OLAE: Progress toward a multi-activity, Bayesian student modeler

Joel Martin
Kurt VanLehn
Learning Research and Development Center
3939 O'Hara St.
University of Pittsburgh
Pittsburgh, PA 15260 USA
(412) 624-0843
martin@cs.pitt.edu, vanlehn@cs.pitt.edu

Topic: Assessment/Student Modeling
Subtopic: Bayesian Models
Other Keywords: Physics, Bayesian Networks.

Abstract

Assessment is a necessary part of education and training. Student modeling is an assessment method that avoids some of the problems of conventional tests. However, there are many obstacles to effective student modeling. In this paper, we address three. A student modeling system should: (a) analyze data in a statistically sound, defensible manner, (b) augment data on a person's performance while they work with data from other tasks, and (c) provide assessments at multiple grain sizes. We present OLAE, a computer assistant to a human assessor that collects data about problem solving in elementary physics, analyzes that data with sound, probabilistic methods, and flexibly presents the results of analysis.

Like it or not, assessment is a part of education and training. Decisions must be made about students, teachers, curricula, and schools. Making these decisions intelligently ultimately requires knowing what a student has learned. This amounts to forming a model of the student's knowledge.

This model may represent a student's knowledge at varying levels of detail. Fine-grained models indicate whether a student has mastered a particular rule, concept or subskill, whereas large-grained models indicate whether a student has mastered a more comprehensive piece of knowledge, say, a particular chapter in a textbook.

In general, a fine-grained student model is needed for decisions that only affect the student for a short time, such as deciding which exercise to pick next or how to explain a difficult concept. Large-grained models are needed for decisions that affect the student for longer periods, such as deciding which physics course to take or which summer job to take.

Student modeling is a form of assessment that was developed to help computer-based tutoring systems decide which exercise to give to a student, when to interrupt, what level of explanation to give, and so on. This technology can also aid human decision makers who would otherwise rely on traditional assessment instruments, such as multiple-choice tests, which are sometimes inadequate (Collins, 1990). For example, multiple choice tests are often not representative of real work in a subject area where solving a problem can take hours or days. Ideally, the assessor would observe a person as they worked and over time obtain an accurate model of that person's strengths and weaknesses. This is exactly what student modeling systems are designed to do.

However, significant problems impede the adoption of student modeling as a bona fide assessment technology. We have identified three, and are building a system that embeds solutions to all of them.

Most student modelers, even the best ones (e.g., Burton's Debuggy, 1982), use heuristics to analyze the raw behavioral data. However, when important decisions must be made on the basis of assessments, heuristics should be minimized. One must be able to defend the conclusions of the student modeler to students, employees, employers and others. Testing organizations are often taken to court, and their analyses must meet or exceed the standards used by scientists in interpreting their data. Thus, building a student model that uses a statistically sound, defensible data analysis is our first goal for the system, which we call OLAE.[1] Our solution is to use Bayesian networks for the data analysis. This technology, which was originally developed for medical diagnosis, is statistically sound.

The second problem is that merely monitoring a person's performance while they work does not always provide enough data. Often, the most interesting thinking occurs when the person is silent and motionless. For instance, when solving a problem, people often pause for a while in order to mentally represent or idealize the problem and to formulate a solution plan. Often it is impossible to infer what they are thinking from their observable behavior, even if they are asked to talk aloud as they go. Our solution to this problem is to include several tasks from the expert-novice literature that have been shown to be sensitive to the problem-representing and solution-planning expertise of subjects. The data from these tasks is integrated with performance data from ordinary tasks in order to form a more accurate assessment of the person's knowledge.

The third problem is that assessments are needed at multiple grain sizes in order to support decision making of many kinds. A curriculum developer might want to see exactly which rules and concepts a subject mastered in order to determine if her instruction was effective. A student might want to see if she understands chapter 5 well enough to go on to chapter 6. A commander for a satellite tracking station might want to find someone who knows celestial mechanics or could learn it quickly. Our solution to this problem is to conduct the assessment at the finest grain size that we think the users will want, and provide simple graphical tools for users to build coarser assessments on top of these fine-grained ones.

The initial version of OLAE is being developed to address these three problems with college physics as the task domain. Physics was chosen for three reasons. First, many physics educators feel that traditional assessment instruments overrate student's understanding in physics (e.g., Hestenes, Wells, & Swackhamer, 1992). We expect OLAE to provide a more accurate assessment than traditional instruments, and in particular, to reveal how students can correctly answer traditional questions while having serious knowledge

[1] OLAE is an acronym for On-Line Assessment of Expertise, because from the point of view of traditional paper-and-pencil based testing, OLAE looks like an on-line system. However, OLAE does not deliver its analyses in real time. The students' behavior is recorded and analyzed later. Thus, from the student modeling point of view, the system does off-line data analysis. Fortunately, the same acronym works for both on-line and off-line assessment of expertise.

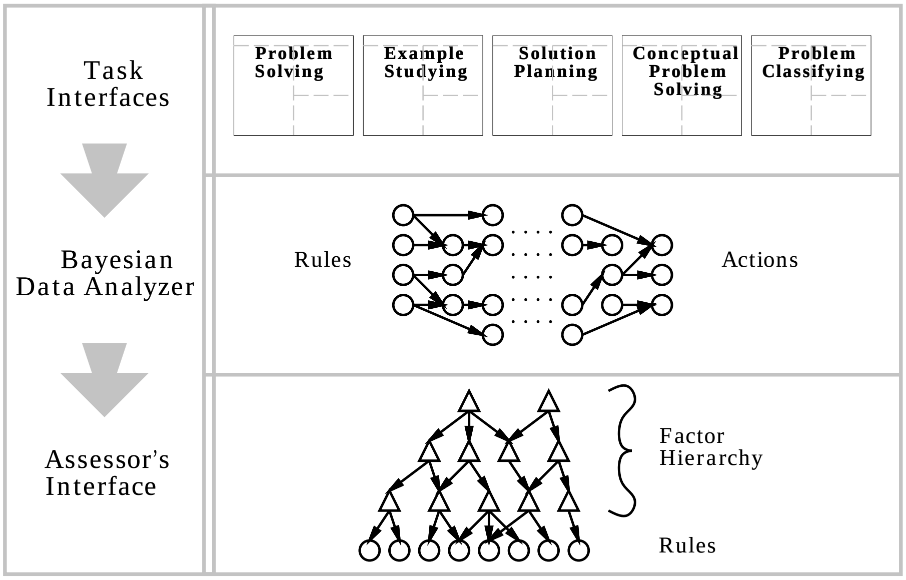

Figure 1: An overview of the three components of OLAE: Interface Tasks, Bayesian data analysis, and an Assessor's interface.

deficits. Second, physics was chosen because it has both procedural and conceptual content. Third, physics was chosen because many of the expert-novice studies were conducted with physics as their task domain. Using physics as the initial task domain for OLAE reduces the risk in adopting these expert-novice tasks as measures of expertise.

The remainder of this paper describes our design for OLAE, its current status and our plans for developing it further. We conclude with an assessment of whether our three goals have been met.

1 The OLAE system

OLAE has three component systems: (1) a set of task interfaces, (2) a Bayesian data analyzer, and (3) an assessor's interface (Figure 1). The task interfaces gather detailed process data from a student engaged in two authentic physics activities (solving problems and studying worked examples), and from three expert-novice tasks. The Bayesian data analyzer then integrates and analyzes these data using defensible, non-heuristic probabilistic algorithms and does so with reference to a well-supported cognitive model of elementary physics expertise. Finally, an assessor's interface makes these analyses available to the human assessor in a graphic form and at multiple levels of detail in order to facilitate informed decision making.

1.1 Data Collection: Task Interfaces

OLAE has task interfaces for each of the following tasks: problem solving, example studying, solution planning, conceptual problem solving, and problem classification. Each will be discussed in turn.

To assess problem solving knowledge, the student answers problems on a computer interface. The computer screen is divided into several windows (Figure 2). Along the top are icons for specific physics problems. The student selects a problem by clicking on its icon.

When a problem is selected, the problem description is displayed in the upper right window. It consists of a statement of what is known and what needs to be found as well as a picture of the problem situation. Right below the problem description is a copy of the picture. Students can draw axes and vectors on this picture, both to help them solve the problem and to demonstrate their knowledge of physics. The students enter equations in the window on the left. They are told to type everything necessary to solve the problem including side calculations and scratch work.

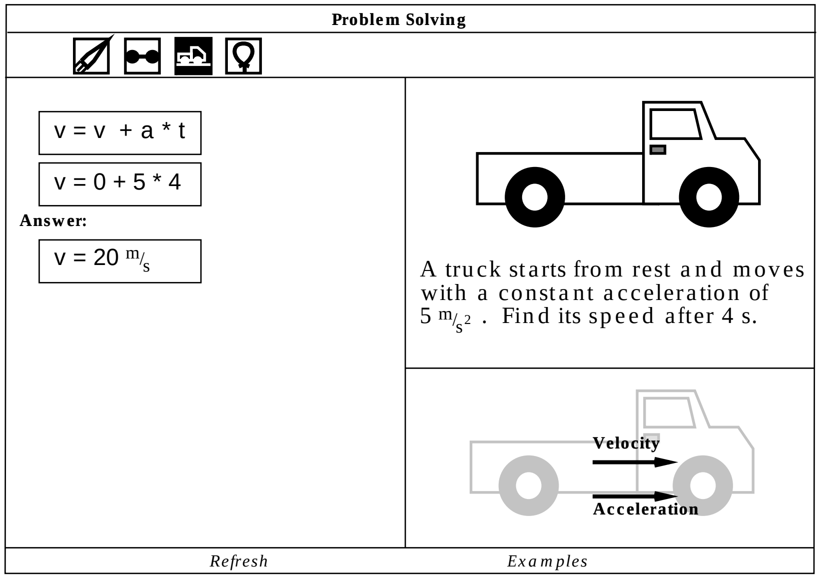

Figure 2: A kinematics problem presented on OLAE's problem-solving interface.

It may happen that as they are solving a problem, they won't know what to do. However, they may remember an example (i.e., a correctly solved problem) that might be helpful. From the problem solving interface, they can refer back to these examples. Clicking on the word *Examples* on the bottom of the screen changes the problem solving interface to an example interface.

The example interface is organized in the same way as is the problem solving interface. In the top window, there is again a list of icons, and the student can select an example by clicking on its icon. The windows have the same information as in the problem solving interface, but the information is hidden until requested. Each equation in the left window is hidden by a shaded box. Boxes also hide the force diagram in the lower right window and the problem description in the upper right window. As the mouse arrow moves over a box, the box opens to reveal that part of the solution to the problem. The student can slowly step through the solution, opening one box at a time. This part of the example-studying interface is called the *poor man's eye-tracker*. It tells OLAE what the student is reading and for how long.

As noted, these interfaces cannot collect information about periods of inactivity that occur, for example when the students are planning their solution. In order to collect additional information, OLAE has three more task interfaces.

- *Solution planning.* The third task interface presents several distinct activities at the same time. Given a problem, the student is asked to list aspects of the problem that make it difficult, to describe their basic approach to solving it, and to estimate its degree of difficulty. All three measures are sensitive to the general level of expertise (Chi, Glaser, & Rees, 1982).

- *Conceptual problem solving.* The fourth task interface asks students to solve "conceptual" problems. In task domains such as physics, whose problems require algebraic or quantitative solutions, it is often possible to pose problems that require no higher mathematics and yet are still challenging. Performance on these problems is also sensitive to level of expertise (e.g., Hestenes, Wells & Swackhamer, 1992).

- *Problem classification.* The fifth task interface asks subjects to sort problems into piles so that all "similar" problems are in the same pile and asks them to label the piles. Experts tend to classify problems according to their solutions while novices tend to classify problems according to the problem statements (e.g., Chi, Feltovitch, & Glaser, 1981).

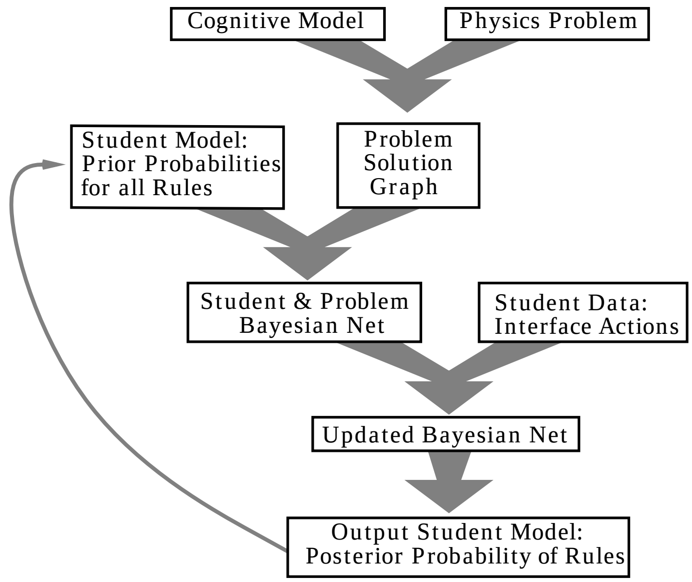

Figure 3: A simple diagram of the flow of data as OLAE updates its model of a student.

1.2 Data Analysis using A Cognitive Model and Bayesian Inference

The Bayesian data analyzer in OLAE builds a probabilistic network based on a cognitive model of elementary physics and uses this Bayesian net to analyze the student data. 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 the conditional probabilities between the values of different variables. A Bayesian net is a convenient representation for probabilistic models. It expresses a joint probability distribution in that it can assign a probability to every possible combination of values. However, it is almost always easier and more efficient to use than a complete joint probability distribution.

We have implemented data analysis modules to analyze data from the problem solving and example studying interfaces. In this paper, we present only the module for analyzing problem solving data.

In OLAE's Bayesian net, there are four types of binary (true, false) nodes that represent whether or not: (a) the student knows a rule from the cognitive model of elementary physics, (b) the student actually used a rule during solution of a given problem, (c) the student believes a particular fact about the given problem, or (d) the student performed a particular action while solving the given problem. These nodes are connected by directed edges (arrows) in the net. Roughly speaking, the edges reflect the many different paths a student might take to solve a given problem. After the student data recorded by the task interfaces is used to set the probability of the nodes that correspond to actions, non-heuristic algorithms propagate this information along the edges to determine which rules a student probably knows.

The data analysis is a multi-stepped process depicted in Figure 3. It begins with the cognitive model and a physics problem. The OLAE cognitive model is a rule-based reasoner based on work with CASCADE (e.g., VanLehn, Jones, & Chi, 1992), a model of physics skill acquisition. We have not yet included many incorrect rules. Earlier analyses (VanLehn & Jones, in press; in prep.) indicate that although there were many cases of missing knowledge causing errors, buggy knowledge was not all that common, at least among the 9 students analyzed. If this trend holds, then it should not be too much trouble to find and encode enough buggy rules for a reasonable coverage of the data.

The set of rules is combined with the current problem to produce a problem-solution graph. The problem-solution graph is a huge directed graph that indicates all possible inferences that can be drawn from the problem's description using OLAE's rules. Whenever a rule can apply 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 that node to a node representing its conclusion (this node is created if it does not exist already). For each antecedent, an edge is entered running from it to the rule application node. An edge is also entered running from the node for the rule to the rule application node. This step of the analysis need

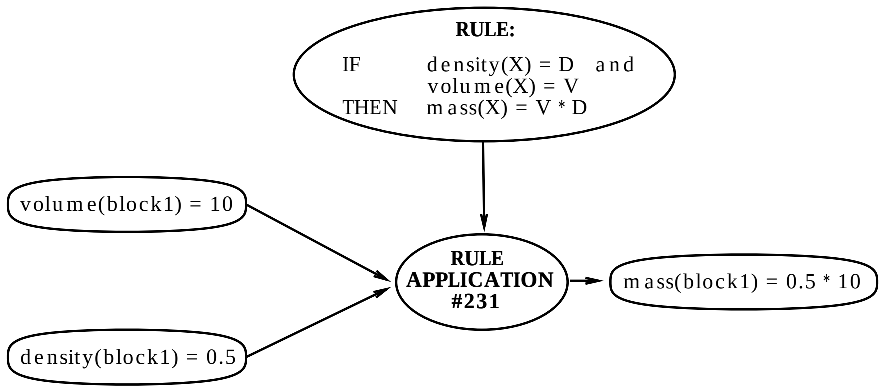

Figure 4: A portion of a Problem-Solution graph illustrating the connections between rules, literals, and rule applications.

only be done once for each problem because it is only dependent on the cognitive model and the problem.

The next step of the data analysis (Figure 3) enters the prior probabilities of the rules for a given student into the corresponding nodes of the Bayesian net. By doing so, the problem-solution graph is personalized for the given student.

OLAE is now ready to accept data from the problem solving task interface. The actions from the interface correspond to leaf nodes (those with no outgoing edges) in the net. The nodes corresponding to actions that were observed on the interface are *clamped*. This means that because the action was observed, the occurrence of the action has a probability of 1.

Once the action nodes are clamped, the new evidence is propagated across the 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). In the final step of analysis, the probabilities of the rule nodes are read out of the Bayesian net and become the updated student model.[2]

As an example, for the problem shown in Figure 2, the first and second steps of the data analysis generate a Bayesian net like that shown in Figure 5. This net is simplified for illustrative purposes. The rules and 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 equations that a student could type. Here they are labeled as either 'correct' or 'wrong' depending on whether they are components of the correct solution to the problem. The intermediate nodes (black squares) represent rule applications and intermediate conclusions.

Suppose that a particular student has a student model with all rules equally likely at a probability of 0.5. Next suppose that the two incorrect actions marked with triangles are observed. These nodes are clamped to a probability of 1, then these probabilities are propagated backward to the rules. Two rules, labeled '$x = vt + 1/2at^2$' and 'use-equation*' in Figure 5, have their probabilities raised to 0.657 and 0.746 respectively. In this case, the student is probably using the incorrect rule: 'use-equation*'.

1.3 Model presentation: Assessor's interface

Although we cannot anticipate all decisions a physics instructor might want to make, we expect that most will be served by a grain size at which rules are a little more finely grained than a list of basic physics principles, such as conservation of momentum or the definition of potential energy. For instance, each of Newton's three laws of motion should be a rule, but we need not represent all the different algebraic variants of $F = ma$. We assume that the assessors will primarily be interested in a person's physics competence, not their algebraic competence.

Further, we assume that assessors will not always want a fine-grained, rule-level assessment of the student, so OLAE allows them to define *factors*. A factor is a function of a set of rules and represents the student's

[2] When OLAE makes the posterior rule probabilities from this problem into the prior probabilities for subsequent problems, it loses important information about conditional dependencies among the rules introduced by the data from previous problems. We are exploring methods for preserving these dependencies.

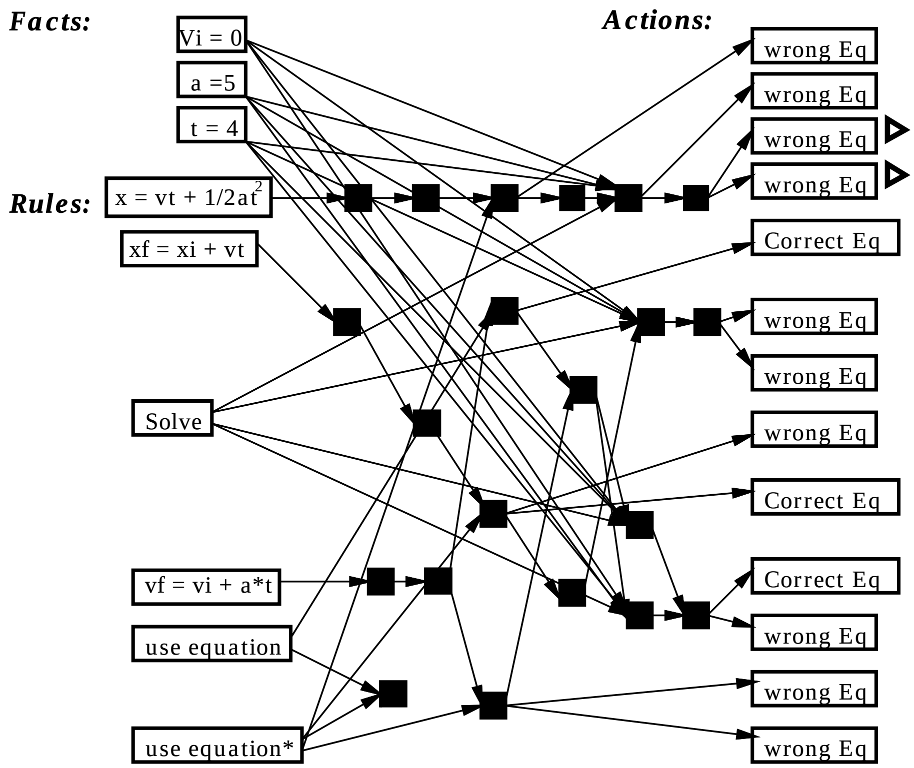

Figure 5: A simplified Bayesian net generated for the truck problem.

mastery of those rules.

The assessor's interface presents another Bayesian net. It has two types of nodes: (a) the rule nodes from the Bayesian net that analyzes data and (b) assessor-defined factors. The edges in the net that point toward a given factor indicate which rules or other factors contribute to that factor. The assessor specifies a factor by creating a node for it and entering edges from rule nodes or other factor nodes that are necessary components of it. For example, the assessor might define a factor *Chapter 5 Mastery*, and add edges that connect it to the three rules for Newton's three laws and the node for *Kinematics mastery*.

The assessor's interface displays a network of rules and factors that looks similar to the network shown in Figure 5. The assessor can scroll through these rather large 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.

The assessor can also manipulate OLAE's assessment. If the assessor strongly believes that the student knows a physics principle, perhaps from having talked with the student, she can manually increase the probability of that principle, and OLAE will update the probabilities of other factors to reflect this new information.

2 Current State and Future Work

The OLAE system is currently implemented in a preliminary form. Data collection has been implemented for all 5 task interfaces and is working smoothly. Data from pilot subjects have been collected, with between 2 and 10 subjects per activity. These preliminary evaluations of the interfaces show that they are easy to use and that they collect a great deal of relevant data about students engaged in authentic activities.

In addition, code for creating a Bayesian network, displaying it, and updating it was written, tested, and distributed to other sites on the internet. Code for generating a network from a knowledge base has been implemented, but some interpretation of problem solving data is currently done by hand. Interpretation of data from other activities has not yet been done.

One difficulty with the current OLAE system is that it can only analyze a few problems before the network gets too big. This is due to fact that the network represents algebraic reasoning as well as physics reasoning. In a simple 24-rule kinematics library that we use to test OLAE, only 7 rules concern physics. This means that most of the nodes in a problem's network concern algebraic actions (e.g., isolating a variable) instead of physics actions (e.g., invoking a kinematic principle). For instance, in one problem, the network has 43 nodes of which 21 correspond to algebraic actions.

Our proposed solution is to assume that the algebraic rules are certainly known by the student so they do not have to be represented in the network. However, the leaves in this reduced network will generally not directly correspond to the actions a student makes, because the student might type an algebraic variant of an equation represented in a leaf node. To solve this, a network's leaves will be connected to observable actions only after seeing the student's solution lines. The basic idea is to work backwards from the observed lines collecting all algebraic derivations that connect the solution line to leaves in the problem's network. Once these derivations have been found, they are compacted and converted to a shallow Bayesian network that connects the leaves to the observed actions.

3 Conclusions

The introduction listed three goals for the OLAE system. Here, we consider the extent to which the existing system satisfies those goals.

The first goal was to demonstrate that student modeling could be conducted with defensible data analyses. So far, we have demonstrated this only for single problems in simple domains. We expect some difficulty in scaling up to multiple problems and larger domains, but expect that our basic Bayesian framework will carry through unscathed.

The second goal was to demonstrate that aspects of expertise that are not easily inferred from process data can be uncovered by instrumenting tasks from the expert-novice literature and integrating those data with process data from problem-solving. At this point, this goal has been only partly achieved. We have selected tasks, instrumented them, collected pilot data, and examined those data by hand. They look useful, but we will not know if this goal can be fully achieved until data analysis and integration algorithms are devised and tested.

The third goal was to provide a flexible, graphical interface for assessors that will allow them to make well-informed decisions. We have a prototype of such an interface. Our next step is to see if real assessors can use it. This requires getting more of the rest of OLAE working so that the assessors will have student data to work with.

Acknowledgments

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 to K. VanLehn, M. T. H. Chi, and R. Glaser.

Bibliography

Burton, R.R. (1982). Diagnosing bugs in a simple procedural skill. In D. Sleeman and J.S. Brown (Eds.) *Intelligent Tutoring Systems*, Academic Press, New York.

Chi, M.T.H., Feltovitch, P., & Glaser, R. (1981). Categorization and representation of physics problems in novices and experts. *Cognitive Science, 5*, 121-152.

Chi, M.T.H., Glaser, R., & Rees, E. (1982). Expertise in problem solving. In R.J. Sternberg (Ed.) *Advances in the Psychology of Human Intelligence, Vol. 1.* Hillsdale, NJ: Erlbaum.

Collins, A. (1990). Reformulating testing to measure learning and thinking. In N. Frederiksen, R. Glaser, A. Lesgold, & M.G. Shafto (Eds.), *Diagnostic monitoring of skill and knowledge acquisition.* Hillsdale, NJ: Erlbaum.

Hestenes, D., Wells, M., & Swackhamer, G. (1992). Force concept inventory. *The Physics Teacher, 30,* 141-158.

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

VanLehn, K., & Jones, R.M. (in prep.). Self-explanation and analogy: A fine-grained analysis of learning during physics studying.

VanLehn, K., & Jones, R.M. (in press). 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., & Chi, M.T.H. (1992). A model of the self-explanation effect. *Journal of the Learning Sciences, 2*, 1-60.