one can be novice in one and an expert in another.

Cost-effective means

User-expertise modeling is, at its best, still in the early state of craftsmanship, as opposed to being a routine procedure that can be reproduced at will, and by anyone with minimal training. Consequently, it is a costly process. Moreover, user-expertise modeling does not generate any benefit by itself. Only through adaptive interfaces will this information be put to profit, which involves yet another phase of complex user interface design in which few people have any experience. Finally, the precise assessment of the benefit of an adaptive interface is difficult and still being debated.

[⋆] Note that a question will be *perceived* as redundant if it is an easy question administered after a success at more difficult questions, or vice-versa, even though they really are two different questions.

Usability and standardization

Another obstacle, closely linked to cost-effective means, is the usability and standardization of the tools for performing user-expertise modeling, and for designing interfaces based on such models. It is a logical conclusion that cost-effectiveness requires standard and easy-to-use expertise assessment tools in order to generate applications that will use this information, but it is worth stressing that the complexity of using such tools for building adaptive interfaces may well be overwhelming. Even with good user-expertise assessment modules, it still is a complex task to design adaptive help or tutoring tools, for example. Hence, great efforts must be devoted to providing user-expertise models that are designed to facilitate their use by interface designers.

Thus it is crucial to deliver a reliable, cost-effective means for user-expertise modeling. If an efficient, "off-the-shelf" and easy-to-use tool were available for this purpose, then the attractiveness of designing adaptive interfaces based on expertise models might increase considerably.

1.2. Perspective on the current approach

The above challenges to user-expertise modeling have remained central considerations in the choice of the current approach and throughout its development. It is thus important to position this approach, as well as alternative approaches, with respect to these considerations.

In particular, the questions of efficiency and usability of user-expertise modeling are important aspects that are generally overlooked in scientific investigations. They are, nevertheless, very relevant for reasons of impact and viability that user-expertise modeling can have in the "real world".

Although this statement is subject to discussion, we can assume, for example, that the popularity of the “stereotypes” approach to user modeling (Benyon et al., 1987; Chin, 1989; Kay, 1994; Rich, 1979; Takeuchi and Otsuki, 1988) stems directly from the simplicity to understand, to implement, and to build adaptive interfaces upon this type of model. It is a very "usable" theory. Unfortunately, it can prove inefficient in some contexts due to its coarse granularity. For example, the expert/novice categorization, coupled with inference rules designed to put a user into one of the two categories, will generally not suffice for applications that cover many dimensions of expertise and a large span of user-expertise. Even if the model were reliable in classifying a user in the right category, the actual use of this information could be unreliable because of its imprecision.

Of course, one could choose to have more categories to divide the expert/novice scale, each having their own inference rules to categorize the user. In addition, it is possible to have one scale for each of the sub-domains of expertise, ensuring that each scale does cover a single dimension of expertise. Hierarchical organizations of user models have also been proposed by Rich (1979; see also Kobsa, 1992) to allow inferences within hierarchies of stereotypes. In doing so, we resolve the lack of preciseness of the model, thus making it potentially more effective, but we also introduce the problem of complexity, both in designing the model, and in using it.

Overlay models of user-expertise (Goldstein, 1982) constitute an alternative to stereotypes. An overlay model is based on the simple principle of defining a given user model as a subset of a global set of knowledge units (KU). This means of user modeling is much more precise and flexible, allowing for C^n number of different user-models instantiations, where n is the number of KU in the knowledge domain, and C is the number of states each KU can take (eg. "mastered", "not-mastered"). But with this advantage comes the drawback that it constitutes a much more complex and costly model to design and to use. For example, assessing directly each KU by a single question requires n questions. To overcome this difficulty, overlay models generally include relations among KU that simplify the task of assessing someone's knowledge state. Relations such as "if you know KU_a, then you also know KU_b", allow inferences to be made in the knowledge assessment process and can greatly reduce the effort involved. Nevertheless, with a large number of KU, it becomes very impractical and error prone to build the complete network of potential relationships among KU by hand.

In fact, a parallel can be drawn between overlay models and stereotypes, to the extent that stereotypes can be considered as subsets of KU. Relations can be defined among stereotypes, thus allowing some types of inferences like those found in overlay models. Hybrid models of this type could prove useful, as they can take advantage of both the stereotypes and the overlay approaches. However, the tradeoff between the granularity and the representative power of a user model, and its complexity and practical utility is still present. In general, the more powerful a model is, the more complex it is to build and to use, be it closer to the stereotypical or to the overlay approach.

The answer thus relies on the automation of the whole process. Indeed, we strongly believe that the most promising way to provide powerful enough user models, such as overlay models paired with inference relations, while making their construction and their use cost-effective and easy-to-use, is through automatic tools both for building user models and for assessing a specific knowledge state.

This is probably the most important advantage of the current approach to expertise modeling: it builds a network of implication relations among KU from a small sample of user data sets, and it uses this induced network to assess more efficiently someone's knowledge state with a limited number of observations, or questions. Because this process is totally automated, it limits the job of the interface designer to the following two essential steps:

– to define what is a proper set of KU that correctly characterizes the knowledge domain;

– to define what are the interface's adaptivity features based on a specific user knowledge state[⋆].

In essence, the knowledge assessment modeling and assessing process is, by itself, reduced to a simple data gathering procedure, granted that the other steps can be automated.

However, we must remember that cost-effectiveness and usability of user-expertise modeling are not the sole challenges that must be met. The user-expertise model must also be valid and reliable in assessing someone's knowledge state.

The question of validity and reliability has many facets to it. First and foremost, it involves testing whether the approach performs well in assessing user knowledge. This is the subject of section 6. Second, validity also involves the notion of generalization across different contexts of application, and across different knowledge domains. Moreover, it also involves the notion of scalability: how well does the approach lend itself to improvements and generalization? This type of validity cannot be demonstrated by a few empirical experiments, but it has to rely instead on an analysis of its theoretical foundations. We will refer to the theory of knowledge spaces (Falmagne et al., 1990) to show the approach's cognitive foundations. The argument is that if the approach is well grounded in solid cognitive foundations, that is, if the model does accurately reflect the psychological reality of expertise, it has a better chance of being generalizable and scalable.

2. Overlay models, knowledge structures, and related work

The model presented here can be considered as a kind of overlay model of user-expertise representation. The user's knowledge state is represented as a subset of a global set of KU. In addition, the global set of KU is inter-connected with a number of implication relations. These relations, say $A \Rightarrow B$, allow inferences of the type "if A is known, then B must be known" and "if B is unknown, then A is unknown". With a strongly connected structure of KU, the process of assessing someone's knowledge state can be highly efficient, such that only a few KU need to be known in order to draw conclusions about the complete knowledge state.

This structure of implications among KU is determined by the order in which we learn concepts, or acquire competencies, and it constitutes one of the most important characteristics of the general learning process. It is a well-known phenomenon in education (for example, see Gagné, 1966, and the concepts of "knowledge hierarchies") and in psychology. In particular, the work of Falmagne, Doignon, and their colleagues (Falmagne et al., 1990) on *knowledge structures* and *knowledge spaces* was a significant contribution to the formal representation of the interdependencies among KU.

Falmagne et al. have shown that the constraints on the order in which we learn KU can be entirely represented by what is known in the field of Artificial Intelligence as AND/OR graphs. This is equivalent to stating that, for every two

[⋆] Mind you, this can still represent a considerable task!

potential knowledge states that a user can reach, the union of these two states is also a potential knowledge state. Using the terminology of set theory, this condition corresponds to the definition of a closure in the space of knowledge states under the union operator, $\cup$.

In terms of inference rules, AND/OR graphs can be specified with the following two rules:

1. "if A is known, then B, and C, ... and N are known",
2. "if A is known, then either B, or C, ..., or N, is known",

where each rule has only one antecedent and one or more consequents. For each consequent, there is an arc from the antecedent to the consequent, representing the ordering constraint. The second rule, involving alternative consequents, is obviously the one that can be achieved via the ‘OR’ operator in the ‘AND/OR’ graph representation. This type of rule is found in cases where a given knowledge state can be reached through a number of alternative knowledge states. For example, creating a loop in the “C” programming language requires mastery of the “for”, “while”, or “do” constructs, but only one of them is required to reach a knowledge state that contains the ability to create loops.

While AND/OR graphs can represent a set of rules of the two types listed above, another type of graph, namely partial orders, or directed acyclic graphs (DAG), can be used to represent ordering constraints involving the first type of rules only. Referring again to set theory terminology, partial orders imply that the space of knowledge states they represent is closed under both the union, $\cup$, and intersection operators, $\cap$. For example, closure under both $\cup$ and $\cap$ could be violated in the above example. Indeed, it is possible to have a programmer that succeeds in an exercise involving loops but knows only the “for” construct, and someone else who also succeeds at the same exercise but only knows the “while” loop. The union of these two knowledge states represents a subject who can solve a problem involving loops without knowledge of any of the corresponding syntactic constructs. It is impossible to represent this type of disjunctive relationship among KU with partial orders where closure under $\cup$ and $\cap$ is assumed.

Although partial orders are simpler and less powerful formalisms than AND/OR graphs, they are nevertheless very useful and have played a much greater role in user modeling. They were used by a number of researchers to represent the implication relations among KU (eg. Goldstein, 1982; Burton, 1982; Bretch and Jones, 1988). More recently, some researchers have used partial orders in conjunction with Bayesian network propagation theory to assess user knowledge states (De Rosis et al., 1992; Lukas and Albert, 1993; Mislevy and Gitomer, 1995).

3. Partial order knowledge structures, POKS

We will focus on the knowledge structures closed under union and intersection, that is, partial order knowledge structures (POKS) which have the formal properties of DAG. Although POKS do not have the ability to represent alternative means of reaching a given knowledge state, they have great significance in expertise modeling. First, a knowledge state has, in general, many more fixed prerequisites than alternative prerequisites, such that a POKS will contain most of the underlying structure in a knowledge domain. Second, the relations in a POKS are transitive. This constitutes a very desirable feature for the knowledge assessment process and for building parsimonious knowledge structures. Finally, a POKS can also contain probabilistic information that will capture some of the information found in alternative prerequisites: alternative prerequisites will be represented in the structure as "weaker" (in a probabilistic sense) prerequisites than would fixed prerequisites. As such, POKS are very useful in assessing someone's knowledge state, which is the goal of user modeling.

3.1. An example

Let us use an example to illustrate the notions behind a POKS. Figure 1 contains a graphic representation of a plausible POKS for UNIX shell commands. It contains 5 nodes which are listed below with a short explanation that relates to Figure 1's question items:

(1) yacc: parser program generator. It generates a parser in "C" source code and it is generally used in conjunction with "lex" which does the lexical analysis part. "y.output" is the file containing the parsing table information.

(2) lex: lexical analyzer. It also generates "C" source code and writes it in a file named "lex.y.c"

(3) sed: string editor for performing string manipulations on whole files. The '&' stands for the whole string matched.

(4) ar: create library archives: the "ar c <files>" creates the archive file

(5) cc: "C" source code compiler: the "-c" flag is used for creating object files.

Insert figure 1
about here

The arcs between the nodes represent surmise relations within the structure. Some of these relations contain *strict prerequisites*, namely (4) ⇒ (5) and (1) ⇒ (2). Indeed, it is necessary to know about the "cc -c" compiler flag in order to generate archive files. It is also the general case that programs generated with "yacc" will use "lex" as a lexical analyzer pre-processor. However, other surmise relations are of an *empirical* nature, namely (2) ⇒ (5) and (2) ⇒ (3): knowledge of "sed" or of the 'C' compiler's "-c" flag is not a prerequisite to using "lex", but it is generally the case that these two KU (3 and 5) will be learned before KU no. 2. Note also that the relation (1) ⇒ (5) and (1) ⇒ (3) are not explicitly specified because they can be derived from transitivity.

Whether the surmise relations are determined by prerequisites, or by other empirical factors that constrain the order of learning among KU, it must be emphasized that the order may be violated in a number of cases, either because of noise in the assessment of mastery, or because the surmise relation is "weak" and there are many exceptions. Nevertheless, the surmise relation should not be ignored because of the noise or the exceptions. *Instead, it should justify the use of a stochastic approach to modeling this phenomenon.*

Another remark that needs to be made with the example is that the KU, or nodes, are defined in terms of procedural knowledge. This does not need to be so. We could also have included concepts such as "regular expressions", "lexical analysis", "object files”, etc. We chose to limit this example to procedural knowledge because it lends itself directly to an operational definition and leads to simple and unambiguous tests with a simple question. By contrast, a concept such as “regular expressions” is much more complex and would require many test items to cover the understanding of all of its ramifications. Nevertheless, it could be represented as a single node in the knowledge structure, but its assessment should include a set of questions as opposed to a simple question. Consequently, its mastery would be represented as a ratio of success over all questions, instead of a dichotomous value: mastered or not mastered. In fact, it is quite conceivable that a KU could be represented by another knowledge structure, independent from the knowledge structure of which it is a member of. In other words, we could have hierarchies of knowledge structures, where the upper levels represent complex KU that encompass large chunk of expertise and that are tied to a knowledge structure at the lower level, until KU are tied to direct observations at the leaves of this hierarchy. Alternatively a set of KU in a single knowledge structure could cover the concept of regular expressions and their global mastery would represent mastery of this concept on a $[0, 1]$ scale.

In fact, there exist a number of potential architectures for building knowledge structures. However, whatever architecture is chosen, the important factors in defining what should constitute a relevant set of KU are that:

1. KU do represent meaningful and significant units in the domain of knowledge;
2. the user’s mastery of each KU can be reliably assessed, and
3. there is some order in the way users learn KU.

The first two factors depend on the domain expert’s ability to break down the knowledge domain into KU. This is the same ability as the one required for developing a good final exam, for example, and it relates to the theory of psychological testing, for which a large body of theory and practice already exists (see for example Anastasi, 1966). The third factor, the ordering of KU, varies across different knowledge domains, but it is a fairly ubiquitous learning phenomenon. The less any of the three factors above is valid, the less the POKS will contain relations, and the less effective and reliable it will be in assessing a user’s knowledge state.

3.2. Definitions

The above example is useful in explaining the theory of POKS and its application to user modeling. Let us now define more precisely the notions involved.

Assume we have a knowledge domain, denoted Q, composed of n KU:

$$Q=\{U_1, U_2, ..., U_n\}$$

A POKS is a partial order over Q where the nodes represent the domain's KU, $\{U_1, U_2, ..., U_n\}$, and the arcs represent implications, or surmise relations. An arc, say from U_i to U_j, is denoted as $S_{i\Rightarrow j}$.

An individual's knowledge state is denoted by R. In accordance to the standard overlay model representation, the knowledge state is represented as a subset of Q:

$$R=\{U_i \text{ mastered} \mid U_i \in Q\}$$

However, let r denote the *inferred* knowledge state:

$$r=\{P(U_1)\,, P(U_2)\,, ..., P(U_n)\,\}$$

where $P(U_i)$ is the probability associated with KU U_i, and n is the number of KU in Q. Thus, $P(U_i)$ represents the probability that U_i is mastered, i.e. $P(U_i \in R)$. Global mastery of the domain Q can be computed as :

$$\frac{\sum P(U_i)}{n}$$

It can be interpreted as the probability that an arbitrarily chosen KU would be mastered by the user in question. Alternatively, in the context where Q represents a term exam for example, it would constitute an estimate of the subject's expected score over the whole test.

Each arc in a POKS, say $S_{i\Rightarrow j}$, has two associated weights, $W_{i\Rightarrow j}$ and $W_{\neg j\Rightarrow\neg i}$. These weights represent the "strength" of the surmise relation. The choice of estimators for these two weights depends on the inference propagation scheme. For example, in some circumstances, a derivation of the $P(\chi^2)$ value could be used as a measure of the strength of the relation, whereas the conditional probabilities provide a measure of the directionality ($A \Rightarrow B$ or $B \Rightarrow A$). In our case, the weights $W_{i\Rightarrow j}$ and $W_{\neg j\Rightarrow\neg i}$ are represented by odds ratios. These ratios measure the influence that a KU has on the odds of another KU. For example, if KU A has a strong positive influence on the odds of B the ratio $\frac{O(B|A)}{O(B)}$ will be high, such that the observation that A is true will bring the updated probability of B close to one. The details of how the weights are obtained is given in section 4.3 whereas their role in the evidence propagation scheme is found in section 5.1.

4. The induction of POKS from data

Knowledge structures are difficult to construct through a process of knowledge engineering with one or more domain experts. We argued that the lack of an automatic means of constructing them constitutes a serious obstacle to the overlay user modeling approach, because it is too tedious and subject to judgment biases. However, some work, including ours, has begun working towards automatic means of constructing knowledge structures. In the domain of mathematical psychology, Falmagne and his colleagues (Falmagne et al., 1990) have developed a technique for inducing the topology of knowledge structures closed under union, which are more complex to build than POKS. The technique was proven successful for inducing small knowledge structures (below 10 nodes). However, it requires relatively large amounts of data (e.g. 400 reported in Falmagne, 1990).

Parallel to this work in psychology, other researchers from the Artificial Intelligence community have developed a number of techniques to induce Bayesian networks from data (Cooper and Herskovits, 1992; Geiger, 1992; Heckerman et al., 1994; Heckerman and Geiger, 1995; Heckerman, 1995; Spiegelhalter et al., 1993; Pitas et al., 1992). Among some of the important findings on this topic, Cooper and Herskovits report successful results in inducing a simulated Bayesian network of 37 nodes and 46 relations (see also Heckerman et al., 1994, for further developments along the same line). However, more than 3000 cases were necessary to recover the original topology of the network. Geiger (Geiger, 1992) has formulated a learning algorithm for uncovering a Bayesian conditional dependence tree. This algorithm combines entropy optimization with Heckerman's similarity networks modeling scheme (Heckerman, 1991). The work of Pitas et al. (1992) is also relevant to this topic. They use entropy measures to guide the induction of the network. Here again, it is plausible that this technique could be used for knowledge structures.

In our own work, as mentioned earlier, we focused on the induction of knowledge structures closed under union and intersection, that is, knowledge structures that can be completely represented by a partial order. We put a strong emphasis on the ability of this structure to produce correct inferences with small amounts of data (the experiments we report here are based on 19 cases). However, the algorithm does not guarantee inducing the optimal topology of a network with respect, for example, to a minimal entropy criterion, or with respect to the maximum likelihood of a topology given a data set. Instead, we focus on the technique's ability to *perform inferences* with the network, not so much on its ability to recover a "true" underlying topology. This is well justified in user modeling because of two reasons:

– our interest lies first and foremost in assessing a user's knowledge state, not necessarily in uncovering the domain's true knowledge structure, and;
– because a large portion of the knowledge structure's relations are probabilistic, (i.e. they are not based on a strict order from which discrete, true or false, and deterministic inferences could be performed), it follows that the topology is not the only factor that influences the validity of the inferences. The topology represents only the directionality of influence among KU. The relations' weights and the evidence updating scheme are other factors that are just as important and that must be taken into account to assess the ability of a knowledge structure to infer an individual's knowledge state.

The current technique induces, from a small number of data cases, a set of binary implication relations from which we can apply *modus ponens* and *modus tollens* inferencing: for example, if we have a relation $A \Rightarrow B$, then it follows:

– if A is mastered, then B is mastered, and;
– if B is not mastered, then A is not mastered.

These inferences are, in fact, probabilistic in the sense that they determine the probability of mastery of B according to some new evidence of the mastery of A, or conversely, the probability of mastery of A given non-mastery of B. The values will depend on the surmise relation's strength, as determined by its associated weights, $W_{i \Rightarrow j}$ and $W_{\neg j \Rightarrow \neg i}$ (section 3.2) and on the evidence propagation scheme.

4.1. The POKS induction technique

The basic idea behind the network induction technique is that, in an ideal case, if there is an implication relation $A \Rightarrow B$, then we would never expect to find that someone knows A but does not know B. This assertion translates into the following two conditions:

$$P(B \mid A) = 1$$

$$P(\neg A \mid \neg B) = 1$$

However, as we stated above, many surmise relations do not have a strict order, such that the two conditional probabilities will be more or less close, but not equal, to 1. Moreover, sampling errors will affect the measured conditional probabilities. A statistical model of an implication, or surmise relation, is thus necessary.

In essence, the statistical model behind the implication relation is based on two test of hypotheses to verify that the conditional probabilities, $P(B \mid A)$ and $P(\neg A \mid \neg B)$ are above a given minimal threshold, and a third test to verify that the conditional probabilities are different from the initial probabilities. These tests are described in the following two sections.

4.1.1. *Hypothesis tests on* $P(B \mid A)$ *and* $P(\neg A \mid \neg B)$

The two test of hypothesis over the conditional probabilities can be stated as follow:

$$P(\,[P(B \mid A) \leq p_c] \mid D) < \alpha_c \qquad (1)$$

$$P(\,[P(\neg A \mid \neg B) \leq p_c] \mid D) < \alpha_c \qquad (2)$$

where:

p_c : minimal conditional probability chosen for $P(B \mid A)$ and $P(\neg A \mid \neg B)$. It can be considered as an indicator of the strength of the knowledge structure's surmise relations.

α_c : the alpha error of the minimal conditional probability tests. It determines the proportion of relations that erroneously fall below p_c.

D : the frequency distribution of co-occurrences of A and B in a data sample, as illustrated in Table I, and where each value of $N_{\bullet\bullet}$ corresponds to one of the following 4 conditions:

1. $N_{A \wedge B}$: co-occurrences of A mastered and B mastered;
2. $N_{A \wedge \neg B}$: co-occurrences of A mastered and B *not* mastered;
3. $N_{\neg A \wedge B}$: co-occurrences of A *not* mastered and B mastered;
4. $N_{\neg A \wedge \neg B}$: co-occurrences of A *not* mastered and B *not* mastered;

Let us demonstrate how these tests of hypothesis can be conducted by first noting that the frequency pair $(N_{A \wedge B}, N_{A \wedge \neg B})$ and $(N_{\neg A \wedge \neg B}, N_{A \wedge \neg B})$ are stochastic variables with a probability distribution that follows the binomial distribution:

$$Bin(k, n, p)$$

where:

$$k = \begin{cases} N_{A \wedge B} & \text{for the pair } (N_{A \wedge B}, N_{A \wedge \neg B}) \\ N_{\neg A \wedge \neg B} & \text{for the pair } (N_{\neg A \wedge \neg B}, N_{A \wedge \neg B}) \end{cases}$$

$$n = k + N_{A \wedge \neg B}$$

and

$$p = \begin{cases} P(B \mid A) & \text{for the pair } (N_{A \wedge B}, N_{A \wedge \neg B}) \\ P(\neg A \mid \neg B) & \text{for the pair } (N_{\neg A \wedge \neg B}, N_{A \wedge \neg B}) \end{cases}$$

In other words, the probability distribution of each frequency pair is determined by a binomial function with $P(B \mid A)$ or $P(\neg A \mid \neg B)$ as one of its argument, and by two cell values in the distribution D.

Thus, the test of hypothesis for $A \Rightarrow B$ can be obtained by computing by a lower tail confidence interval over a binomial function:

$$p(X \leq N_{A \wedge \neg B}) = \sum_{i=0}^{N_{A \wedge \neg B}} \binom{n}{i} p^{n-i} (1-p)^i \tag{3}$$

where n has the same definition as above, and where p is set to the desired minimal conditional probability, p_c. This formula represents the probability that as small a

TABLE I. Distribution of observed co-occurrences

	B	$\neg B$
A	$N_{A \wedge B}$	$N_{A \wedge \neg B}$
$\neg A$	$N_{\neg A \wedge B}$	$N_{\neg A \wedge \neg B}$

number as X of *un*predicted results would be observed if the true probability of a predicted result were exactly p. The smaller the probability given by the formula is, the less likely it is that the true probability of a predicted result is *less than* p.

4.1.2. *Interaction test*

The two tests of hypothesis on conditional probabilities ensure that the minimal "strength" of the relation is above a predetermined threshold, p_c. However, we still need to verify that the conditional probabilities are different than the non-conditional probabilities, that is:

$$P(B \mid A) \neq P(B)$$

$$P(\neg A \mid \neg B) \neq P(\neg A)$$

These conditions can be verified through a χ^2 test on the 2 × 2 contingency table:

$$P(\chi^2) < \alpha_i \tag{4}$$

where α_i is the alpha error of interaction. For small samples ($N < 50$), the *Fisher exact test* should be used instead.

These three tests are sufficient to characterize a surmise relation and to ensure that (1) its "strength" is above a minimum and (2) that a maximum error tolerance is set.

4.2. AN EXAMPLE OF IMPLICATION RELATION INDUCTION

The following section illustrates how the induction technique is applied to a specific example.

Assume we wish to verify the existence of $A \Rightarrow B$. In the first step of implication relation induction, a two-dimensional contingency table for the co-occurrences of A and B is compiled from an empirical data set. Table II shows a possible table of co-occurrences.

In the second step of the induction method, the above mentioned three tests of hypothesis are conducted.

Suppose that in this example, p_c=0.85 and α_c=α_i=0.20. Accordingly the binomial hypothesis test for $P(B \mid A)$ can be computed as follows from equation 3:

TABLE II. Example distribution of observed co-occurrences

	B	$\neg B$
A	20 ($N_{A \wedge B}$)	1 ($N_{A \wedge \neg B}$)
$\neg A$	8 ($N_{\neg A \wedge B}$)	1 ($N_{\neg A \wedge \neg B}$)

$$\begin{aligned}P(x \leq N_{A\wedge\neg B}) &= P(x \leq 1)\\ &= P(x{=}0) + P(x{=}1)\\ &= \binom{21}{0}\ 0.85^{21}\ 0.15^{0} + \binom{21}{1}\ 0.85^{20}\ 0.15^{1}\\ &= 0.155\end{aligned}$$

hence, $P(x \leq N_{A\wedge\neg B}) < \alpha_c$

where symbol $\binom{j}{k}$ represents the number of combinations of k in j. The inference with $A \Rightarrow B$ in the *modus ponens* direction is significant with confidence level $(1 - \alpha_c)$.

In a similar way, the test for $P(\neg A \mid \neg B)$ yields:

$$\begin{aligned}P(x \leq N_{A\wedge\neg B}) &= \binom{2}{0}\ 0.85^{2}\ 0.15^{0} + \binom{2}{1}\ 0.85^{1}\ 0.15^{1}\\ &= 0.98\end{aligned}$$

hence, $P(x \leq N_{A\wedge\neg B}) \not< \alpha_c$

Since the test of $P(\neg A|\neg B)$ is not satisfied, $A \Rightarrow B$ cannot be used for *modus tollens* inference. Hence, the implication relation $A \Rightarrow B$ is rejected. Indeed, with $N_{A\wedge\neg B}$=1, it would require a value greater or equal to 19 for $N_{\neg A\wedge\neg B}$.

The interaction test (inequality 4) does not need to be conducted in this example since the second test on $P(\neg A \mid \neg B)$ failed.

4.3. Estimating $W_{a\Rightarrow b}$ and $W_{\neg b\Rightarrow\neg a}$

Besides the POKS' topology, the two parameters linked with implications relations, $W_{a\Rightarrow b}$ and $W_{\neg b\Rightarrow\neg a}$, are also determined by the data samples. The choice of estimators for these two weights depends on the inference propagation scheme. In the current study, the values $W_{a\Rightarrow b}$ and $W_{\neg b\Rightarrow\neg a}$: correspond to the two odds ratios:

$$\begin{aligned}W_{a\Rightarrow b} &= \frac{O_{\text{est}}(B \mid A)}{O_{\text{est}}(B)}\\ W_{\neg b\Rightarrow\neg a} &= \frac{O_{\text{est}}(A \mid \neg B)}{O_{\text{est}}(A)}\end{aligned}$$

where $O(X)$ represents the odds of X and $O(X \mid Y)$ represents the odds of X given Y, that is:

$$O_{\text{est}}(X) = \frac{P_{\text{est}}(X)}{P_{\text{est}}(\neg X)}$$

$$O_{\text{est}}(X \mid Y) = \frac{P_{\text{est}}(X \mid Y)}{P_{\text{est}}(\neg X \mid Y)}$$

The values for $P_{\text{est}}(B \mid A)$ and $P_{\text{est}}(\neg A \mid \neg B)$ can be computed from the data sample. The following formula was used to obtain their values:

$$P_{\text{est}} = \frac{k+1}{n+2} \tag{5}$$

where:

$$n = \begin{cases} N_{A \wedge B} + N_{A \wedge \neg B} & \text{for } P(B \mid A) \\ N_{\neg A \wedge \neg B} + N_{A \wedge \neg B} & \text{for } P(\neg A \mid \neg B) \end{cases}$$

and,

$$k = \begin{cases} N_{A \wedge B} & \text{for } P(B \mid A) \\ N_{\neg A \wedge \neg B} & \text{for } P(\neg A \mid \neg B) \end{cases}$$

5. Using Bayesian network inferences with POKS

Once a knowledge structure is obtained and parametrized according to the method described in the previous section, the task of assessing someone's knowledge state corresponds to using this structure with a Bayesian network induction technique in order to estimate each node's "truth value", i.e. the probability of mastery.

More specifically, every time a node is assigned a new probability, such as when mastery or failure is observed, then every other node it connects to is reassigned a new probability of mastery, and the process is repeated recursively until all paths from the originating node are followed. In the case where the probability of mastery is increased (i.e. observation of a success), then the implication links are followed in the forward direction, whereas if the probability is decreased (i.e. failure), the links are followed in the backward direction.

Although the basic principles behind the use of Bayesian network induction for knowledge assessment are relatively straightforward, the sound and complete application of Bayesian inference is a complex matter. Well known algorithms such as (Lauritzen and Spiegelhalter, 1988; Pearl, 1988) are among mathematically sound techniques for performing Bayesian network updating. They are relatively complex and computationally expensive, but they guarantee correct answers provided a number of assumptions are met. Other approaches are based on simulation (Henrion, 1988; Pearl, 1992). These approaches yield approximate results with an accuracy that is a function of the computational cost.

In our case, considering that we build and parametrize the knowledge structure with relatively small samples, the inferences that result from such structures are doomed to contain a significant amount of noise and we cannot assert that all assumptions required for the above mentioned algorithms are fully met. For these reasons, the choice of a sophisticated Bayesian inference scheme that can yield exact results is not necessarily worth the implementation efforts and computational cost involved, because sampling noise or violated assumptions can outweigh the improvement in accuracy these methods provide over other simpler ones, such as MYCIN (Buchanan and Shortliffe, 1984) or Dempster-Shafer (see Buchanan and Shortliffe, 1984). Hence, in accordance with our principles of parsimony, ease-of-use and ease of implementation, we opted for a simpler method that nevertheless was proven efficient in the context of expert system inferences: the Duda et al. (1976) evidence propagation scheme based upon odds ratios. It was used initially in the Prospector expert system and we will refer to it as the Prospector evidence propagation scheme.

It should be stressed that better performance might be obtained with more recent Bayesian network propagation algorithms, but this would also come at a greater implementation and computational cost. As such, the Prospector algorithm should not be considered as a second class choice if we take into account these pragmatic issues of cost in the user knowledge assessment approach (cf. the remarks on potential obstacles to user modeling made in the introduction). This algorithm should, instead, be considered as the minimal performance standard that one can expect from the general POKS approach.

5.1. The Prospector evidence updating scheme

We will briefly describe the Prospector algorithm in this section. However, the reader is referred to Giarratano and Riley (1989) for an introduction to the topic and a more detailed description of the technique.

The Prospector algorithm uses the notions of *likelihood of sufficiency* and *likelihood of necessity* for updating a node's probability. Given a surmise relation $A \Rightarrow B$, these likelihood are defined respectively as:

$$LS = \frac{O(B \mid A)}{O(B)} \tag{6}$$

$$LN = \frac{O(A \mid \neg B)}{O(A)} \tag{7}$$

They correspond respectively to $W_{a \Rightarrow b}$ and $W_{\neg b \Rightarrow \neg a}$, the POKS relations' weights described in section 4.3.

It follows that if we know A to be true (i.e. $P(A)=1$), then the probability of B can be updated using this form of the above equation :

$$O(B \mid A) = LS\, O(B) \tag{8}$$

and conversely, if B is known false ($P(B)=0$), then :

$$O(A \mid \neg B) = LN\, O(A) \tag{9}$$

The odds ratios are obtained using the estimated probabilities (formula 5).

In other words, and assuming we observe $P(A)$ =1 with our example $A \Rightarrow B$, it is the relative effect of A on the odds of B that is propagated in the relation. If A has a strong effect on B, LS will be very high and it will bring the probability of B close to 1, whereas if A has little effect on B (i.e. if LS is close to 1) then the probability of B will only increase slightly. An analogous relationship holds for LN when propagating backward.

5.2. Pooling and propagation of evidence

Equations 8 and 5.1 provide means to update a probability from a single observation. However, we need to specify the case of node update based on multiple observations. Moreover, we need a mean of updating from partial evidence, which corresponds to the case where a node's probability has changed during the evidence propagation process, but not from a direct observation.

The process of multiple evidence update relies on the assumption of independence (i.e. the evidence propagation problem). From the knowledge structure's network topology, we can derive a number of independence assumptions. For example, given the following network relationships:

$$A \Rightarrow B$$
$$B \Rightarrow C$$
$$D \Rightarrow B$$

we can conclude that $P(B \mid A)$ =$P(B \mid A, C, D)$, or we can conclude that if B is known true, then $P(C \mid A)$ =$P(C)$, etc.

Most relevant in our context, the independence assumption enables the "pooling" of evidence. For each of the incoming arcs in a node H, $\{E_1 \Rightarrow H, E_2 \Rightarrow H, ..., E_n \Rightarrow H\}$, we can write:

$$O(H \mid E_1, E_2, ..., E_n) = \prod_i^n L_i O(H)$$

where L_i=LS if E_i is known true and L_i=LN if it is known false.

This makes updating with values that are *true* or *false* very simple and efficient. However, it does not provide an updating mechanism with probabilistic values. An *ad hoc* scheme was developed for this purpose. Given the implication chain $A \Rightarrow B \Rightarrow C$, the basic principle is to respect the following conditions:

$$P(C \mid A) = \begin{cases} P(C \mid B) & \text{if } P(B \mid A) = 1 \\ P(C) & \text{if } P(B \mid A) = P(B) \end{cases}$$

and

$$P(A \mid \neg C) = \begin{cases} P(A \mid \neg B) & \text{if } P(B \mid \neg C) = 0 \\ P(A) & \text{if } P(B \mid \neg C) = P(B) \end{cases}$$

A linear interpolation is made for every value in between the conditions' boundary values $\{0, P(B), 1\}$ (two linear segments are thus necessary for a full model—one for the forward propagation, $P(C \mid A)$, and another one for a backward propagation, $P(A \mid \neg C)$). The next section provides more details on the computations involved, but the reader is again referred to Giarratano and Riley (1989) for a more complete overview.

5.3. Evidence propagation example in a simple knowledge structure

We will use Figure 2's simple knowledge structure to demonstrate a short example of evidence propagation with the Prospector scheme. This example is useful to readers unfamiliar with Bayesian inference schemes and, in particular, the Prospector evidence propagation scheme, but it also helps to build a clearer picture of the overall system's behavior.

Figure 2's schema contains three surmise relations. The nodes' joint distribution for each relation is reported in Table III. The relations' values for LN and LS were computed over this joint distribution. For example, the value of LS for $U_1 \Rightarrow U_3$ can be obtained using equation 6. However, let us use an equivalent form of equation 6 that is simpler to compute because it involves probabilities instead of odds ratios (Giarratano and Riley, 1989):

$$LS = \frac{P(A \mid B)}{P(A \mid \neg B)} \tag{10}$$

Using equation 5 to compute the estimates of the probabilities we have:

$$\begin{aligned} LS &= \frac{(N_{A\wedge B} + 1)/(N_{A\wedge B} + N_{\neg A\wedge B} + 2)}{(N_{A\wedge\neg B} + 1)/(N_{A\wedge\neg B} + N_{\neg A\wedge\neg B} + 2)} \\ &= \frac{(16 + 1)/(16 + 12 + 2)}{(0 + 1)/(0 + 36 + 2)} \\ &= 21.53 \end{aligned}$$

The values for LN can be obtained in a similar manner from the joint distributions. The likelihoods LN and LS are the only parameters required by the propagation algorithm, and we can now proceed with the example to demonstrate the details of the propagation computations.

Insert figure 2
about here

Evidence propagation occurs after a node's probability has changed. Evidence will propagate forward to the connected nodes if the change is positive, whereas it will propagate backward if the change is negative. Assuming that observation of mastery changes a node's probability to 1, and observation of non-mastery to 0, let us simulate two scenarios of evidence propagation. Table IV reports the results of such a simulation when the initial probability of all nodes are set to 0.5 (note that they could also have been initialized to their estimated probability). The first scenario corresponds to the successive observations of mastery of U_1 followed by U_2. The second scenario corresponds to the observation of non-mastery of U_4.

Let us examine in detail the computation involved in the very first observation (U_1) in order to see how these results are obtained:

According to equation 8, we have:

$$\begin{aligned} O(U_3 \mid U_1) &= LS\ O(U_3) \\ &= 21.53 \times \frac{0.5}{1-0.5} \\ &= 21.53 \end{aligned}$$

From the general formula

$$P(X) = \frac{O(X)}{O(X)+1} \tag{11}$$

we can derive $P(U_3 \mid U_1)$ from $O(U_3 \mid U_1)$ as follows:

$$\begin{aligned} P(U_3 \mid U_1) &= \frac{21.53}{21.53+1} \\ &= 0.956 \end{aligned}$$

Next, the propagation continues forward from node U_3 to node U_4. According to the propagation scheme (section 5.2), this process consists in estimating $P(U_4 \mid U_1)$ based on a linear interpolation between $P(U_4)$ and $P(U_4 \mid U_3)$. The updated value is in part a function of the amount of change in $P(U_3)$ induced by the observation of U_1. It is computed as follows:

$$P(U_4 \mid U_1) = P(U_4) + \frac{P(U_3 \mid U_1) - P(U_3)}{1 - P(U_3)} \left[P(U_4 \mid U_3) - P(U_4) \right] \tag{12}$$

All values in this formula are known with the exception of $P(U_4 \mid U_3)$ which is derived in the same manner as $P(U_3 \mid U_1)$ was derived:

$$\begin{aligned} O(U_4 \mid U_3) &= LS\ O(U_4) \\ &= 17.95 \times \frac{0.5}{1-0.5} \\ &= 17.95 \end{aligned}$$

TABLE III. Joint distributions and likelihood ratios

Relation	**Cell**				**Likelihood**	
	$N_{A\wedge B}$	$N_{A\wedge\neg B}$	$N_{\neg A\wedge B}$	$N_{\neg A\wedge\neg B}$	LS	LN
$U_1 \Rightarrow U_3$	16	0	12	36	21.53	0.075
$U_2 \Rightarrow U_3$	20	4	8	32	5.32	0.245
$U_3 \Rightarrow U_4$	28	0	12	24	17.95	0.051

and by equation 11 we have:

$$\begin{aligned} P(U_4 \mid U_3) &= \frac{17.95}{17.95 + 1} \\ &= 0.947 \end{aligned}$$

Substituting the values in equation 12 yields:

$$\begin{aligned} P(U_4 \mid U_1) &= 0.5 + \frac{0.956 - 0.5}{1 - 0.5} [0.947 - 0.5] \\ &= 0.908 \end{aligned}$$

A number of remarks could be made about the scenario's results, but let us summarize this discussion with two important observations:

- the updated probability of a node is a function of (1) its prior probability, (2) the amount of change in the neighboring node, and (3) the strength of the neighboring node's relation with the current node (note that this corresponds to three expressions on the right side of equation 12);
- the relative change induced by an observation will decrease as a function of distance (i.e. the number of arcs traversed from the observed node).

These properties are what we expect in an evidence propagation algorithm. Thus, although the Prospector scheme does not conform to exact Bayesian inference, it has all the desired properties to make it a good approximation.

6. Experiments in knowledge assessment

The knowledge structure induction technique, paired with the Prospector evidence propagation scheme, was used in knowledge assessment experiments with two different approaches to knowledge assessment, namely assessment through random sampling, and assessment through a selective questioning process. These experiments were conducted with data on UNIX commands.

TABLE IV. Scenarios of evidence propagation

Node observed	**Probability**			
	U_1	U_2	U_3	U_4
Scenario 1:				
Initial	0.5	0.5	0.5	0.5
U_1	*1.0*	0.5	0.956	0.908
U_2	*1.0*	*1.0*	0.991	0.977
Scenario 2:				
Initial	0.5	0.5	0.5	0.5
$\neg U_4$	0.111	0.226	0.048	*0.0*

6.1. The UNIX$^{\mathrm{TM}}$ data set

The UNIX data set is composed of 34 KU. They represent 34 items in a questionnaire on UNIX commands that was administered to 19 subjects. The questions span a very large range of difficulty, from basic file manipulation, to advanced data processing and system maintenance commands. Similarly, the 19 subjects' expertise also spans a very wide range, from that of casual users to professional programmers and system administrators. Figure 3 provides a performance graph for the 19 subjects and figure 4 contains some examples of the 34 questions that represent the KU. It shows that the performance spans from 8 correct answers to 33. The average success rate is 64%. A large range of subject scores with an average around 50% is generally preferable in order to construct the full knowledge structure.

Insert figure 3 about here

Insert figure 4 about here

6.2. Simulation methodology and knowledge structures construction

The performance of the knowledge assessment scheme is done through a simulation process: we simulate the observation of a KU and propagate this evidence through the knowledge structure and, thereafter, verify the accuracy of the inferences made. To avoid any positive bias in the simulations, a different knowledge structure is constructed for each and every subject, such that the knowledge structure does not contain the data from the subject with which the simulation is conducted, i.e. we built 19 individual knowledge structures from data from 18 subjects.

Because of the small number of data cases, we used low constraining values for p_{min} and for α_c and α_i:

$$p_{\mathrm{min}}=\alpha_c=\alpha_i=0.5$$

The resulting knowledge structures were composed of 224 to 331 implication relations with an average around 300.

In spite of the significant amount of noise introduced by the choice of low constraining parameters, the knowledge structures are still relatively efficient because the knowledge domain is highly structured. However, it must be stressed that these values could reset in different result patterns under different conditions (see section 6.5 on the effect of p_{min}, α_c, and α_i).

6.3. Knowledge assessment through sampling and observation

The first experiment was conducted with the paradigm of assessing someone's knowledge state without any control over which KU is sampled. This situation is an approximation of unobtrusive monitoring of a user's behavior through the computer. It is typical of advisory systems in which recommendations are provided after sub-optimal behaviors are detected.

We set the initial probability of each KU at 0.5, thus providing the system with no initial information. However, we could also have set the initial probability at the value of the average success rate from our sample of 19 subjects. Previous experiments (Desmarais and Liu, 1993) have shown that the general behavior of the system is similar in both cases[⋆].

The experiments consisted of randomly sampling a portion of an individual's knowledge state, and propagating the information on the mastery and non-mastery of KU across the knowledge structure, and finally reassigning new probabilities to unsampled KU. The residual error between the probability estimates and the real values are computed for all KU. This process is applied for sampling from 0% to 100% of an individual's knowledge state, which of course ultimately results in perfect assessment.

6.3.1. *Results*

Figures 5-a and 5-b provide two different perspectives on the simulation's results. They illustrate the evolution of the absolute and the relative standard error score as a function of the knowledge state's sampled proportion. The standard error score is defined as:

$$\mathrm{S.E.} = \sqrt{\frac{\sum (\mathrm{x_{est}} - \mathrm{x_{obs}})^2}{\mathrm{N}}} \tag{13}$$

In our context, the variable N corresponds to the number of KU (34), x_{obs} is 1 if the corresponding KU is *known* and 0 otherwise, and x_{est} is the probability of mastery as estimated by the system. This formula can be interpreted as the average error in predicting individual KU mastery for a given subject.

Figure 5-a illustrates the *absolute* standard error evolution from 0% to 100% of the knowledge state's sampled proportion, averaged over all 19 subjects. A linear function, that starts and ends at the same data points as the simulation curve, corresponds to the no-inference condition. It represents the reduction in standard error due to observation only and it is given as a comparison point. The results assume that no error remains after a KU is observed, and thus the standard error is 0 after all KU are observed.

The absolute standard error of estimates provides an idea of the accuracy of the predictions as a function of the sampled knowledge state proportions. However, what is most informative for the knowledge assessment performance is to verify the *reduction of standard error due to the inferences*. This is provided in figure 5-b. It represents the relative standard error reduction with respect to the *unobserved* KU. It is thus computed over the subset of KU that have not yet been observed and with the following formula:

$$\Delta \mathrm{S.E.}_{\mathrm{i}} = \frac{\mathrm{S.E.}_{\mathrm{i}} - \mathrm{S.E.}'_{\mathrm{i}}}{\mathrm{S.E.}_{\mathrm{i}}}$$

[⋆] Note that this statement may not be true for average success rates close to 0% or 100%.

where S.E.$_i$ is the original standard error of unobserved KU before any observations, and S.E.$'_i$ is the standard error of the same set of KU after the update from the observed KU. Figure 5-b represents the relative standard error reduction averaged over all 19 subjects. This measure can be interpreted as the information added by the inferences on a scale of 0 to 1.

Insert figure 5
about here

The results show that the relative standard error reduction is proportionally small at the beginning, but that it is cumulative and follows a progressive increase from 0% to 100% sampling proportion[⋆]. It reaches approximately a 20% relative error reduction after sampling half the knowledge states. This is significant in demonstrating that the knowledge assessment scheme is valid with such a small sample size; but it nevertheless constitutes a relatively small contribution from a practical point of view[⋆⋆]. We will see in the section on controlled knowledge assessment how this can be improved with a different sampling scheme.

6.3.2. *A note on knowledge assessment through monitoring*

Although, for our purpose here, random sampling yields an acceptable approximation for the performance of the knowledge assessment process when monitoring the user's behavior, the general assumption that random sampling of KU corresponds to the monitoring paradigm is inexact in at least two ways: (1) commands have different frequencies of usage, which makes some of them much more likely to be observed than others, and, most importantly, (2) we cannot observe non-mastery directly and thus could not conclude that a KU is unknown (i.e. we can only infer *mastery* from observation of correct usage). As a consequence, random sampling in mastered KU solely would correspond more closely to knowledge assessment through monitoring. However, inference of solely mastered KU would necessarily lead to a bias towards overestimating the average mastery and would require a correction.

One technique that can be used to compensate for the lack of direct observation of non-mastery is based on the probability of occurrence of skills observation, within a given period of time, or within the context of a task. For example, if, for someone who masters an application's command C_i, f_i is the frequency of usage of this command in a set of N commands, or in a period of time P, then the probability of occurrence of C_i for every N commands, or for every period P, can be approximated with the ratios $\frac{f_i}{N}$ or $\frac{f_i}{P}$ respectively. Thus, the probability that someone does not master C_i given that it has not been observed in N commands, or in a period P, can be easily estimated from a binomial, or a Poisson distribution (see for example Desmarais, et al, 1993 and Desmarais, et al., 1987). This allows a probabilistic inference of unknown KU through a monitoring process.

[⋆] In fact, the results stop at 33 of the 34 knowledge states' KU sampled because no error reduction is possible after all KU are observed.

[⋆⋆] This conclusion holds in the context of this single experiment, but with a different sample size and a different knowledge domain, the random sampling could provide very different results.

Care must be taken, however, to take into account clustering effects that often occur when monitoring KU and which can bias the estimates of the probability of occurrences.

Another technique is through the observation of inefficient means of completing some tasks, which can be indicative of the fact that the efficient means are not mastered. Here again, care should be taken to ensure that the means taken are truly inefficient and that the observations do not stem from other factors that can explain the use of inefficient methods (see for example, Zissos and Witten, 1985, on heuristics to determine true inefficiencies).

6.4. Controlled knowledge assessment

Another paradigm for knowledge assessment, different from sampling based on unobtrusive observation of KU, corresponds to the controlled choice of sampled KU. In this case, questions or test items are administered to the user in order to verify mastery or non-mastery of KU. Although, for some applications, the choice could be determined by objectives such as presenting test items that are at an appropriate level of difficulty for the user (this would correspond to an "exerciser" application, for example), we will focus on the objective of optimizing the knowledge assessment process. The questions, or KU, will thus be chosen based on the *expected amount of information they will provide*. The sampling process will thereby provide a good estimate of the user's knowledge state *with a minimal number of KU tested.*

6.4.1. *Entropy-driven sampling*

One such method of optimizing the knowledge assessment process, to provide the maximum information with the least number of KU tested, is the entropy-driven sampling. The objective is to choose the KU that has the highest chance of reducing the entropy in the knowledge structure.

Entropy is essentially a measure of the amount of uncertainty in a system of stochastic events. Its general definition is:

$$H(Z) = -\sum_{z} p_z \log p_z$$

where Z is a system of stochastic events, $\{z_1, z_2, ..., z_n\}$. Given N systems and assuming they are conditionally independent, the global entropy is their summation:

$$H(Z_1, Z_2, ..., Z_N) = -\sum_{i}^{N} \sum_{z} p_{iz} \log p_{iz}$$

A knowledge structure can be considered as a set of systems corresponding to the network's nodes, each node having two possible states (events): true or false. Thus, given a knowledge structure, Q, composed of N KU, the entropy can be measured in the following way:

$$\begin{aligned} H(Q) &= H(U_1, U_2, ..., U_N) \\ &= -\sum_{i}^{N}\sum_{z}^{2} p_{iz} \log p_{iz} \\ &= -\sum_{i}^{N} (p_i \log p_i + (1-p_i) \log(1-p_i)\,) \end{aligned}$$

where p_i is the probability that KU U_i is mastered by the user. This equation assumes all KU are independent of each other, which is of course an invalid assumption, by the definition itself of a Bayesian network with one or more relations. However, it is a close enough approximation for our purpose, as the simulation demonstrates in the next section.

Based on the above formula, it is now possible to derive the KU that is most likely to reduce entropy. This is given by the expected value of entropy given an observed KU, as computed by the following formula:

$$H(Q' \mid U_i \text{ is observed}) = \left[p_i \cdot H(Q' \mid U_i{=}1)\,\right] + \left[(1-p_i) \cdot H(Q' \mid U_i{=}0)\,\right]$$

$H(Q' \mid U_i{=}1)$ is the total entropy computed from the knowledge structure's state after a success at KU U_i and $H(Q' \mid U_i{=}0)$ is the total entropy after a failure. Given the expected value of entropy for every KU, the choice of the most informative KU thus corresponds to the KU U_i with the lowest value.

6.4.2. *Results*

Figure 6 shows the result of the entropy-driven knowledge inference process. It can be seen that performance is significantly better than the random inference scheme. The relative standard error reduction is approximately reduced by half after sampling 50% of the knowledge state, compared to approximately 20% for the random sampling condition. An almost perfect assessment is reached after 29 KU are observed.

Insert figure 6
about here

6.5. THE EFFECT OF p_c AND OF α_i, α_c

The experiments described in the previous sections were conducted with:

p_c=0.5

α_i=α_c=0.5

These parameters are used in the knowledge structure induction process. They determine the minimal strength and the acceptable errors, or uncertainty, of the knowledge structure's surmise relations. Consequently, they affect the number of relations that will be induced, the overall network's topology, and the overall inference process. In general, the more relaxed these parameters are, the greater will be the number of relations. This increase on the connectivity will in turn impact on the "quantity" of inferences generated by the observations of KU, as well as on their accuracy (more inferences with less accuracy, or vice-versa).

We explored the effect of α[⋆] on the knowledge assessment performance by conducting another set of experiments with different α values. Figure 7 illustrates the results of these experiments when using different values for the α parameters and shows their respective effect on the standard error scores. In addition Table V reports the maximum and minimum size of the knowledge structures as a function of the α value. The experiments were conducted with entropy-driven sampling.

Insert figure 7
about here

The results show a clear effect of the choice of α on the performance. Although little difference is seen between α=0.5 and α=0.8, the performance for α=0.2 is significantly lower than the other two. This is explained by the very few relations induced in the knowledge structures built with α=0.2. However, in spite of the greater number of relations found in the knowledge structures with α=0.8 than the ones with α=0.5, the performance is not improved by this increase in the number of inferences. The reason is because the added relations with α=0.8 are much less reliable and result in an increase of inaccurate inferences.

This should by no means be interpreted as a general conclusion, as the impact of α and p on the standard error score reduction can be very different according to the topology of the knowledge structure, the sample size, and with the evidence propagation scheme. What these results suggest is that there is a need to adjust α and p_c according to the specific characteristics of the data set (eg. the number of subjects in the data sample and the degree of certainty desired). Further work is required to provide guiding principles on the adjustment of α and p with the characteristics of the data set.

[⋆] For simplicity, let us refer to equal values of α_i and α_c as α and assuming α_i=α_c.

TABLE V. Knowledge structures size.

Alpha	**Number of relations**	
	minimum	maximum
0.2	30	57
0.5	224	331
0.8	310	490

7. Conclusion

The experiments on knowledge assessment have shown that the POKS technique is effective in using observed KU to infer correctly an individual's knowledge state, either through the monitoring of a user's behavior, or through a selective questioning process. However, the selective questioning process, based on entropy minimization, was shown to be much more efficient in reducing the standard error score of knowledge assessment than is random sampling.

The experiments also demonstrated that the α parameter, used in building the knowledge structure, can have significant effects on the knowledge assessment performance. Indeed, it was shown that the knowledge structures built with α=0.5 and α=0.8 had clearly a better performance than the one built with α=0.2. The effects of α on the knowledge assessment performance suggest that more needs to be known about this parameter and about the p_{min} parameter. It is likely that their effect will vary with sample size and with the knowledge domain's characteristics.

In fact, we need to develop a better understanding of the reliability of the POKS approach with respect to the characteristics of the topology of the knowledge structure, and with respect to the α and p_{min} parameters, and the sample size. Moreover, an analysis of the variability of the knowledge assessment process over individual differences must also be investigated. Indeed, even if the average performance of the approach is good, it also needs to be consistent across users. Otherwise, poor performance with some users would result in frustrations on their part. Techniques for assessing the level of confidence in the knowledge assessment with regards to an individual user would thus be an important asset to the approach.

Nevertheless, in spite of these current limitations, the approach appears to be a good candidate to meet the challenges raised in the introduction. It does succeed in progressively reducing the knowledge assessment error and it is a completely automatic process that is easy to implement, as it uses simple algorithms for both knowledge structure induction and for evidence propagation. It thus reduces the process of knowledge assessment to its most fundamental element: defining the KU that correctly represent a knowledge domain. The rest of the process lends itself to automation and, hence, relieves the end-user from the burden of working on the tedious and complex details of a precise knowledge assessment procedure.

Acknowledgements

We are grateful to Cameron Hayne, Daniel Engelberg and to the three anonymous reviewers for their helpful comments on previous versions of this paper. We also whish to thank Anthony Jameson for his thorough review and helpful comments on the manuscript. This research was supported by the Computer Assisted Training grant from the Computer Research Institute of Montreal.

References

Anastasi, A. (1966). *Psychological testing*. The Macmillan Company, New York.

Benyon, D., Innocent, P., and Murray, D. (1987). System adaptivity and the modeling of stereotypes. In *Proceedings of IFIP INTERACT'87: Human-Computer Interaction*, pages 245–253.

Bretch, B. and Jones, M. (1988). Student models: The genetic graph approach. *International Journal of Man-Machine Studies*, 28(5):483–504.

Buchanan, B. G. and Shortliffe, E. H., editors (1984). *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, Reading, Mass.

Burton, R. (1982). Diagnosing bugs in a simple procedural skill. In (Sleeman and Brown, 1982).

Chin, D. (1989). KNOME: Modeling what the user knows in UC. In Kobsa, A. and Wahlster, W., editors, *User Models in Dialog Systems*, pages 74–107, Berlin, Heidelberg. Springer.

Cooper, G. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347.

De Rosis, F., Pizzutilo, S., Russo, A., Berry, D. C., and Molina, F. J. N. (1992). Modeling the user knowledge by belief networks. *User Modeling and User-Adapted Interaction*, 2(4):367–388.

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

Desmarais, M. C. and Liu, J. (1993). Exploring the applications of user-expertise assessment for intelligent interfaces. In *InterCHI'93, Bridges between worlds*, pages 308–313.

Desmarais, M. C. and Pavel, M. (1987). User knowledge evaluation: an experiment with UNIX. In Bullinger, H.-J. and Shackel, B., editors, *Proceedings of Interact'87*, pages 151–156. Elsevier Science Publisher (North Holland).

Duda, R. O., Hart, P. E., and Nilsson, N. J. (1976). Subjective Bayesian methods for rule-based inference systems. In Webber, B. L. and Nilsson, N. J., editors, *Readings in artificial intelligence*, pages 192–199. Tioga Publishing, Palo Alto, CA.

Falmagne, J.-C., Doignon, J.-P., Koppen, M., Villano, M., and Johannesen, L. (1990). Introduction to knowledge spaces: how to build, test and search them. *Psychological Review*, 97(2):201–224.

Gagné, R. (1966). *The conditions of learning*. Hold, Rinehart and Winston, New York.

Geiger, D. (1992). Proceedings of the eight conference on uncertainty in artificial intelligence. In *Fourth International Conference on User Modeling*, pages 92–97.

Giarratano, J. C. and Riley, G. (1989). *Expert systems: Principles and programming*. PWS-KENT Publishing, Boston, MA.

Goldstein, I. (1982). The genetic graph: a representation for the evolution of procedural knowledge. In (Sleeman and Brown, 1982), pages 51–77.

Gordon, J. and Shortliffe, E. H. (1984). The Dempster-Shafer theory of evidence. In Buchanan, B. G. and Shortliffe, E. H., editors, *Rule-Based Expert Systems*. Addison-Wesley, Reading, M. A.

Heckerman, D. (1991). *Probabilistic similarity networks*. The MIT Press, Cambridge, MA.

Heckerman, D. (1995). A Bayesian approach to learning causal networks. In *Uncertainty in AI: Proceedings of the Eleventh Conference*, pages 285–295.

Heckerman, D. and Geiger, D. (1995). Learning Bayesian networks: A unification for discrete and Gaussian domains. In *Uncertainty in AI: Proceedings of the Eleventh Conference*, pages 274–284.

Heckerman, D., Geiger, D., and Chickering, D. (1994). Learning Bayesian networks: The combination of knowledge and statistical data. In *Uncertainty in AI: Proceedings of the Tenth Conference*, pages 293–301.

Henrion, M. (1988). Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In *Proceedings of Uncertainty in Artificial Intelligence 4*, pages 149–163.

Kay, J. (1994). Lies, Damned Lies and Stereotypes: Pragmatic Approximations of Users. In *Proceedings of the 4th International Conference on User Modeling*, pages 175 – 184, Hyannis, MA.

Kobsa, A. (1992). Towards inferences in BGP-MS: combining modal logic and partition hierarchies for user modeling (preliminary report). In *Proceedings of the Third International Workshop on User Modeling*, pages 35–41.

Lauritzen, S. and Spiegelhalter, D. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society B*, 50:157–224.

Lukas, J. and Albert, D. (1993). Knowledge assessment based on skill assignment and psychological task analysis. In Strube, G. and Wender, K., editors, *The cognitive psychology of knowledge*. Elsevier Science.

Mislevy, R. J. and Gitomer, D. H. (1995). The role of probability-based inference in an intelligent tutoring system. *User Modeling and User-Adapted Interaction, In this issue*.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA.

Pearl, J. (1992). Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence*, 32:241–288.

Pitas, I., Milios, E., and Venetsanopoulos, A. N. (1992). A minimum entropy approach to rule learning from examples. *IEEE Transactions on Systems, Man and Cybernetics*, 22(4):621–635.

Rich, E. (1979). User Modeling via Stereotypes. *Cognitive Science*, 3(3):329 – 354.

Sleeman, D. and Brown, J., editors (1982). *Intelligent Tutoring Systems*. Academic Press, London.

Spiegelhalter, D., Lauritzen, S., and Cowell, R. (1993). Bayesian analysis in expert systems. *Statistical Science*, 8:219–282.

Takeuchi, A. and Otsuki, S. (1988). A study of student models and learner-machine interaction. In Ercoli, P. and Lewis, R., editors, *Artificial intelligence tools in education*, pages 87–104. North-Holland, Amsterdam.

Zissos, A. and Witten, I. (1985). User modeling for a computer coach: a case study. *International Journal of Man-Machine Studies*, 23:250–729.

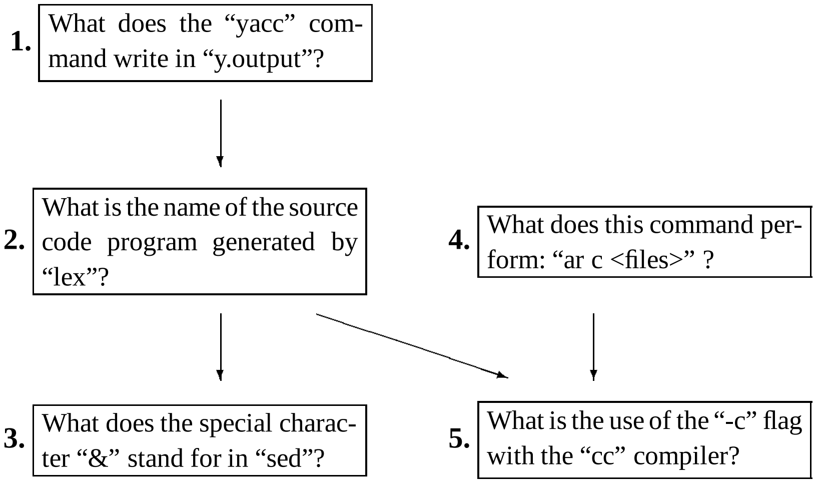
1. What does the "yacc" command write in "y.output"?
2. What is the name of the source code program generated by "lex"?
4. What does this command perform: "ar c <files>" ?
3. What does the special character "&" stand for in "sed"?
5. What is the use of the "-c" flag with the "cc" compiler?

Fig. 1. Inference network with UNIX command KU.

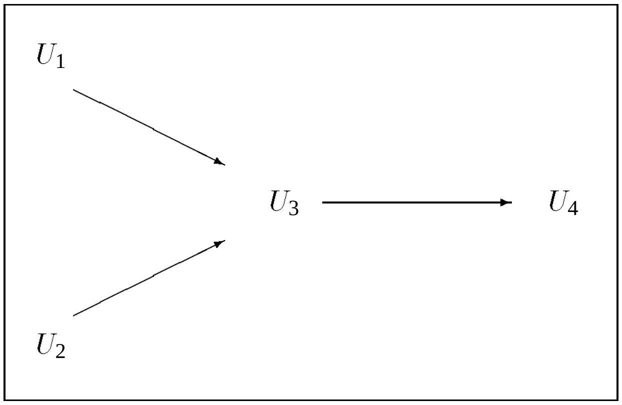

Fig. 2. Simple knowledge structure.

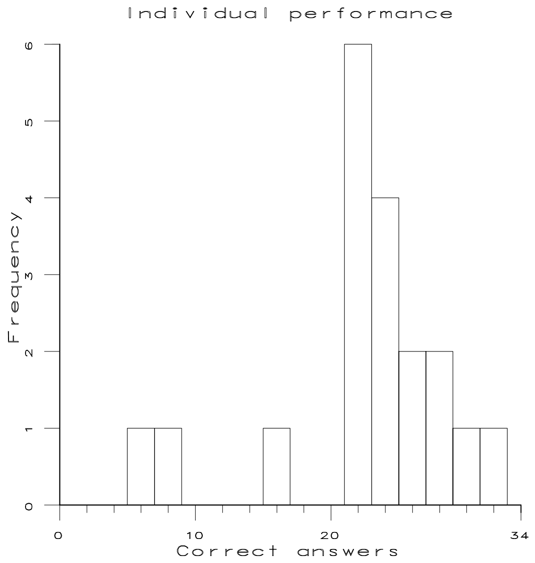

Fig. 3. Frequency count of individual performances on the 34 item questionnaire.

Question 1: What is the name of the command to rename or move a file?

(a) `move`
(b) `mv`
(c) `rn`
(d) `cp`
(e) `cat`
(f) `dd`
(g) Do not know

Question 12: What is the name of the command to list the jobs waiting to be printed?

(a) `lpq`
(b) `lprm`
(c) `lp -q`
(d) `jobs`
(e) `pq`
(f) `print -l`
(g) Do not know

Question 30: Which of the following ‘sed’ commands transforms the first line into the second one?

```
Line 1: aaa bbb ccc
Line 2: aaaxbbbxccc
```

(a) `s/ /x/g`
(b) `s/ /x/;s/ /x/`
(c) `2s/ /x/`
(d) `S/ /x/`
(e) `1,$s/ /x/`
(f) `s /x/2`
(g) `s /x/$`
(h) Do not know

Fig. 4. Some examples of test items from the UNIX questionnaires.

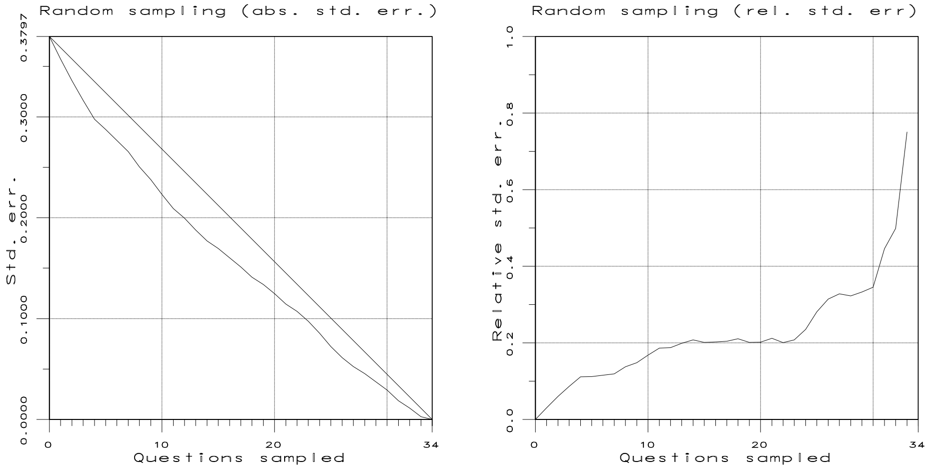

Figure 5-a

Figure 5-b

Fig. 5. Standard error score evolution. Figure 5-a represents the absolute standard error evolution (it is paired with a straight line that represents the no-inference condition and serves as a comparison) whereas 5-b is the relative standard error reduction due to the inferences.

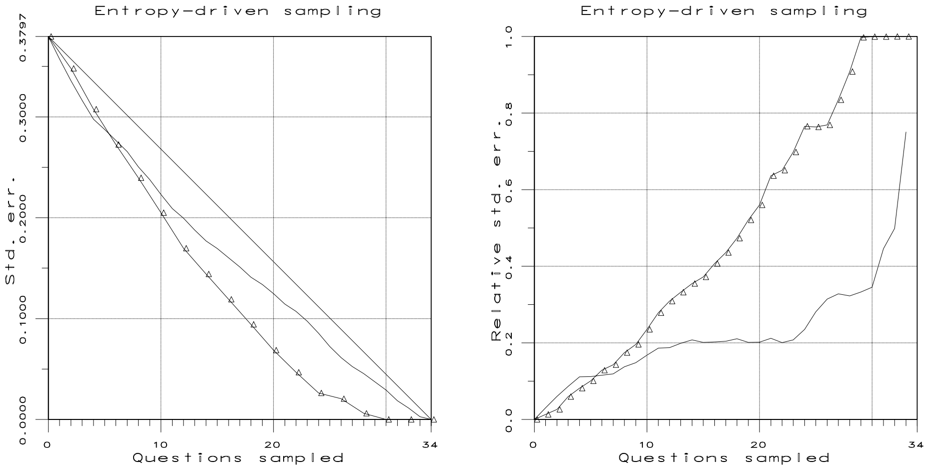

Figure 6-a

Figure 6-b

Fig. 6. Standard error evolution for the entropy-driven sampling condition (see figure 5 for explanations). The entropy-driven performance corresponds to the lines marked with triangles. The random sampling condition (full line) is reproduced in this figure for comparison purpose.

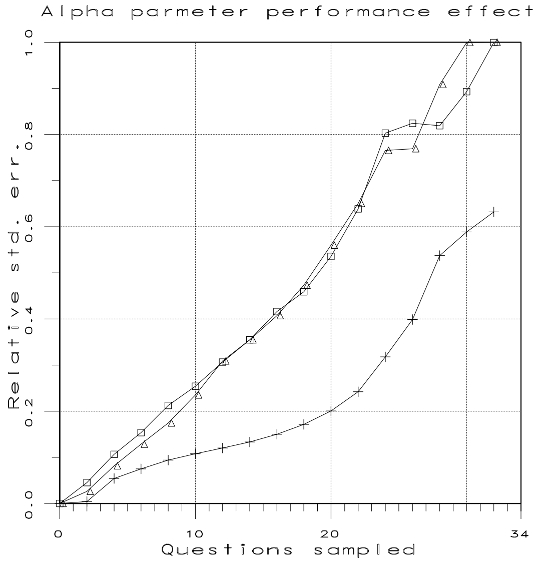

Fig. 7. The relative standard error reduction score of three knowledge structures built with three different α_i and α_c parameters: α_c=α_i=0.8 (line-marker with square), α_c=α_i=0.5 (line-marker with triangles), and α_c=α_i=0.2 (line-marker with crosses). $p_{\min}$=0.5 for all knowledge structures.

Dr. M. C. Desmarais is a researcher at the Computer Research Institute of Montreal and the scientific director of the Computer Assisted Training group at this institute. Dr. Desmarais received his Ph.D. in psychology from the University of Montreal. He has worked in the areas of cognitive psychology, artificial intelligence, and user interfaces. His most recent efforts are devoted to apply the knowledge assessment technology to training and to performance support systems applications.

Dr. A. Maluf received his BE in Electrical Engineering from the American University of Beirut in 1987, ME and Ph.D in Electrical Engineering from McGill University in 1991 and 1995 respectively. Currently he is an Advisor in the Knowledge-Base System Group at the Computer Research Institute of Montreal.

Dr. J. Liu is Assistant Professor of Computing Studies at Hong Kong Baptist University. He worked for several years as Software Engineer, Research Associate, and Senior Research Agent at both software companies research institutes in Canada, prior to joining the university in 1993. The primary themes of Liu's research work have been the studies of computational approaches to data modeling and visualization, uncertainty management and reasoning, and learning in autonomous agents. Liu received his Ph.D. in Electrical Engineering from McGill University. He is a member of IEEE and ACM.