Research Report

Evaluation of Automated Diagnosis and Instruction for Mathematics Problem-Solving

Jody S. Underwood

Research & Development

July 2007
RR-07-28

Evaluation of Automated Diagnosis and Instruction for Mathematics Problem-Solving

Jody S. Underwood
ETS, Princeton, NJ

July 2007

As part of its educational and social mission and in fulfilling the organization's nonprofit charter and bylaws, ETS has and continues to learn from and also to lead research that furthers educational and measurement research to advance quality and equity in education and assessment for all users of the organization's products and services.

ETS Research Reports provide preliminary and limited dissemination of ETS research prior to publication. To obtain a PDF or a print copy of a report, please visit:

http://www.ets.org/research/contact.html

Abstract

The past several decades have seen numerous approaches toward automated diagnosis and instructional support of students engaged in mathematics problem-solving. These approaches typically involve detailed analysis of potential solution paths for problems, formal representations of correct and incorrect answers, and support in the form of feedback or explanations to students during the process of solving a problem. The approaches of each of a number of representative systems (ACED, ALEKS, Cognitive Tutors, Andes, and Assistments) will be described through a critical evaluation of how they represent content knowledge, their approaches to diagnosis, and their approaches to instructional support. Finally, recommendations are made for new approaches to automated diagnosis and instructional support of mathematics problem-solving.

Key words: Mathematics education, intelligent tutoring systems, automated diagnosis, feedback, student modeling, proficiency representation

Acknowledgments

The author wishes to thank the researchers who reviewed the papers and systems that contributed to this report: Yigal Attali, Malcolm Bauer, James Fife, Edith Aurora Graf, Irvin Katz, Peggy Redman, Valerie J. Shute, Jody S. Underwood, and Juan-Diego Zapata-Rivera. Additional thanks go to Val Shute and Aurora Graf for their helpful remarks on earlier drafts.

Table of Contents

The purpose of this review is to identify existing approaches to the automated diagnosis of mathematics problem solving. In the 2005 fiscal year, a group of reviewers (listed in the Acknowledgments section) reviewed 14 papers and systems that focus on the automated diagnosis and instruction of problem solving in mathematics. A subset of these systems was chosen for this review, including ACED, ALEKS, Andes,[1] Assistments, and Cognitive Tutors (see Table 1), because they are working systems and have been used by the target audience as well as evaluated. The review ends with recommendations for future development of a diagnostic capability in support of learning.

Table 1

Representative Automated Diagnosis and Instructional Support Systems

System	Reference	Content area
Adaptive Content for Evidence-based Diagnosis (ACED)	Shute, Graf, & Hansen, 2005	Algebra (sequences as patterns)
Assessment and Learning in Knowledge Spaces (ALEKS)	Falmagne, Cosyn, Doignon, & Thiery, 2004	Algebra, arithmetic
Andes1 & Andes2	VanLehn, Lynch, Schulze, Shapiro, Shelby, Taylor, et al., 2005	Physics (equations, derivations, & solutions)
Assistments	Razzaq, Feng, Nuzzo-Jones, Heffernan, Koedinger, Junker, et al., 2005	Algebra
Cognitive tutors	Anderson, Corbett, Koedinger, & Pelletier, 1995	Algebra, geometry, LISP programming, other areas

It is difficult to separate the diagnosis from the instructional support that these systems perform; they work hand-in-hand. Thus, it is useful to describe both diagnosis as well as instructional support in this review.

The past several decades have seen numerous approaches toward automated diagnosis and instructional support of students engaged in problem solving. These approaches, which are represented by the reviewed systems, typically involve detailed analysis of potential solution paths for problems, formal representations of correct and incorrect answers, and support in the form of feedback or explanations to students during the process of solving a problem. The approaches of each of the representative systems will be described by looking at how they represent content knowledge, their approaches to diagnosis, and their approaches to instructional support, using examples from the reviewed systems for clarity (see Table 2). These categories are more fully described and explored in the sections that follow.

To talk about approaches to diagnosis and instructional support, it is necessary to first describe how the systems represent the knowledge, skills, and abilities (KSA) that the systems leverage to carry out diagnosis and instructional support.

1. Representing KSAs

ACED has a proficiency model that structures KSAs in terms of general-to-specific relations. For example, *understands sequences as pattern*s is the most general KSA that ACED has in its proficiency model and *can extend a geometric sequence* is at the most specific level. The general-to-specific relation is defined by *part-of links*—for example, *can extend a geometric sequence* is part of its parent KSA, *solves problems using geometric sequences*. This structure allows the system to make inferences about student knowledge of more general KSAs based on performance on items on more specific KSAs. That is, if a student can extend a number of geometric sequences, the system can say with some confidence that this student knows something about sequences as patterns in general.

ALEKS uses a precedence network to structure its KSAs. ALEKS characterizes item types with short labels (e.g., word problems on proportions). To create the precedence model, experts were asked to determine the answers to the following two questions. A precedence structure is created based on the answers.

Q1. Suppose that a student is not capable of solving problem p. Could this student nevertheless solve problem p'?

Q2. Suppose that a student has not mastered problems p1, p2, . . ., pn. Could this student nevertheless solve problem p'?

Table 2

Approaches to KSA Representation, Diagnosis, and Instructional Support for All Reviewed Systems

System	KSA representation	Diagnosis			Instructional support		
		KSA proficiency determination	Common errors	Item selection	Explanations	Persistent student model	Provides answer
ACED	General to specific part-of links between KSAs	Bayesian inference network, probabilistic determination	yes	Weight of evidence algorithm to adaptively select an item	Common error feedback or hint	yes	yes
ALEKS	Precedence network of KSAs	Initial adaptive test, knowledge state, probabilistic determination	yes	Set selected after initial assessment, easier problems selected during learning mode	Solution steps, common error feedback	yes	yes
Andes1	Links between items and KSAs only	Bayesian procedures, probabilistic determination	yes	Student selects new items. Andes1 uses Bayesian analysis to determine which step within an item to help a student	Right/wrong feedback after each step, hints, common error feedback	yes	yes
Andes2	Links between items and KSAs only	None	yes	Student selects new items	Right/wrong feedback after each step, hints, common error feedback	no	yes

(Table continues)

Table 2 (continued)

System	KSA representation	Diagnosis			Instructional support		
		KSA proficiency determination	Common errors	Item selection	Explanations	Persistent student model	Provides answer
Assistments	Links between items and KSAs only	Uses amount & nature of assistance to judge limitations	yes	Predetermined order	Scaffolded questions about solution steps, on-demand hints, common error feedback	yes	yes
Cognitive tutors	Links between items and KSAs only	Bayesian procedures, probabilistic determination	yes	Tasks are selected to target deficient skills	Common error feedback, next step with explanation, context-specific help	yes	yes

Falmagne et al. (2004) showed an example from the resulting precedence network where "word problems on proportions" precedes "multiplication of monomials," which in turn precedes "graphing a line through a given point with a given slope." While the two questions imply a direct prerequisite relation from one item type to another, the resulting precedence model seems to be more general than a prerequisite relation, in that, as Falmagne et al. pointed out, some concepts are taught in a particular order even though there may be no logical or pedagogical reason to do so (p. 4). That is, the experts answered the questions, and hence, ALEKS bases its KSA relations based on the order they are generally taught (chronological).

In Andes1 and Andes2, the author of the problem creates a graphical description of the steps to solve the problem, and the Andes engine automatically generates the problem solution space. These contain the concepts and equations (KSAs) that are needed to solve the problem and are defined for each problem. There are no links between the KSAs. Assistments have links from items to KSAs, but there are no connections between any of the KSAs. The links between items and KSAs are defined by experts.

The Cognitive Tutors use a production system to model knowledge. Each rule in the production system represents a skill that can be used to solve a problem or make progress in solving a problem. Each problem has links to the skills necessary to solve that problem, but there are no links between the KSAs themselves.

2. Performing Diagnosis

Diagnosis is performed differently for different purposes. These purposes include determining overall KSA proficiency, identifying errors that are commonly made, and selecting the next task or item.

2.1 KSA Proficiency and Student Modeling

ACED performs KSA proficiency diagnosis. It uses a Bayesian inference network to compute probabilistic estimates of a student's proficiency based on his performance on each task and through all sessions. At any point, the system has a snapshot of the student's estimated proficiency for all KSAs.

ALEKS starts a session with a student by administering an adaptive assessment. This initial assessment is adaptive in that the choice of each new question is based on an aggregate of responses to all previous questions. Using its knowledge structure and the initial assessment,

ALEKS probabilistically determines the KSAs the student has mastered and a set of problem types he is ready to learn. It maintains a persistent student model that keeps track of performance on KSAs through all sessions with a student.

Andes1 uses a Bayesian student model that keeps track of the proficiency of students for different tasks and concepts. While this information can be used to decide which problem a student should do next, they discontinued use of this feature of Andes1 in developing Andes2 because the developers determined that keeping track of a student model is only useful when students have a choice in the order in which they learn things, or at the pace at which they learn them.

In Assistments, information about KSA performance is collected while the student is solving a problem. The amount and nature of the assistance that students receive is used to judge student KSA strengths and weaknesses. During the weekly sessions students have with Assistments, the system maintains a persistent student model to keep track of students' abilities with the goal of providing increasingly accurate predictions of how students will perform on a particular standardized mathematics test.

The Cognitive Tutors estimate students' mastery of the mathematics skills in its rule base by using a Bayesian procedure that computes the probability the student has learned each of the KSAs.

2.2 Common Errors

Common errors are those which many students tend to make while solving a problem. All the systems (ACED, ALEKS, Andes, Assistments, Cognitive Tutors) identify common errors either during or after problem solving. In ACED and Assistments, expert opinion was used to determine commonly found errors and to create relevant feedback when they are encountered, but there is not a lot of information about how common errors are determined in the other three systems. Typically, when a common error is detected in a student's answer, the tutor provides hints or explanations with that context in mind. A recent evaluation of ACED, for instance, has demonstrated positive results in terms of student learning by taking this approach to diagnosis and instructional support (see Shute, Hansen, & Almond, in press).

2.3 Item Selection as Diagnosis

Item selection may be accomplished by predetermining a set of items to present to students or through adaptive means by an intelligent system. According to Shute and Zapata-Rivera (in press), researchers often distinguish between macroadaptation and microadaptation. For instance, Snow (1992) described two different levels of adaptive interaction. *Macroadaptation* typically occurs at the outset of instruction, based on results from some analysis of incoming skills, abilities, or disabilities, and students are assigned to different instructional paths from the start, as is done by ALEKS's initial assessment. In *microadaptation*, students' specific needs are diagnosed during the course of instruction, and the next assessment question is selected based on these diagnoses.

ACED uses a weight of evidence algorithm to adaptively select an item that will provide the most new information about a student (microadaptation).

Using its knowledge structure and an initial assessment, ALEKS probabilistically determines the KSAs the student has mastered and determines a set of problem types he is ready to learn (macroadaptation). If a student is unable to solve a problem, ALEKS can switch to a related but easier problem, where *easier* is defined in terms of precedence (microadaptation). If both problems require a KSA that the student does not have, ALEKS has no way of identifying it.

Andes1 and Andes2 do not do adaptive item selection at all. While Andes1 uses a Bayesian student model that keeps track of the proficiency of students for different tasks and concepts, and could in theory select new items, the Naval Academy students were assigned specific problems for homework, so Andes1 does not need, and in fact is not empowered, to select homework problems.

Assistments have a predetermined order in which to present items to students (non-adaptive). The Cognitive Tutors assign students new problems that target deficient skills (microadaptation).

3. Instructional Support as Diagnosis

Each of these systems provides instructional support in response to a student entering an incorrect answer or incorrect step in solving a problem. These approaches include letting the student know whether it was right or wrong, hints about how to produce better work or how to proceed, full explanations of how to solve the problem from beginning to end, descriptions of potentially useful concepts, help with problem-solving steps, and providing a solution to the

problem. These instructional approaches in response to diagnoses are also a form of microadaptation.

In ACED, instructional support is provided to the student in the form of a brief explanation when a solution to a question is incorrect. It tailors the feedback to address common errors or, if the solution is not a common error, gives a general hint about how to proceed. Consider a task in which ACED asks the student to find the common difference in the arithmetic sequence: 4, 7, 10, 13. Suppose the student types in 16 as the answer. The feedback says, "Nice try, but incorrect. You typed the next number in the sequence, but you should have typed in the common difference, which is 3."

Whenever a student attempts to solve a problem while learning in ALEKS, the system responds by saying whether or not the answer is correct and, if it is incorrect, what the student's error may be (by attempting to match it against representations of common errors). For example, consider the problem: "Express 98 as a product of prime numbers." If a student enters "98 = 2 x 14" ALEKS responds with, "Some of the factors you found are correct, but you also added some incorrect ones. Make sure your product is equal to 98." If a student enters "98 = 2 x 49" ALEKS responds with "Make sure all the factors in the expression are prime numbers." When a student asks for an explanation, ALEKS provides an explanation that lists all the solution steps to a problem, through to the solution. If a student continually provides incorrect responses, ALEKS may suggest that the student look up the definition of a certain word in the dictionary.

Andes1 tried to determine which correct equation the student intended by using a syntactic distance technique (Gertner, Conati, & VanLehn, 1998, as reported in VanLehn et al., 2005). Despite considerable tuning, they report that this technique never worked well. Even when Andes1 identified a correct equation that was what the student intended to enter, its hints only pointed out the difference between the two. For instance, if the equation the student intended to enter was $Fw = mc*g$ and the student actually entered $Fw = -mc*g$, then Andes1 could only say "Sign error: You need to change $-mc*g$ to $+mc*g$." When receiving such a hint, students rewrote the equation as directed and were mystified about what the new equation meant and how Andes1 derived it. That is, because Andes1 used a syntactic, knowledge-free technique for analyzing errors, it gave syntactic, knowledge-free hints on how to fix them. They state that such hints probably did more harm than good.

Andes2, in contrast, gives hints on errors only when the hint is likely to have pedagogical value. Toward this end, it has a set of error handlers, each of which recognizes a specific kind of error and gives a hint sequence that helps the student learn how to correct it. For errors that are likely to be careless mistakes, Andes2 gives unsolicited help, while for errors where some learning is possible, Andes2 gives help only when asked. This policy is intended to increase the chance that students will repair substantive errors without asking for help. Self-repair may produce more robust learning, according to constructivist theories of learning (e.g., Merrill, Reiser, Ranney, & Trafton, 1992, as reported in VanLehn et al., 2005). There are two different types of help available in Andes2 that generate a sequence of hints: What's Wrong Help and Next Step Help. The teaching hint, "If you are trying to calculate…" states the relevant piece of knowledge. These hints are kept as short as possible, because students tend not to read long hints (Anderson et al., 1995; Nicaud, Bouhineau, Varlet, & Nguyen-Xuan, 1999, both reported in VanLehn et al., 2005). Although teaching hints allow just-in-time learning, real-world students are sometimes more concerned about getting their homework done than with learning (Dweck, 1986, as reported in VanLehn et al., 2005).

Assistments are built around individual items. When a student answers an item incorrectly, students are not allowed to try the item again, but instead must answer a sequence of scaffolding questions presented one at a time, creating a path to a correct solution. The student works though the scaffolding questions, with the ability to ask for hints. The student can ask for a hint up to three times, where successively more detailed hints are given, and the answer is presented the third time. If a student provides an incorrect answer, a buggy message is displayed in response to that error, similar to feedback from ACED. It also summarizes overall student performance and reports it to the teacher.[2]

In the Cognitive Tutors, no comment is given if the student is correct. If the student makes a recognizable error, a buggy production rule fires and a help message is presented to guide the student down a path to the correct solution. If the student appears to be floundering (e.g., repeats the same type of error three times or makes two mistakes that the tutor does not recognize), the tutor will provide a correct next step in a solution, along with an explanation. As with Assistments, a student can ask for help up to three times. The help is specific to the context that the student is in, and the answer is presented the third time.

3.1 Assistance With Solving Multistep Problems

How can a tutor help students with multistep problems? This section is adapted from an interesting review by VanLehn et al. (2005) about tutoring individual steps in multistep problem solving. They illustrated three possible (of potentially many) user interfaces that take different approaches to entering intermediate steps when solving an equation.

User Interface A:

3x + 7 = 25

3x = _____

x = _____

User Interface B:

3x+7 = 25

_____ = _____

_____ = _____

_____ = _____

x = _____

User Interface C:

3x+7 = 25

3x = _____ Justification: _____

x = _____ Justification: _____

User Interface A encourages a specific strategy, namely subtracting the 7 then dividing by 3. However, it also blocks another strategy, namely dividing by 3 then subtracting 7/3. User Interface B allows students to enter any expressions they want in the blanks, as long as the resulting equations are mathematically equivalent to the first equation. This user interface allows students to use many kinds of problem-solving strategies, including inefficient ones. User Interface C enforces the same strategy as User Interface A, but in addition requires the student to justify each step mathematically, requiring "subtract 7 from both sides" in the first justification blank. In the final row, the student must enter "18/3" or "6" in the blank and "divide by 3 on both sides" in the justification blank.

Many tutoring systems have user interfaces that constrain student reasoning, but their impacts on learning are mixed. For instance, Singley (1990, as reported in VanLehn et al., 2005) found that students' learning was accelerated when they pursued a goal-directed problem-solving

strategy by explicitly entering their intermediate goals. However, a similar manipulation did not have the same effect in a geometry tutoring system (Koedinger & Anderson, 1993, as reported in VanLehn et al., 2005) and yielded only a mild positive effect with a propositional logic tutoring system (Scheines & Sieg, 1994, as reported in VanLehn et al., 2005). Constraining student reasoning is probably just as complex a design issue as feedback or hints. There is not yet an empirically grounded theory of which constraints are beneficial to whom at what times in their learning.

Andes2 puts little constraint on students' reasoning and problem solving. Its user interface is like User Interface B of Figure 1—students can fill in the blanks any way they want as long as their equations and other entries are true statements. The user interface is left unconstrained so that it would be similar to pencil and paper because, in general, one gets higher transfer from training to testing when the user interfaces are similar (Singley & Anderson, 1989, as reported in VanLehn et al., 2005). Although the unconstrained Andes2 user interface might reduce the *rate* of student learning compared to a constrained user interface, transfer literature suggests that mastery of physics on the Andes2 user interface should almost guarantee mastery on pencil and paper. Moreover, keeping the user interface unconstrained makes Andes2 less invasive.

3.2 Other Support for Learning

Most homework help systems present a score that is a function of the correctness of the student's answer and the number of hints received (VanLehn et al., 2005). Andes2 similarly computes and continually displays an overall score while students use the system. This score is not used in any summative way; it is only for the use of the student while solving the problems. To compute the score, Andes2 puts little weight on answers. Instead, it measures the proportion of entries that were made correctly. VanLehn et al. (2005) found that counting hints negatively in the score tends to discourage students, so Andes2 only subtracts points when students ask for bottom-out hints. In addition, Andes2 tries to encourage good problem-solving habits by awarding points for entering certain information explicitly. Students can also see the subscores from which their score was computed. These scores are per problem and are not cumulative. Some students seem to be highly motivated by their Andes2 score, even though they understand that nothing depends on it.

Andes2 includes help that is not sensitive to the current problem-solving state. Such unintelligent help consists of text and other passive material where students search for useful information. An example is the Andes2 Cheat Sheet, which is a hierarchical menu of all the equation-generating concepts known to Andes2. Selecting an equation brings up a short explanation or example.

4. System Evaluations

Each of the systems has been evaluated in terms of student learning, predictive validity of the KSAs, or both. The dependent measures include simply using the system, feedback type, task selection, instructional method, and proficiency estimation. All the systems that evaluated student learning show various degrees of improvement.

4.1 ACED

Shute, Hansen, and Almond (in press) presented the results of an evaluation study done with ACED. Three main features of ACED were tested (feedback type, task sequencing, and proficiency estimation) with the goals of determining whether such a system works both as an assessment and to support learning. The learning questions were (a) For students using the ACED assessment, does *any learning* occur?, (b) What is the contribution of *explanatory feedback* to learning?, and (c) What is the contribution of *task sequencing* to learning? The main assessment question was: How well do the estimated proficiencies, derived from an underlying Bayes net, match outcome performance? Two main variables manipulated in the study were feedback type (accuracy-only vs. explanatory) and task sequencing (adaptive vs. linear). The accuracy-only feedback provided information to the learner about the correctness of her solution immediately after a response was entered, while the explanatory feedback described the rationale and procedure for the correct solution. There were four conditions in the experiment—three representing ACED variants, and one no-treatment control group. The ACED conditions were (a) adaptive task sequence and explanatory feedback, (b) adaptive task sequence and accuracy-only feedback, and (c) linear sequence and explanatory feedback. The fourth condition was a control group (no intervention). The three combined ACED conditions (i.e., not individual comparisons) showed significantly greater learning gains compared to the control group. There was also a significant difference between the adaptive-explanatory vs. adaptive-accuracy-only conditions, where the former group performed better on the posttest than the latter, leading to the conclusion

that context-sensitive help (often based on common errors) is effective. There was no effect of task sequencing (i.e., no significant difference between adaptive-explanatory and linear-explanatory conditions). The null finding with task sequencing is likely because the algorithm used in ACED focused on maximizing the *precision* of proficiency estimates and not the educational value of selecting the "next task."

4.2 ALEKS

ALEKS's predictive validity is respectable: "In the knowledge structure for Beginning Algebra, for example, as it is used by students today, the correlation between predicated and observed answers hovers between .7 and .8, depending on the sample of students" (Falmagne et al., 2004, p.12). They are referring here to the correlation between predicted and observed responses to the last question, based on the final knowledge state. This last question of the initial assessment is not included in the estimation of the final knowledge state, so it acts like a test.

The authors report that the precedence graphs have been further refined via data from thousands of students (Falmagne et al., 2004). However, no information was found about controlled evaluations using ALEKS as a teaching instrument—from (a) the available/posted research papers on the ALEKS web site, (b) several Google and Google Scholar searches on the topic of evaluations of the system, (c) ERIC and PsychLit searches, or (d) from corresponding by e-mail with the ALEKS information site and the first author directly.

4.3 Andes1

Several studies were conducted to analyze the effectiveness of Andes1 (VanLehn et al., 2002; VanLehn & Niu, 2001, as reported in VanLehn, 2005), and it was found that Andes1 significantly improves student learning. On one evaluation of Andes1, the mean posttest exam score of the students who did their homework on Andes1 was approximately one standard deviation higher than the mean exam score of students who did the same homework with pencil and paper (Schulze, Shelby, Treacy, Wintersgill, VanLehn, & Gertner, 2000, as reported in VanLehn et al., 2005). In addition, the Bayesian student modeling technique was found to be a highly accurate assessment of student mastery (VanLehn & Niu, 2001).

4.4 Andes2

Andes2 was evaluated in an introductory physics class every fall semester from 1999 to 2003. In general, Andes2 students learned significantly more than non-Andes2 students (VanLehn et al., 2005). The overall effect size was somewhat smaller for the final exam (0.25) than the interim exams (0.61). This may be partially due to the fact that roughly 30% of the final exam addressed material not covered by Andes2.

4.5 Assistments

Razzaq et al. (2005) conducted a study that compared two different tutoring strategies within Assistments: one in which students were first coached to set up a proportion before guiding them through the problem, and the other in which students were just guided through the problem. The authors reported a main effect for learning overall, but no difference between the two tutoring strategies.

4.6 Cognitive Tutors

Early evaluations of the Cognitive Tutors usually, but not always, showed significant achievement gains. Best-case evaluations showed that students could achieve at least the same level of proficiency as conventional instruction in one third of the time. Empirical studies showed that students were learning skills in production-rule units and that the best tutorial interaction style was one in which the tutor provides immediate feedback, consisting of short and directed error messages (Anderson et al., 1995). The tutors appear to work better if they present themselves to students as nonhuman tools to assist learning rather than as emulations of human tutors. Students working with these tutors display transfer to other environments to the degree that they can map the tutor environment into the test environment. In addition, Anderson et al. found that:

- It’s possible to take a complex competence, break it down into its components, and understand the learning and performance of that competence in terms of the components.
- In successive hinting, students are often annoyed with the vague initial messages and decide there is no point in using the help facility at all.
- Students who overuse hints learn little.

- It is more meaningful to hold constant the level of mastery required and look at differences in time to achieve that level.
- As students have more opportunities to use a production rule across exercises, their performance on the rule improves.
- Knowledge tracing has substantial impact on student achievement level.
- It is essential to tell students exactly what to do if necessary to allow them to proceed.
- Achievement gains are higher the second year teachers work with the tutor.

5. Considering Potential Applications

In order to consider the different types of diagnosis and instructional support presented in the previous sections, it is useful to list some potential uses to be considered (see Table 3). These are just preliminary ideas; researchers should determine what their goals are in developing a diagnostic system. Is it to provide homework help, thereby creating a completely different product than it has done before? To create a diagnostic tool that teachers can use to place and group students, to provide remediation for students who need help and to provide challenging material for those who are ready? Or is it software or a web-based subscription that parents can buy for their children to help them with math, reading, or writing? Table 3 lists some of the features to be considered for each of these types of systems, including whether it needs to do diagnosis, provide instructional support, be connected to a curriculum, or measure achievement of KSAs and if so, to what extent. These features may be able to help determine which approaches to take. For example, if no instructional support is needed, should items and KSAs be linked to curriculum and standards?

5.1 Homework Help

A homework help system would require diagnosis and instructional support. It could be linked to a curriculum, since the students will be doing homework for a class, or it could be linked to general topics that align with any curriculum. Achievement needs to be gauged for the purpose of the student completing the homework problems.

5.2 Teacher Use

In this case, diagnosis needs to be done with an eye toward placement or suggesting groups of students that need help in the same areas. No automated instructional support needs to be included since the teacher will be providing that. It should be linked to the curriculum and state standards so that the teacher can relate the content to what is being taught in class. Again, achievement needs to be assessed in terms of levels for the purpose of suggesting groups of students for teachers to work with.

Table 3

Potential Purposes for the Development of Diagnostic Assessments and Instructional Support, With Associated Features to Help Make Decisions for What to Include in the Respective Systems

Purpose	Diagnostic	Instructional support	Linked to curriculum	Achievement measured
Homework help	Yes	Yes	Not necessary, but might add value	Sufficient to complete homework
Teacher use (e.g., placement, grouping)	Yes	No	Yes	Levels of performance
Remediation/ challenge for individuals	Yes	Yes	Not necessary, but might add value	Mastery
Interim assessments	Yes	No	More important to be linked to standards and final test	Progress in competencies

5.3 Remediation/Challenge

This purpose is very close to the homework help idea, but targeted achievement should be mastery, since the system has to make decisions on what to teach next.

5.4 Interim Assessments

Many school districts are beginning to use interim assessments for the purpose of tracking student progress on the standards that are tested at the end of the year. Teachers want more diagnostic information about what students currently know so they can determine what they need to focus on.

These purposes will be referred to in the Discussion section.

6. Discussion

While many intelligent tutoring systems have displayed impressive results in laboratory studies (Shute & Psotka, 1996), the analyses necessary for these approaches are expensive to produce (e.g., Anderson et al., 1995) and therefore not easily scaleable. Most research on automated diagnosis and instruction tends to focus on interesting student modeling and pedagogical issues, such as how to reliably and accurately record student performance, how to identify and provide feedback for common errors, and how and when to deliver hints, feedback, and instruction. Related issues are also addressed, such as how to structure a student model, how to represent all solution paths, and how to select items. Some are issues of computation and scalability, and some are issues of pedagogy and learning. In much of the research presented in this review, these questions were not addressed in terms of *whether* they should be implemented, but only *how* they should be implemented; the researchers assumed that they are good things to do. In this section, these topics will be discussed and recommendations will be provided for directions to take.

6.1 Student Models

Student models have been used to (a) decide which problem a student should do next, (b) provide information to a teacher about a student for the purpose of grouping, lesson planning, and placement, (c) give the student feedback on problem-solving actions, and (d) predict performance on exams. One question that has not been asked is whether the information needed for each of these uses can be discerned at a single session, or whether student models need to be maintained across sessions.

While many of the systems in this review maintain a persistent student model across problem-solving sessions, the accuracy of student models can degrade as students forget as well as learn new things outside the context of the system. Sometimes, a student simply has not

learned (in a long-term sense) the KSAs they have worked on during a session. In artificial intelligence, this type of situation has been called *the frame problem.* Charniak and McDermott (1987) described it as the problem of inferring whether something that was true before is still true now. Brooks (1999) and others have circumvented the frame problem by using the world as its own model. If they want to know if a door is still locked, they will check the actual door rather than consult an internal model. There are situations in which maintaining a student model makes sense, and cases where it would be better to use the student as his own model by simply asking the student again. One response to this problem is to keep track of a student model only for self-paced courses because students do not have a choice in the order in which they learn things, or in the pace at which they learn them (VanLehn et al., 2005). None of the other systems reviewed has addressed the knowledge degradation issue.

Thus, research on student model degradation is an open area of research and could make a broad contribution to the field. One approach that ignores degradation, while still being able to diagnose difficulties, is to use a student model that keeps track of student performance just for the current session to support instruction or item selection, and to not make any assumptions about what was previously learned.

6.2 KSA Representation

How should the structure of a set of KSAs be defined? The systems reviewed link KSAs (a) to individual problems, (b) to each other via precedence relations, and (c) to each other via general-to-specific relations.

As described earlier, ACED uses general-to-specific part-of links to define its KSA structure. This allows the system to infer a student's level of knowledge of one KSA based on student performance on KSAs lower in the tree structure. However, this structure does not facilitate the diagnosis of more fundamental difficulties that students have. It can say which KSAs the student is not performing well on with some level of confidence, and it could present the student with tasks that are lower in the part-of hierarchy. As it is currently designed, however, only the leaf nodes contain tasks; in theory this can be easily remedied. However, even if there were tasks associated with each KSA, the part-of relation does not represent prerequisites. To figure out the cause of a student's difficulty in terms of which KSAs the student knows and is ready to learn, prerequisite KSAs would need to be added to the structure. For example, the prerequisites needed to extend a geometric sequence are (a) know how to construct

a geometric sequence given an initial term and a common ratio and (b) find a common ratio. This information is not represented in the ACED KSA structure; in fact, these KSAs are all at the same level in the current structure. This means that it cannot reason about where a student's difficulties lie.

ALEKS bases its conclusion that a student is ready to learn something on a *curricular chronology* of what is taught in class. That is, if the student seems to have mastered something typically taught earlier in the curriculum, then he is presumably ready to learn something typically taught later in the curriculum. What seems to be missing is that the two pieces may have nothing to do with each other; they may be in different strands of the mathematics curriculum. While ALEKS has gained statistical strength in its KSA structure through having many students use the system, it would be hard to use such a KSA structure to diagnose particular problems that a student is having.

It would be beneficial to consider taking a completely different approach to representing KSAs, so that they are scalable and modular (i.e., can be used in any curriculum). Underwood and Underwood (2007) proposed a system design that uses a prerequisite representation of mathematics concepts and that does not maintain a persistent student model. Where ALEKS bases its KSA relations on when they are generally taught in the curriculum (chronological), it would be worthwhile to have the structure reflect whether the knowledge of one KSA is required in order to learn another (causal). This approach would accurately determine what a student knows as well as identify the KSAs that the student is ready to learn, thereby informing both diagnosis and instruction.

6.3 Initial Assessments

If the purpose of the assessment is solely for diagnostic purposes (e.g., for placement without explicit instruction or feedback), then an initial adaptive assessment would be an efficient way of getting the appropriate information about KSA performance. If the purpose is to provide some instructional support to a broad area (e.g., algebra), then an initial assessment can give an indication of where a student is having problems. If a student is using a system for problem solving with supporting help (e.g., in a homework help environment), then an initial assessment might be less useful and perhaps frustrating for a student who just wants to get the homework done. It would be interesting to see if these hypotheses are correct by providing initial adaptive assessments for these different purposes.

6.4 Instructional Support

This report has outlined many issues surrounding the design of instructional support, including feedback about the veracity of a solution, explanations listing all the solution steps to a problem, scaffolding a student through the steps of a problem by asking questions, listing the solution, hints related to common errors, and hints about how to proceed. Each has its place in an instructional system, and more research needs to be done to determine when and where each is most effective. We discuss two of these below.

Common errors. The systems reviewed have tried to simulate what human tutors and teachers do to help students learn. One thing they all have in common is that they try to identify the common errors that students make. While this seems like a reasonable approach, there are a few problems inherent: (a) If the system is wrong in its identification of an error, it can be confusing to the student; (b) If the system is right, it still does not know why the student did the wrong thing (i.e., it may give an incorrect explanation of why the student made that error); (c It is difficult, if not impossible, to completely specify all the errors students can make when trying to solve a problem, let alone all the possible interactions among multiple errors; and (d) The cost associated with doing error analyses is very high. Human tutors have the added luxury of being able to see students' reactions to a response. The computer cannot rely on that. However, the computer might lose the respect of the student if a diagnosis is incorrect. It would be better to use computers to do what they do well—take the time to do a complete diagnosis. The systems described in this review all provide "low-stakes" tutoring; that is, the system should be able to quickly assess what a student knows without asking too many questions that take a lot of time, similar to how teachers assess students' understanding through homework assignments.

Additional research is needed about whether and how to provide instructional support using an alternate method without identifying common errors, which is expensive, not exhaustive, and not always correct. See Underwood & Underwood (2007) for one such approach to this type of support.

Giving the answer. All the systems reviewed eventually give the answer to the student during a help sequence. Pedagogically, it can be sound to present the solution to a problem so the student can work out how the solution was attained, or when the focus is on the conceptual level of solving a problem (e.g., setting up the right equations). However, it can also be abused by the less motivated student (Aleven & Koedinger, 2000, as reported in VanLehn et al., 2005). For

example, students sometimes click through the other hints quickly in order to reach the bottom-out hint. More research is needed to determine when it is appropriate to give an answer.

6.5 Keeping to Known Solutions Paths

VanLehn et al. (2005) made an interesting distinction between two types of correct responses:

- Correct means that the entry is valid. A valid definition is one that defines a quantity that actually exists. A valid statement is one that follows logically from the givens of the problem and the principles of the task domain.
- Correct means that the entry is valid and it appears in a known solution path to the problem.

The advantage of the second definition is that it prevents students from wandering down paths that do not lead to a solution and can make their problem solving more efficient. However, it also implies that the students will not have the experience of recovering from an error. If all their homework is done with immediate feedback that keeps them on known solution paths, students might have their first experience coping with recovery on test problems. So, it seems that the first definition of correctness might give student more experience with recovery and raise their scores on test problems that invite traveling down false paths. However, there appears to be no empirical work comparing the two types of feedback (VanLehn et al., 2005).

It would also be beneficial to research whether predetermining all possible solutions paths is tractable; it could be that for a certain set of problems, solution paths can be computed on the fly. For example, evaluate whether the next step a student takes can lead to a solution. Or, simply evaluate whether the student takes a correct step even if it is not efficient. There are a variety of tools—for example, that evaluate the equivalence of mathematics expressions, equations, and inequalities—that can be used to evaluate whether a student has entered a correct solution step.

7. Conclusions

This report has provided an analysis of automated approaches to diagnosis and instructional support for mathematics problem solving. At this point, it is worthwhile to ask if there are more scaleable approaches to automating diagnosis and instruction. There is a large literature that has examined what human tutors do—such as analyzing what they say, how they said it, and when it is

best said. For example, Merrill et al. (1992) reviewed research on human tutoring strategies in various domains and compared them to an intelligent tutoring approach called “model tracing.” Other research on human tutoring strategies tends to focus on such things as grouping students and peer tutoring (e.g., Cardona, 2002) and are not explicit about the specific strategies used in those settings. In a recent literature review that had a focus on human tutoring strategies, Naidu (2006) found that many human tutors have students solve problems while addressing errors as they arise and confirming steps when correct. She found that human tutors do not often assess a student’s general understanding of the problem solving. This approach is what many researchers are trying to automate in creating diagnosis and instructional systems.

In one case of an intelligent tutoring system design (Sleeman, Ward, Kelly, Martinak, & Moore, 1991), the researchers decided to take a procedural approach because that was the approach of all the teachers they interviewed—except one, who used a probing approach to figure out the sources of her students’ problems. They found this point worth mentioning, but did not investigate which approach would be more effective in an automated tutoring environment. It could be that probing is more effectively done by a computer even when not commonly done by people.

One approach that has not been taken by any automated tutor is to find out what students know, as opposed to how much they know. The latter approach reflects a more traditional, summative approach to assessment; or, what they “know” that happens to be incorrect, which reflects formative assessments that focus on identifying common errors and providing feedback based on those assessments. Underwood and Underwood (2007) described an approach with the idea that to build on the KSAs a student has, it is essential to first find out what they are (Carpenter, Fennema, & Franke, 1996). Trying to build on KSAs that a student does not have is misguided. This approach may prove to be a tractable solution to automating diagnosis and instructional support.

A final recommendation is to do a deeper literature review of human tutoring strategies to determine if there are any approaches not previously investigated for automated diagnosis and instructional support to mathematics problem solving that would lend itself to automation. Given the demand by parents and teachers for more detailed and diagnostic information from assessments, finding a tractable automated approach to providing this information is a worthwhile endeavor.

References

Aleven, V., & Koedinger, K. R. (2000). Limitations of student control: Do students know when they need help? In G. Gauthier, C. Frasson, & K. VanLehn (Eds.), *Lecture notes in computer science: Vol. 1839. Intelligent tutoring systems: 5th international conference, ITS 2000* (pp. 292–303). Berlin: Springer.

Anderson, J. R., Corbett, A. T., Koedinger, K., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of Learning Sciences*, *4*(2), 167–207. Retrieved March 2, 2006, from http://act-r.psy.cmu.edu/papers/129/CogTut_Lessons.pdf

Brooks, R. A. (1999). Cambrian intelligence: The early history of the new AI. Cambridge, MA: MIT Press.

Cardona, C. (2002). Adapting instruction to address individual and group educational needs in maths. *Journal of Research in Special Educational Needs*, *2*(1), 1–17.

Carpenter, T. P., Fennema, E., & Franke, M. L. (1996). Cognitively guided instruction: A knowledge base for reform in primary mathematics instruction. *The Elementary School Journal*, *97*(1), 3–20.

Charniak, E., & McDermott, D. (1987). *Introduction to artificial intelligence.* Reading, MA: Addison-Wesley.

Dweck, C. S. (1986). Motivational processes affecting learning. *American Psychologist, 41*, 1040–1048.

Falmagne, J.-C, Cosyn, E., Doignon, J.-P, & Thiery, N. (2004). *The assessment of knowledge, in theory and in practice.* Retrieved February 20, 2006, from http://www.business.aleks.com/about/Science_Behind_ALEKS.pdf.

Gertner, A., Conati, C., & VanLehn, K. (1998). Procedural help in Andes: Generating hints using a Bayesian network student model. In *Proceedings of the 15th national conference on artificial intelligence, Madison, Wisconsin* (pp. 106–111). Retrieved March 2, 2006, from http://www.cs.ubc.ca/~conati/my-papers/Gertner-AAAI98.pdf

Koedinger, K., & Anderson, J. R. (1993). Reifying implicit planning in geometry: Guidelines for model based intelligent tutoring system design. In S. P. Lajoie & S. J. Derry (Eds.), *Computers as cognitive tools*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Merrill, D. C., Reiser, B. J., Ranney, M., & Trafton, J. G. (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences, 2*(3), 277–305.

Naidu, P. (2006). Literature review: Tutoring. *The Journal of the Association for the Tutoring Profession, 1*(1). Retrieved May 9, 2006, from http://www.jsu.edu/depart/edprof/atp/ejournal.htm

Nicaud, J.F., Bouhineau, D., Varlet, C., & Nguyen-Xuan, A. (1999). Towards a product for teaching formal algebra. In S. P. Lajoie & M. Vivet (Eds.), *Artificial intelligence in education* (pp. 207–214). Amsterdam: IOS Press.

Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N. T., Koedinger, K. R., Junker, B., et al. (2005). The Assistment project: Blending assessment and assisting. In C.K. Looi, G. McCalla, B. Bredeweg, & J. Breuker (Eds.), *Proceedings of the 12th artificial intelligence in education conference* (pp. 555–562). Amsterdam: ISO Press. Retrieved March 7, 2006, from http://www.assistment.org/project/papers/AIED2005/mainAIED.pdf

Scheines, R., & Sieg, W. (1994). Computer environments for proof construction. *Interactive Learning Environments, 4*(2), 159–169.

Schulze, K.G., Shelby, R.N., Treacy, D.J., Wintersgill, M.C., VanLehn, K., & Gertner, A. (2000). Andes: An intelligent tutor for classical physics. *The Journal of Electronic Publishing, 6*(1). Retrieved June 7, 2007, from http://www.press.umich.edu/jep/06-01/schulze.html

Shute, V. J., Graf, E. A., & Hansen, E. (2005). Designing adaptive, diagnostic math assessments for sighted and visually-disabled students. In L. PytlikZillig, R. Bruning, & M. Bodvarsson (Eds.), *Technology-based education: Bringing researchers and practitioners together* (pp. 169–202). Greenwich, CT: Information Age Publishing.

Shute, V. J., Hansen, E., & Almond, R. (in press). An assessment for learning system called ACED: Designing for learning effectiveness and accessibility (ETS Research Rep.). Princeton, NJ: ETS.

Shute, V. J., & Psotka, J. (1996). Intelligent tutoring systems: Past, present and future. In D. Jonassen (Ed.), *Handbook of research on educational communications and technology*. New York: Scholastic Publications.

Shute, V.J. ,& Zapata-Rivera, D. (in press). Adapter chapter: Adaptive tools in educational communications and technology. In J. M. Spector, D. Merrill, J. van Merriënboer, & M. Driscoll (Eds.), *Handbook of research on educational communications and technology* (3rd ed.). Mahwah, NJ: Erlbaum Associates.

Singley, M. K. (1990). The reification of goal structures in a calculus tutor: Effects on problem solving performance. *Interactive Learning Environments, 1*, 102–123.

Singley, M.K., & Anderson, J.R. (1989). *The transfer of cognitive skill*. Cambridge, MA: Harvard University Press.

Sleeman, D., Ward, R. D., Kelly, A. E., Martinak, R., & Moore, J. L. (1991). An overview of recent studies with PIXIE. In P. Goodyear (Ed.), *Teaching knowledge and intelligent tutoring*. Norwood, NJ: Ablex Publishing.

Snow, R. E. (1992). Aptitude theory: Yesterday, today, and tomorrow. *Educational Psychologist, 27*(1), 5–32.

Underwood, J. U., & Underwood, I. M. (2007, April). *Immediation: Automatically diagnosing what students know, to teach what they are ready to learn*. Paper presented at the annual meeting of the American Educational Research Association, Chicago, IL.

VanLehn, K, Lynch, C., Schulze, K, Shapiro, J.A., Shelby, R. Taylor, L., et al. (2005). The Andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence in Education, 15*(3). Retrieved February 23, 2006, from http://www.andes.pitt.edu/Pages/AndesLessonsLearnedForWeb.pdf

VanLehn, K., Lynch, C., Taylor, L., Weinstein, A., Shelby, R. H., Schulze, K. G., et al. (2002). Minimally invasive tutoring of complex physics problem solving. In S. A. Cerri, G. Gouarderes, & F. Paraguacu (Eds.), *Lecture notes in computer science: Vol. 2363. Proceedings of the 6th international conference on intelligent tutoring systems* (pp. 367–376). Berlin: Springer.

VanLehn, K., & Niu, Z. (2001). Bayesian student modeling, user interfaces and feedback: A sensitivity analysis. *International Journal of Artificial Intelligence in Education, 12*(2), 154–184.

Notes

[1] The Andes systems are being reviewed because their focus is on algebraic-type equations to solve physics problems and therefore are relevant to a review on mathematics problem solving. Where there is reference to “Andes” it means both Andes1 and Andes2.

[2] For example, Assistments generate reports to answer the following questions about items, students, KSAs, and student actions: Which items are my students finding difficult? Which items are my students doing worse on compared to the state average? Which students are (a) doing the best, (b) spending the most time, and (c) asking for the most hints etc.? On which of the approximately 80 KSAs that we are tracking are students doing the best/worst? What are the exact actions that a given student took?