Int. J. Human–Computer Studies (1995) **42**, 575–591

Student assessment using Bayesian nets

JOEL MARTIN

National Research Council, M-50 NRC, Ottawa, Ontario K1A 0R6, Canada

KURT VANLEHN

Learning Research and Development Center, 3939 O'Hara St., University of Pittsburgh, Pittsburgh, PA 15260, USA

We describe OLAE as an assessment tool that collects data from students solving problems in introductory college physics, analyses that data with probabilistic methods that determine what knowledge the student is using, and flexibly presents the results of analysis. For each problem, OLAE automatically creates a Bayesian net that relates knowledge, represented as first-order rules, to particular actions, such as written equations. Using the resulting Bayesian network, OLAE observes a student's behavior and computes the probabilities that the student knows and uses each of the rules.

1. Introduction

An assessment system determines what a student knows. This information is used by an assessor (i.e. the student's teacher, the student, education researchers, etc.) to make decisions. Unlike most conventional assessment (testing), which measures how much a student knows, OLAE† measures what the student knows. In particular, given a "domain model" containing a nearly complete list of the relevant pieces of knowledge in a domain, OLAE assigns a probability to each subset of the domain model that indicates how likely it is that the student knows just that subset of the domain knowledge. This detailed analysis of the student's knowledge can be aggregated in various ways appropriate for the assessor's decision making.

1.1. MODELS OF PROBLEM SOLVING

A domain model of problem solving is the union of all correct and incorrect rules (components of knowledge) that students and experts use for solving problems in a given domain. These rules are specified as logical Horn clauses and can be used with a Prolog interpreter to solve physics problems. The computational specification of the domain model distinguishes it from alternative models of expertise (e.g. Hunt & Mistrell, 1994).

The particular rules in OLAE's current domain model are sufficient to solve a

† OLAE is an acronym for On-Line Assessment of Expertise or Off-Line Assessment of Expertise, depending on how quickly assessments can be produced. The current version of OLAE does not deliver its analyses in real time and hence performs its analysis off-line.

1071-5819/95/060575 + 17$08.00/0

test suite of college physics problems and to model the problem solving behavior of a sample of students. They are adapted from the CASCADE model, which is based on substantial empirical analyses (VanLehn & Jones, 1992*a*, *b*). The rules not only produce the same answers as the students but can also model the problem solving process students engage in when answering questions in the domain. The rules in the domain model will be augmented and made more complete as problems are added to the test suite and as students are observed using novel methods, whether correct or incorrect, for solving the problems.

1.2. UNCERTAIN ASSESSMENT

Assessment is the problem of determining what a student knows. With reliable evaluations of their performance students, teachers, and school boards can improve the distribution of pedagogical resources. In OLAE, assessment produces a student model, i.e. a collection of correct and incorrect rules from the domain model are known and used by a particular student. A student model is essentially a rule-based computer program that computes answers to actual problems in the same way as does the student. OLAE uses such an approach because assessments of which rules a student uses are necessarily uncertain. There are two general situations in which a student's behavior does not determine what that student knows. First, the student may produce an answer or intermediate result that includes an unintentional slip, such as a typing error. This unintentional error may mimic the effects of some rule, resulting in an incorrect indication that the student was using that rule. Second, a student may simply be guessing, and occasionally produce correct answers. Third, when there are many ways to produce the same answer to a problem, the answer is not enough to credit the student with any specific knowledge. This can occur when there are multiple correct paths to the answer. For instance, if a physics student correctly determines the velocity of a ball just before it hits the ground, she may have done so using an equation of motion that relates distance, acceleration, and velocity or she may have used equations for calculating potential and kinetic energies.

To address this uncertainty, OLAE uses Bayesian nets. The Bayesian approach was chosen because it not only allows the ranking of hypotheses, but also acknowledges the impact of prior knowledge about relative frequencies. Early work has often found that some rules are much more common than others, sometimes by one or two orders of magnitude. For instance, the most common subtraction misconception (i.e. "always subtracting smaller digits from larger ones") occurs about 100 times more frequently than the least common ones (VanLehn, 1990). When there is a strong difference in population base rates, decisions must be based upon both those expectations and the student's behavior.

2. OLAE

OLAE collects data from a student using five computer based activities, one of which is physics problem solving. This paper will focus exclusively on the problem solving activity. The other activities are described in Martin and VanLehn (1993, in press). This section describes OLAE's input (student behavior) and output

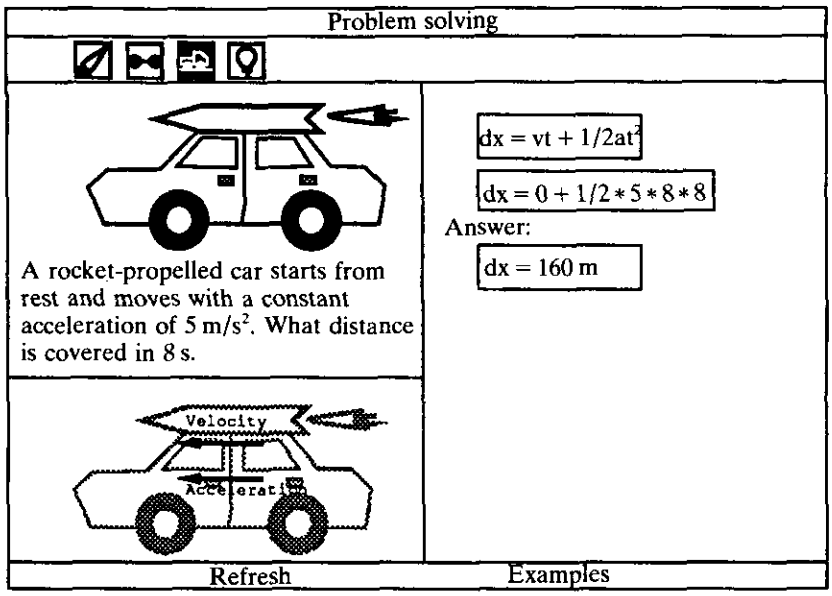

FIGURE 1. A kinematics problem presented on OLEA's problem-solving interface.

(assessment presentation), and the way that OLAE uses the behavioral data to calculate the assessments.

2.1. DATA COLLECTION AND ASSESSMENT PRESENTATION

The data about a student's problem solving behavior is collected online using a graphical interface.† The interface divides the screen into several windows (Figure 1). Along the top of the screen are icons for specific physics problems. The student selects a problem by clicking on its icon and can choose to do problems in any order.

When a problem is selected, the problem description is displayed in the upper left window. It consists of a statement of what is known, what needs to be found, and a picture of the problem situation. Right below the problem description is a copy of the picture. The student uses this window to draw axes and vectors that describe the physical situation of the problem. The window on the right is a "worksheet" in which the student is to show all their work, including intermediate equations and their final answer. OLAE records all the student's actions and the duration of those actions. All of these actions are parsed and contribute to the assessment.

Once a probabilistic assessment has been determined as described below, OLAE presents that assessment to any interested party which may include the student or a teacher. The assessment is computed at the level of rules in the domain model. However, the assessor may wish to view that assessment at different levels of abstraction. For example, a parent may not care what specific rules are missing or incorrect, but may care about overall mastery of various parts of the physics curriculum.

To allow assessors to view coarse-grained assessments, OLAE permits them to

† This interface is written in LISP using CLIM an is run on a DECstation 5133. Those interested in adapting this interface may contact the first author.

define factors. These factors are aggregations of the rule-based assessment and are similar to constructs in other assessment systems, such as "facets" in Hunt and Minstrell's (1994) DIAGNOSER. Specifically, in OLAE a factor is some function of the probabilities of a specified set of rules and represents the student's overall mastery of those rules. For example, there may be a factor called "Kinematics Mastery" that is a two-valued variable: either the student has mastered Kinematics or she has not. The factor would be defined as some imperfect conjunction of the student's knowedge of the rules necessary for solving kinematics problems.

The assessor's interface displays a network of rules and factors as shown in Figure 2. The assessor can scroll through these networks in order to get an overview of the student's competence. At any time, the assessor can zoom in on a factor. A window appears containing OLAE's assessment of the degree to which the student has mastered the factor represented as a probability distribution, a list of factors that contributed to the selected factor, and a list of factors to which the selected factor contributes.

Finally, the assessor can manipulate OLAE's assessment. For instance, if the assessor has spoken to the student and strongly believes that the student knows a

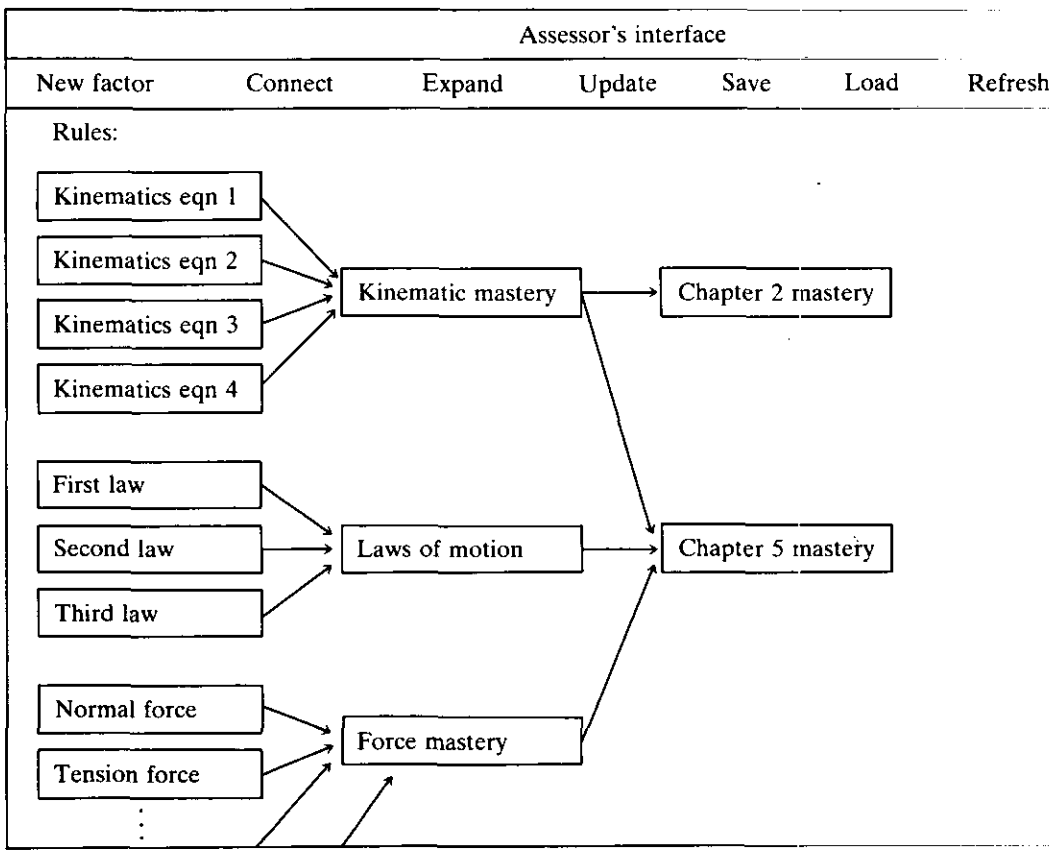

FIGURE 2. A Bayesian net on the assessor's interface. This is a compressed Bayesian network containing only *rule nodes* and *fact nodes*. Later figures will include actions and facts.

particular rule, the assessor can manually increase the probability of that rule, and OLAE will update the probabilities of other factors to reflect this new information.

2.2. PEFORMING THE ASSESSMENT

For each problem, OLAE builds a probabilistic network based on the domain model and uses this Bayesian net to analyse student behavior. A Bayesian net is a directed acyclic graph. Each node in the graph refers to a variable with two or more values. Edges in the graph specify dependencies between the values of different variables, with actual conditional probabilities associated with each node. The edges are directed to specify different patterns of conditional dependencies. For example, one variable might represent a rule and another might represent an answer to a problem. The link between those two variables would represent that the probability of the answer depends in some way on whether the rule is known. The prior probability of knowing the rule would be associated with the Rule node.

A Bayesian net is a representation for a joint probability distribution in that it can assign a probability to every possible combination of values for all the variables. In OLAE, it can assign a probability to any subset of the rule library. A Bayesian net is almost always easier and more efficient to use than a complete joint probability distribution. It achieves this efficiency through conditional independence assumptions.

In OLAE's Bayesian net, there are four types of node that represent whether or not: (a) the student knows a rule from the student model of elementary physics (*rule node*), (b) the student actually used a rule during solution of a given problem (*rule application node*), (c) the student believes a particular fact about the given problem (*fact node*), and (d) the student has performed a particular action (*action node*). *Fact nodes* include equations that the student might write. These nodes are connected by directed edges (arrows) in the net. Different paths of directed edges through the graph represent the many different paths of inferences a student might follow when solving a given problem. After the student data are used to create *action nodes* and clamp them to true, probabilistically exact (or heuristic) algorithms propagate this information along the edges to determine which rules a student probably knows.

The analysis of a student's actions is a multi-stepped process as depicted in Figure 3. It begins with the domain model and a physics problem. The OLAE domain model is a rule-based reasoner based on CASCADE (VanLehn, Jones, & Chi, 1992; VanLehn & Jones, 1993*a, b, c*), a model of physics skill acquisition. It consists of 250 rules for Newtonian mechanics and Kinematics. It includes several incorrect rules identified in earlier analyses. The domain model is applied to the current problem to produce a problem solution graph. This step of the analysis need only be done once for each problem, because it is independent of the student data.

The problem–solution graph is a large (150 nodes) directed graph that indicates all possible inferences that can be drawn from the problem's description using OLAE's rules. Whenever a rule can be applied to produce a conclusion from certain antecedents, a node is entered into the network to represent the rule application (see Figure 4). An edge is entered running from the *rule application node* to a *fact node* representing its conclusion (this node is created if it does not exist already). For each antecedent (a fact used to justify firing the rule), an edge is entered

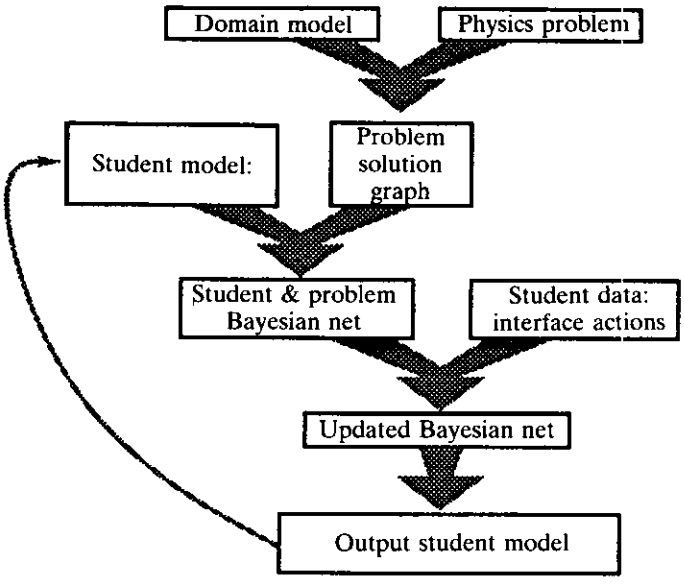

FIGURE 3. The flow of information in the OLEA system.

running from its *fact node* to the *rule application node.* An edge is also entered running from the node for the rule to the *rule application node.* If a fact has a corresponding observable action, an *action node* is created and an edge is created from the *fact node* to the *action node.* No edges leave the *action node.*

Many of the intermediate nodes in such a net are hidden variables, that is, they are not observable. These nodes represent intermediate facts (what the student believed but did not say) and rule applications (what the student did silently). OLAE assumes their existence based on the domain model of physics problem solving. These nodes are part of the problem–solution graph because they capture the probabilistic structure of the problem.

The prior probabilities that are used in the Bayesian nets in OLAE are currently set to be uniform. With data from a large number of students, these can be set

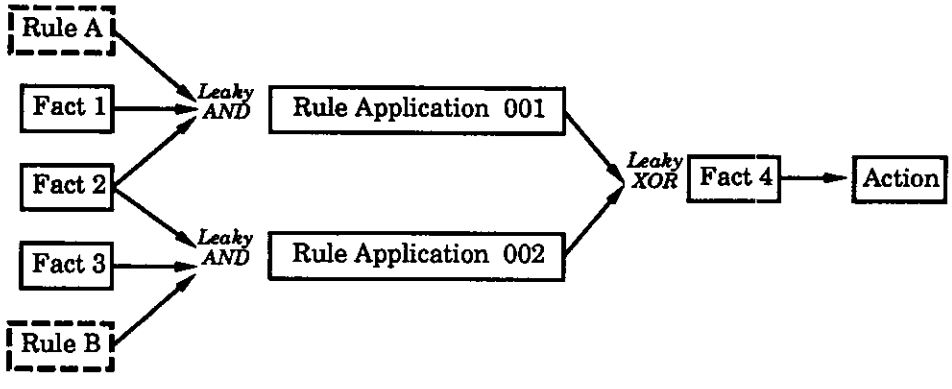

FIGURE 4. An OLEA Bayesian net: a problem–solution graph.

TABLE 1
The noisy AND relationship, numbers in parentheses assume $\epsilon = 0.05$

A	B	C = 0		C = 1	
0	0	$1-\epsilon^2$	(0.9975)	ϵ^2	(0.0025)
0	1	$1-(1-\epsilon)*\epsilon$	(0.9525)	$(1-\epsilon)*\epsilon$	(0.0475)
1	0	$1-(1-\epsilon)*\epsilon$	(0.9525)	$(1-\epsilon)*\epsilon$	(0.0475)
1	1	$2\epsilon-\epsilon^2$	(0.0975)	$(1-\epsilon)^2$	(0.9025)

empirically using the EM parameter estimation technique. The conditional probabilities used are derived from two assumptions. First, OLAE assumes that rule application is almost logical: if the rules and all its antecedents are known, then the rule will "almost always" be applied, where "almost always" is computed using a leaky-AND gate. There is a system-wide "slip" parameter (ϵ in Table 1) that determines the probability that a boolean function will be computed incorrectly. Second, OLAE assumes that people rarely infer the same fact twice. Thus, a *fact node* with multiple inputs is true only if just one input is true. This relationship is implemented with a leaky-XOR gate (Table 2).

The next step of the analysis (Figure 3) connects two Bayesian networks: the student model and the problem–solution graph. OLAE connects the two Bayesian networks by creating an edge from a *rule node* in the student model to the corresponding *rule node* in the problem–solution graph. The student model is the Bayesian network generated based on previous assessments of the student. The initial student model may have additional nodes (SM-0, SM-1, SM-2, etc) that enforce theoretical dependencies between rules. For example, if there are two very similar rules, one correct and one incorrect, those two rules would likely be mutually exclusive. A particular student would use only one of them. Any such dependencies may be encoded in the initial student model.

In order to maintain all probabilistic dependencies between rules, the student model should consist of problem solution graphs for all previous problems. However, retaining such a graph can become computationally infeasible. As a result, OLAE can summarize the dependencies in the Bayesian net. It uses either an exact dependency preserving technique (Shachter, 1986) or a heuristic technique (Martin & Billman, in press).

Once the student model for preceding problems has been connected to the

TABLE 2
The noisy XOR relationship, numbers in parentheses assume $\epsilon = 0.05$

A	B	C = 0		C = 1	
0	0	$1-\epsilon^2$	(0.9975)	ϵ^2	(0.0025)
0	1	$2\epsilon-\epsilon^2$	(0.0975)	$(1-\epsilon)^2$	(0.9025)
1	0	$2\epsilon-\epsilon^2$	(0.0975)	$(1-\epsilon)^2$	(0.9025)
1	1	$1-\epsilon^2$	(0.0975)	ϵ^2	(0.0025)

TABLE 3
Generating and using equivalency sets of physics equations

```
Once per problem-solution graph:
    Collect all facts that correspond to equations
    Form all algebraic combinations of two or more equations
    Put each equation and each combined equation in canonical form
    Store the equations in a hash table
          along with pointers to the corresponding Fact nodes

On-Line Use:
    Accept an equation from the user
    Put into canonical form
    Using the canonical equation as a key,
          retrieve Fact node pointers from the hash table
```

problem–solution graph for the current problem, OLAE is ready to process data from the interface. However, the equations from the interface do not correspond directly to nodes in the problem–solution graph. They must first be interpreted to remove algebraic variation (e.g. that $F = ma$ and $m = F/a$ are the same equation). Such translation is often needed for assessment in limited domains. The assessment is focused on one area of expertise, such as statistics, physics, or computer science, and is less interested in assessing the subskills, such as algebraic manipulation, necessary for the focus area.

Removing algebraic variation is not as easy as it may seem. Table 3 shows how the equations are interpreted, and Table 4 shows a sample set of equations that are equivalent under algebraic manipulation and variable substitution. Once for each problem, the problem–solution graph is used to determine a set of equations that might be entered by a student while solving that problem. These equations are then translated into a canonical format by moving all terms to one side of the equation, collecting like terms, sorting variables alphabetically and by power, removing common factors, etc. These canonical equations are then stored in a hashed database along with the *fact nodes* that correspond to them.

When an equation is typed on the interface, it is put into canonical form and

TABLE 4
Generating and using equivalency sets of physics equations

$$M * \cos(\theta) + N * \cos(\beta) = 0$$

$$M = \frac{-(N * \cos(\beta))}{\cos(\theta)}$$

$$M = \frac{-(40 * \cos(45))}{\cos(60)}$$

$$-56.57 = -\frac{28.28}{0.5}$$

looked up in the database and the *fact nodes* corresponding to it are retrieved.† A new *action node* is added to the problem–solution graph with incoming edges from each of the *fact nodes* that have observable consequences. The new *action node* represents the action performed by the student. It is assigned a probability of 1.0. The edges from *fact nodes* to *action nodes* implement a leaky-XOR relation. Exactly one of the possible derivations of the action can be true.

Once the problem–solution graph as been augmented with the student's actions, the new evidence is propagated across the Bayesian net. In other words, OLAE calculates the probability of every value for every node, given the new evidence. There are many sound methods for doing this propagation (Pearl, 1988). OLAE provides two methods for doing this propagation: one is fast and heuristic, and the other is slow and exact. The fast method is a form of stochastic simulation (Hrycej, 1990) and is fast enough to update probabilities in real time. However, its updating will produce probabilities with some degree of error, with the error decreasing as the amount of processing time increases. The slow and exact method used in OLAE is the Lauritzen and Spiegelhalter method (1988).

3. An example assessment

As an example, consider the simple domain model shown in Table 5. This domain model has five rules in a shorthand representation for Horn clauses. In this representation, the head of the clause is an equation and the body is a set of conditions. Any quantity in the equation in the head of the rule may be designated as the consequent of the rule at any time during rule use. All other quantities in the equation then become additional antecedents of the clause. In essence, this representation is a shorthand for several Horn clauses. The first rule in Table 5 is equivalent to three Horn clauses, each with a different quantity as the consequent.

In the sample domain model (Table 5), the first two rules concern how to project a vector onto an axis to determine a component of the magnitude. The first rule correctly states that the X component of the vector is the product of the vector's

TABLE 5
The rule library for the simple physics domain

```
mag-cos-rule:  proj(S,x)=mag(S)*cos(incline(S)).
cos-rule:      proj(S,x)=cos(incline(S)).
sum-all-rule:  net-force(Body,Axis)=sum-over(foreach:F in:Set addend
                        :proj(F,Axis))
                        :-find-all-forces(Body, Set).
sum-one-rule:  net-force(Body,Axis)=proj(F1,Axis)
                        :-find-force(Body, F1).
sum-two-rule:  net-force(Body,Axis)=proj(F1,Axis)+proj(F2,Axis)
                        :-find-force(Body, F1)
                          find-force(Body, F2)
                          F1()F2.
```

† When a student's equation is an algebraic combination of the equations from two or more *fact nodes* in the problem–solution graph, a new *fact node* and a new *action node* are created that represent that combined equation. It is implemented as a leaky-AND combination of the constituent *fact nodes*. This new *fact node* is then connected to the new *action node*.

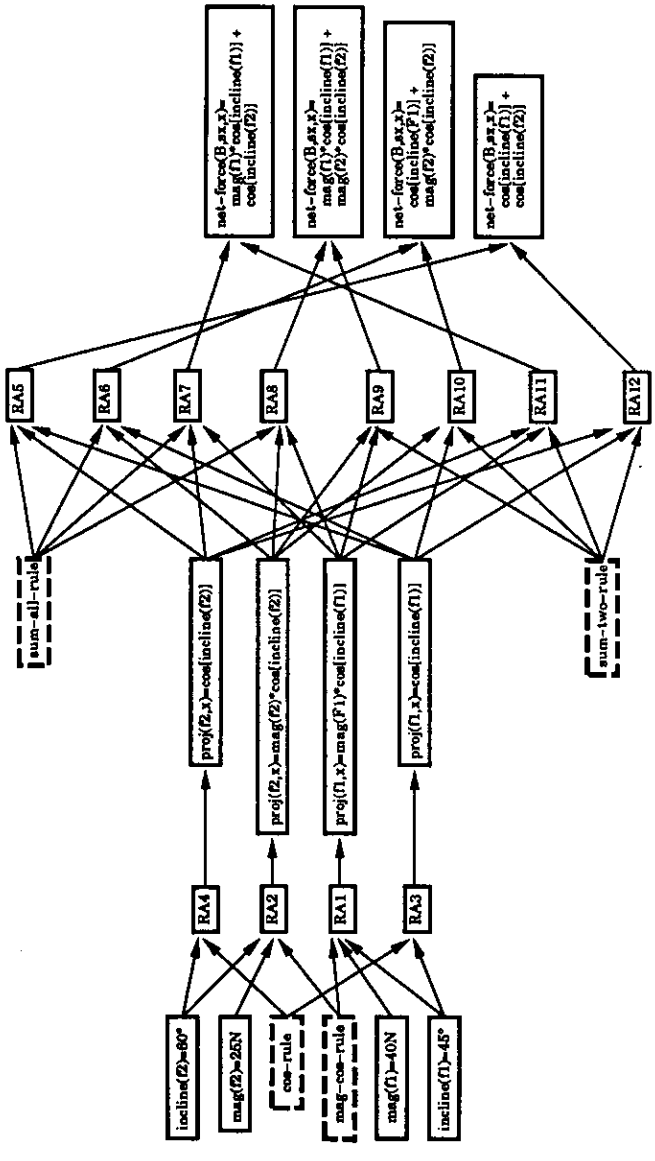

FIGURE 5. A Bayesian net created for all four rules in Table 5 in the context of a particular problem. This is a problem–solution graph consisting of *rule nodes*, *fact nodes* and *rule application nodes*.

TABLE 6
A simple Newtonian physics problem of combining multiple forces acting on one physical body

Two forces act on a point object as follows: 40 N at 45° and 25 N at 60° Find the magnitude of the resulting force.

magnitude and the cosine of its angle with the x-axis. The second rule incorrectly omits the influence of the vector's magnitude. The last three rules describe how the same axis components of vectors can be combined along an axis. The third rule correctly states that the resulting magnitude on the axis is the sum over the same axis components from all forces acting on the physical body. The fourth and fifth rules are overly specific and are only correct when there are just one or two forces (respectively) acting on the body.

For the problem shown in Table 6, the first step of the analysis generates a Bayesian net like that shown in Figure 5. The rules are shown in dashed boxes and the given parts of the problem are on the left side of the figure. The rules are labeled with their rule-names. The possible actions are on the right side and represent various interpreted equations that a student could type. Here they are represented with no substitutions. The intermediate nodes represent rule applications (labeled "RA") and intermediate conclusions.

Suppose that a particular student is new to OLAE and has a default student model (Figure 6) with all rules having some arbitrary probability (here, 0.33). As noted above, the initial student model may have additional nodes (SM-0, SM-1, SM-2, etc) that enforce theoretical dependencies between rules. For example, if there are two very similar rules, one correct and one incorrect, those two rules would likely be mutually exclusive. A particular student would use only one of them. In Figure 6, the first two rules are mutually exclusive as are the third and fourth, and the third and fifth. When the student is presented with the problem in Table 6, the

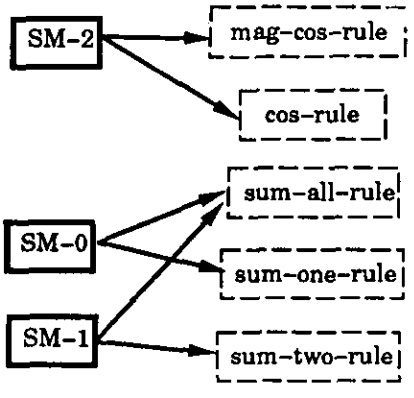

FIGURE 6. The initial state of the student model for the rules in Table 5. It represents the prior probabilities on the rules, including the fact that the first two rules are mutually exclusive, as are the third and fourth, and the third and fifth.

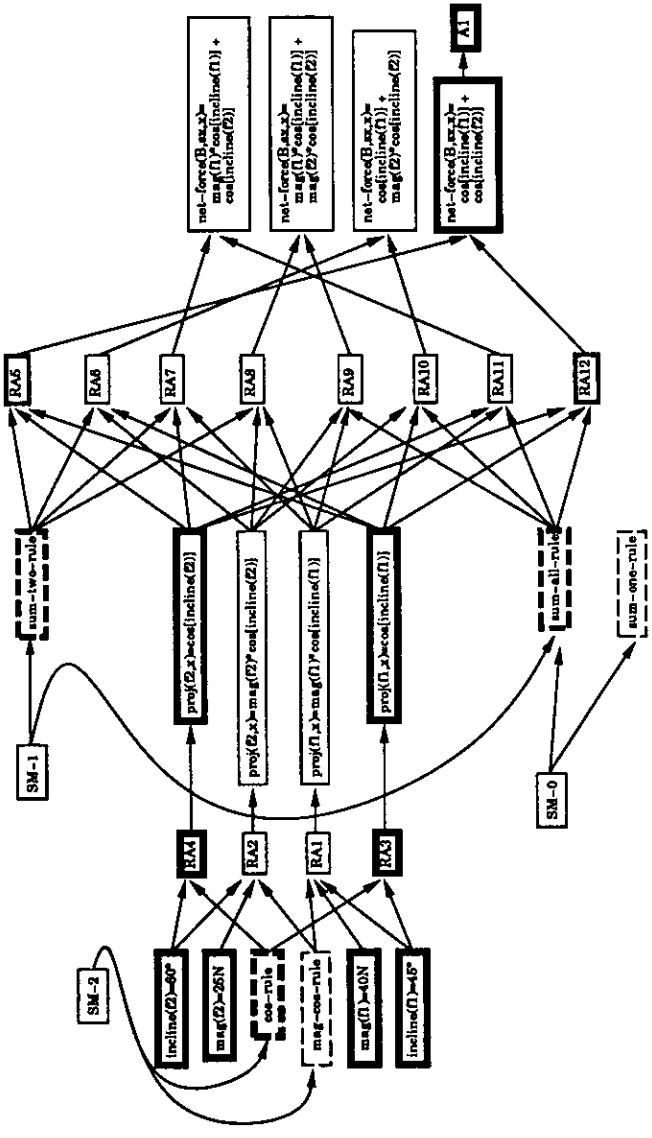

FIGURE 7. The state of the student model for the rules in Table 5 after the first problem.

student model is connected to the problem solution graph producing the Bayesian net in Figure 7.

Suppose that the student writes the equation, $F = \cos(45) + \cos(60)$. This equation is put into canonical form. It is used as a key to the database of possible equations and it matches an equation corresponding to the node labeled "net force $(B, sx, x) = \cos[\text{incline}(f1)] + \cos[\text{incline}(f2)]$" (bottom-most node in the last column of Figure 7). A new node is created to represent the action and a deterministic link is created from the equation node to the new *action node.* The new equation node is clamped to a probability of 1.0, and this new information is propagated backward to the rules. The rule labeled "cos-rule" in Figure 7, has its probability raised from 0.33 to 0.94. In this case, the student is almost certainly using the incorrect rule, "cos-rule" and not the correct rule "mag-cos-rule" ($p = 0.03$). The rules "sum-two-rule" and the "sum-all-rule" are equally likely with a probability of 0.49, because the observed equation could have been generated by either. Finally, the probability of the rule "sum-one-rule" decreases from 0.33 to 0.228. After updating the probabilities, the whole network will be carried along as the student model. As noted earlier, OLAE can compact the Bayesian nets when they grow too large.

In Figures 6, 7 and 8 the nodes that are highly probable (0.8–1.0) have a thick border, the nodes that are moderately probable (0.5–0.8) have a medium border, and the remaining nodes are low probability.

Suppose a second problem is analysed and attached to the network (Figure 8). In

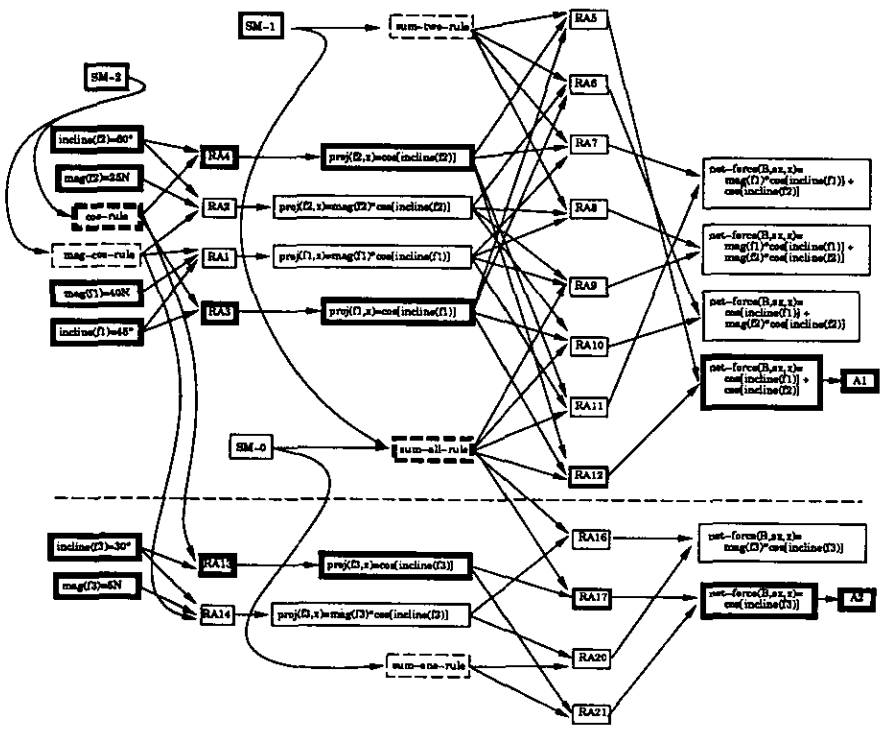

FIGURE 8. The state of the student model for the rules in Table 5 after the second problem.

this case, the second problem was a simpler, related problem in which the student only had to compute the x-component of a single force. Again, student actions for this problem are matched (via canonicalization) to the equation *fact nodes,* a new *action node* is created, that node is clamped to a probability of 1.0 and the net's probabilities are updated.

Suppose that the student typed the equation, $F = \cos(75)$. Now, the probability that the "cos-rule" is known has increased from 0.94 to 0.998. The probability that the "sum-all-rule" is known has also increased from 0.49 to 0.77 because it is consistent with both observations. The probability of the "sum-two-rule" has decreased to 0.206 and the probability of the "sum-one-rule" has not changed. Furthermore, the overall Bayesian net now represents the fact that the student either has the "sum-all-rule" or both the "sum-one-rule" and the "sum-two-rule". If OLAE later sees evidence to discredit "sum-all-rule", then it immediately has a strong belief in both the other rules. After analysing the second problem, OLAE uses the whole Bayesian net as the new student model, and the process repeats once for every remaining problem.

With real problems and a more complete domain model, a problem–solution graph for a single physics problem has between 100 and 200 nodes. We have compared OLAE's assessment for two such problems to assessments performed by a human from verbal protocols.† In all cases where the human assessor determined that the student knew a rule, OLAE had assigned a probability of greater than 0.85 to that rule. Similarly, in all cases where the human assessor determined that the student did not know a rule, OLAE had assigned a probability of less than 0.15.

4. Discussion

OLAE is an assessment tool like the student modeling component of many intelligent tutoring systems. However, it uses a normative (or defensible) approach to managing uncertainty. OLAE uses such an approach because highly accurate assessments can improve the distribution of pedagogical resources. Students, teachers, and school boards can have objective, reliable evaluations of their performance and can change their efforts accordingly.

A few other student modeling efforts have used Bayesian networks as a basis for assessment (Villano, 1992; Sime, 1993; Petrushin & Sinitsa, 1993; Gitomer, Steinberg, & Mislevy, 1994). Unlike OLAE though, all of these efforts involve non-automatic generation of the Bayesian nets, i.e. a human must construct the network for each problem. In addition, they do not use a computationally sufficient cognitive model of the problem solving involved. As one example, Villano (1992) represents expertise as the ability to solve specific problems rather than mastery of several components of knowledge. As a result, Villano's assessments cannot communicate precisely what a student does not know and cannot identify the components of knowledge that must be taught.

OLAE also differs from other student modeling efforts using Bayesian networks in that it is specifically tailored to scientific domains. It provides techniques for interpreting actions that are equations and related diagrams.

† These protocols were collected and analysed before OLAE was constructed.

Finally, the way OLAE generates its Bayesian network is similar in spirit to Poole's (1992) methods for the representation and use of probabilistic Horn clauses. OLAE's approach could be implemented using Poole's scheme or those proposed by Breese (1992) and Charniak and Goldman (1989).

4.1. FUTURE WORK

One component of OLAE requiring future work is the interpretation and integration of the additional tasks to complement the assessment obtained from problem solving behavior. Physics students also naturally engage in other behavior that can be diagnostic of what the student knows and indeed what they learn through study. For instance, when students study worked out examples, they spend differing amounts of time on different parts of the solution and show distinct patterns of referring back to previous parts of the solution. We have implemented an interface to capture this information from students as they study and have determined that the information gleaned about a student is relevant to study habits such as the use of self explanation (Chi, Bassok, Lewis, Reimann & Glaser, 1989).

Besides integrating different tasks, we also plan a more extensive evaluation of OLAE. Currently, OLAE's assessment has been verified against two hand-done protocol analyses. We plan to extend these analyses first by comparing OLAE's assessment to a human tutor's assessment and comparing both of these to a gold standard, i.e. a line by line fine-grained assessment verbal protocols. This study will measure the external validity of OLAE. Subsequently, we will evaluate OLAE's internal validity by measuring its success at correctly interpreting artificially created student data. For these assessments, we will create artificial students that each know some random assortment of rules for physics. Then CASCADE, the cognitive model of physics problem solving, will solve a variety of problems using the artificial student's knowledge base. OLAE's task is then to work backwards from problem solving actions to infer what rules the artificial student knows.

Although OLAE is a very general assessment tool for equation-based domains, there are two additional characteristics that would increase its utility for practical applications. First, OLAE should recognize the possibility of partially known rules to enhance its ability to model rule-learning. Currently, OLAE assumes that students either know or do not know a rule. It also assumes that if the student knows a rule and that rule can be used, it will be used. This is not very realistic. As a rule is being learned or forgotten, it may be used only sporadically. In such cases, the rule is somewhere between known and not known.

A second enhancement to OLAE would allow the automatic revision or wholesale learning of cognitive domain models. This would permit automatic acquisition of incorrect rules as well as accurate rules. Currently, the practical use of OLAE hinges on the existence of an appropriate, empirically justified model of the domain. Producing such a model is a difficult task. If it can be generated automatically using a large data sample, OLAE would be immensely more useful.

To do this, OLAE could use existing algorithms (Quinlan, 1990; Pazzani & Kibler, 1992) to extract a set of Horn clause rules sufficient to produce all the actions in a large collection of existing data. However, most of these are restricted to very small first-order theories. Therefore, learning large theories, on the order of our physics

knowledge, would be very time consuming. Indeed, similar attempts at learning cognitive domain models have become bogged down in the combinatorial explosion of possible hypotheses (Langley & Ohlsson, 1984; Kowalski & VanLehn, 1988).

The authors wish to thank the anonymous reviewers for assistance with an earlier draft of this paper. The research reported in this paper was sponsored by the Cognitive Science division of the Office of Naval Research under grant number, N00014-91-J-1532.

References

BREESE, J. S. (1992). Construction of belief and decision networks. *Computational Intelligence,* **8,** 624–648.

BURTON, R. R. (1982). Diagnosing bugs in a simple procedural skill. In D. SLEEMAN & J. S. BROWN (Eds.) *Intelligent Tutoring Systems.* New York: Academic Press.

CHARNIAK, E. & GOLDMAN, R. (1989). A semantics for probabilistic quantifier-free first-order languages, with particular application to story understanding. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence,* pp. 1074–1079. Detroit, MI: Morgan-Kaufmann.

CHI, M. T. H., FELTOVITCH, P. & GLASER, R. (1981). Categorization and representation of physics problems in novices and experts. *Cognitive Science,* **5,** 121–152.

GITOMER, D. H., STEINBERG, L. S. & MISLEVY, R. J. (1994). Diagnostic assessment of troubleshooting skill in an intelligent tutoring system. In P. NICHOLS, S. CHIPMAN & S. BRENNAN, Eds. *Cognitive Diagnostic Assessment.* Hillsdale, NJ: Lawrence Erlbaum.

HRYCEJ, T. (1990). Gibbs sampling in Bayesian networks. *Artificial Intelligence,* **46,** 351–363.

HUNT, E. & MINSTRELL, J (1994). A cognitive approach to the teaching of physics. In K. MCGILLY, Ed. *Classroom Lessons: Integrating Cognitive Theory and Classroom Practice.* pp. 51–74. Cambridge, MA: MIT Press.

KOWALSKI, B. & VANLEHN, K. (1988) Cirrus: inducing subject models from protocol data. In V. PATEL, Ed. *Proceedings of the Tenth Annual Conference of the Cognitive Science Society,* pp. 623–629. Hillsdale, NJ: Erlbaum.

LAURITZEN, S. L. & SPIEGELHALTER, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society,* **50,** 157–224.

LANGLEY, P. & OHLSSON, S. (1984). Automated cognitive modeling. In *Proceedings of the National Conference on Artificial Intelligence,* pp. 193–197. Austin, TX: Morgan-Kaufmann.

MARTIN, J. D. & BILLMAN, D. O. (in press). The acquisition and use of overlapping concepts. *Machine Learning.*

MARTIN, J. D. & VANLEHN, K. (1993). OLAE: Progress toward a multi-activity, Bayesian student modeler. In (P. BRNA, S. OHLSSON, H. PAIN, Eds.), *Proceedings of the 1993 World Conference on AI and Education.* Charlottesville, NC: Association for the Advancement of Computing in Education. pp. 410–417.

MARTIN, J. D. & VANLEHN, K. (in press). A Bayesian approach to cognitive diagnosis. To appear in (CHIPMAN, S. Ed.) *Cognitive Diagnosis.* Hillsdale, NJ: Lawrence Erlbaum.

PAZZANI, M. & KIBLER, D. (1992). The utility of prior knowledge in inductive learning. *Machine Learning* **9,** 57–94.

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.), *Proceedings of the 1993 World Conference on AI and Education.* Charlottesville, NC: Association for the Advancement of Computing in Education. pp. 426–432.

POOLE, D (1992). Probabilistic Horn abduction and Bayesian networks. Technical report 92-2, University of British Columbia. Department of Computer Science.

QUINLAN, J. R. (1990). Learning logical definitions from relations. *Machine Learning* **5,** 239–266.

SHACHTER, R. D. (1986). Evaluating influence diagrams. *Operations Research* **34,** 871–882.

SIME, J. (1993). Modeling a learner's multiple models with Bayesian Belief nets. In (P. BRNA, S. OHLSSON, H. PAIN, Eds.), *Proceedings of the 1993 World Conference on AI and Education.* Charlottesville, NC: Association for the Advancement of Computing in Education. pp. 426–432.

VANLEHN, K. (1990) *Mind Bugs: The Origins of Procedural Misconceptions.* Cambridge, MA: MIT Press.

VANLEHN, K. & JONES, R. M. (1993*a*). Learning by explaining examples to oneself: A computational model. In (A. MEYROWITZ & S. CHIPMAN, Eds.), *Cognitive Models of Complex Learning.* Boston, MA: Kluwer Academic.

VANLEHN, K. & JONES, R. M. (1993*b*). What mediates the self-explanation effect? Knowledge gaps, schemas or analogies? In (M. POLSON, Ed.), *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Lawrence Erlbaum.

VANLEHN, K. & JONES, R. M. (1993*c*). Better learners use analogical problem solving sparingly. In P. UTGOFF (Ed.), *Proceeding of the Tenth International Workshop on Machine Learning,* pp. 338–345. San Mateo, CA: Morgan Kaufmann.

VANLEHN, K., JONES, R. M. & CHI, M. T. H. (1992). A model of the self-explanation effect. *Journal of the Learning Sciences* **2,** 1–60.

VILLANO, M. (1992). Probabilistic student models: Bayesian belief networks and knowledge space theory. *Proceedings of the 2nd International Conference on Intelligent Tutoring Systems, Lecture Notes in Computer Science, 608.* Berlin: Springer-Verlag.