When and Why Does Mastery Learning Work: Instructional Experiments with ACT-R "SimStudents"

Benjamin MacLaren and Kenneth Koedinger

Carnegie Mellon University, Pittsburgh PA
(ben@cs.cmu.edu), (koedinger@cmu.edu)

Abstract. Research in machine learning is making it possible for instructional developers to perform formative evaluations of different curricula using simulated students (VanLehn, Ohlsson & Nason, 1993). Experiments using simulated students can help clarify issues of instructional design, such as when a complex skill can be better learned by being broken into components. This paper describes two formative evaluations using simulated students that shed light on the potential benefits and limitations of mastery learning. Using an ACT-R based cognitive model (Anderson & Lebiere, 1998) we show that while mastery learning can contribute to success in some cases (Corbett & Anderson, 1995), it may actually impede learning in others. Mastery learning was crucial to learning success in an experiment comparing a traditional early algebra curriculum to a novel one presenting verbal problems first. However, in a second experiment, an instructional manipulation that contradicts mastery learning led to greater success than one consistent with it. In that experiment learning was better when more difficult problems were inserted earlier in the instructional sequence. Such problems are more difficult not because they have more components but because they cannot be successfully solved using shallow procedures that work on easier problems.

1 Introduction

This paper describes some early experiments with a "pedagogical domain theory" we have developed for "early algebra" problem solving (Koedinger & MacLaren, 2002). This theory posits knowledge-specific generalizations about quantitative reasoning and yields explanations of and predictions about student problem-solving and learning behavior during the transition from arithmetic to algebraic competence. We describe our first efforts at using the EAPS (Early Algebra Problem Solving) theory to experiment with alternative teaching strategies.

First we will briefly review developmental data from early algebra learners, followed by an ACT-R model of this data that provides an explanation of how students acquire knowledge. Then we will describe two "SimStudent" (simulated student) experiments, teaching the model according to a standard and a novel curriculum. These experiments test some of the conditions under which Mastery learning does and does not work. And because these experiments are based on a symbolic model, we can provide a detailed explanation for when and why mastery learning works or does not. Finally, we discuss some of the limitations of the current work and how future research might address some of these concerns.

S.A. Cerri, G. Gouardères, and F. Paraguaçu (Eds.): ITS 2002, LNCS 2363, pp. 355–366, 2002.

1.1 Mastery Learning

Mastery learning theory (e.g., Bloom, 1987) presumes that complex skills can be broken down into components, and claims that "mastering" the components first will yield better performance in the long run. Mastery learning is based on the idea that the "failure to learn prerequisites skills is likely to interfere with students' learning of later skills" (Slavin, 1987).

Many researchers have supported the ideas behind mastery learning (Corbett & Anderson, 1995; Kulik, Kulik & Bangert-Drowns, 1990) but there are still some who remain unconvinced (e.g., Slavin, 1987). The main concern raised by Slavin is that remedial efforts made to benefit most of the students in a classroom will harm the learning of the better students. While this concern is well addressed by the use of individualized instruction using computer tutors, there are still unresolved issues in designing mastery learning curricula. For example, what exactly does it mean to break down a problem into "easier" problems, or prerequisites?

A more detailed explanation of when and why mastery learning works would help shed light on the debate about the value of mastery learning. We provide such an explanation based on the EAPS theory of quantitative knowledge and the utility learning mechanism of the ACT-R theory (Anderson & Lebiere, 1998).

1.2 An Integrated Model of Story and Equation Problem Solving

EAPS is designed to model student knowledge behind two kinds of quantitative reasoning tasks, story problem solving and equation solving, that have been addressed separately in prior research. The theory is informed by this prior research, by the constraints of ACT-R, and by difficulty factor assessments (DFAs) comparing student performance on story problems and matched equations (Koedinger & Nathan, 2002).

	Result-Unknown Problems	**Start-Unknown Problems**
Story Problems	After buying donuts at Wholey Donuts, Laura takes the $1.10 she paid and subtracts the 10 cents charge for the box they came in. Then she divides the resulting amount by the donut price of 25 cents to find the number of donuts bought. How many donuts did she buy?	After buying donuts at Wholey Donuts, Laura multiplies the number of donuts she bought by their price of 25 cents per donut. Then she adds the 10 cents charge for the box they came in and gets $1.10. How many donuts did she buy?
Word Equations	Starting with 1.10, if I subtract 0.10 and then divide by 0.25, I get a number. What is it?	Starting with some number, if I multiply by 0.25 and then add 0.10, I get 1.10. What number did I start with?
Equations	(1.10 – 0.10) / 0.25 = X	X * 0.25 + 0.10 = 1.10

Fig. 1. Example problems from DFA study with different combinations of difficulty factors for the problem representation and directionality

Two factors from these DFAs are illustrated in Figure 1, unknown position and presentation type. The pair of problems in each row of Figure 1 differs in where the problem unknown is positioned. The problems in column 1 are called Result-Unknowns because the unknown is the result of the process described. The problems in column 2 are called Start-Unknown because the unknown is at the start. Variations within the columns illustrate a second factor. They require the same underlying arithmetic, but differ in the representation in which they are presented. The "Story Problems" in the first row are presented verbally and include reference to a real world situation (e.g., wages). The "Word Equations" in the second row are also presented verbally but do not include a situation. Finally, "Equations" are presented symbolically and have no situational information.

In contrast with common expectations (Nathan & Koedinger, 2000), students performed better, not worse, on story problems than matched equations. The results are summarized in Figure 2. Students were 65% correct on verbal problems (like those in the top two rows of Figure 1) but only 42% correct on equations (the bottom row).

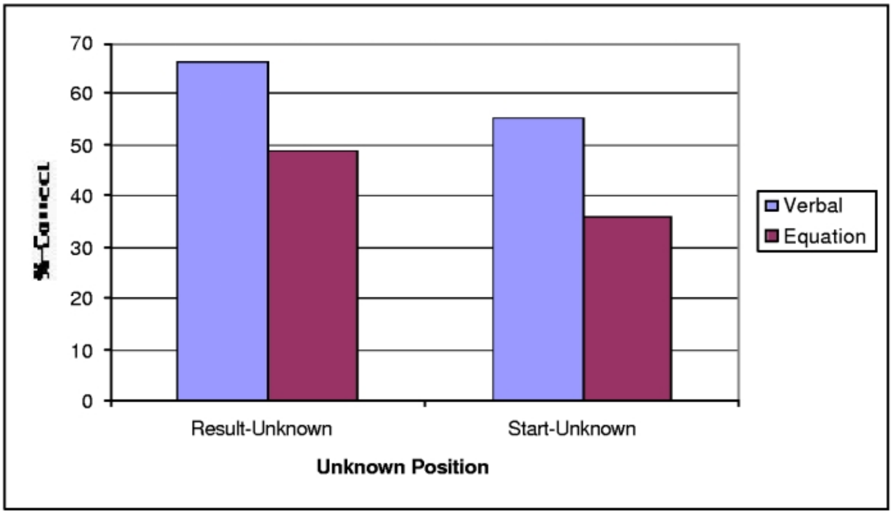

Fig. 2. Results from DFA study showing main effect of problem representation

2 A Cognitive Model of the Empirical Data

We have developed a series of cognitive models to explain these and other observations in the DFA data, called EAPS (Koedinger & MacLaren, 2002). In EAPS the general flow of control is 1) comprehend the problem presentation (whether story, word, or equation) to extract relevant arithmetic operators and their arguments, 2) manipulate the operators as necessary (e.g., invert them), and 3) solve any arithmetic subgoals that are produced. EAPS models two general classes of errors: arithmetic and conceptual. Conceptual errors include things like forgetting to change the sign when removing an operator in the verbal representation or confusing the order of operations in the symbolic representation. For arithmetic errors, we model bugs (miss-alignment of decimal places in doing arithmetic) and slips (e.g., 2*3=5).

EAPS explains the DFA results by hypothesizing differences in verbal versus equation comprehension processes. First, consistent with students' extended English language experience and limited symbolic algebra language experience, the production rules that comprehend verbal representations are more likely to fire than those that comprehend equations. Second, the equation comprehension productions include certain buggy productions that result from shallow processing of past instructional experiences (e.g., not respecting order of operations).

EAPS is implemented in the ACT-R production system (Anderson & Lebiere, 1998). Components of knowledge include declarative chunks to represent quantities and quantitative relations and if-then production rules to represent procedures for quantitative reasoning.

2.1 ACT-R's Utility Learning Mechanism

One central issue in production-based cognitive architectures like ACT-R is how to choose which production to fire when several match. If the condition side of several productions applies, there is a conflict that needs to be resolved. To model these situations, ACT-R includes a "rational" component for conflict resolution based on decision theory. To determine which production to fire if there is a conflict, the expected gain of each production is computed. The expected gain or utility of a production is defined to be equal to PG–C, where G is the estimated value of the current goal, P is the estimated likelihood that executing the production will eventually satisfy the current goal and C is the estimated cost of executing the production.[1]

The P term is determined by another equation: R_{alpha} / (R_{alpha}+R_{beta}), where R_{alpha} and R_{beta} refer to the number of eventual successes and failures that occurred when a given production fired respectively (Lovett, 1998). These successes and failures include those before a simulation (estimated from empirical data) and those during a simulation. As a model runs and gets feedback, it updates the R_{alpha} and R_{beta} values for each production that fires in a problem, thus changing their estimated R values.

In ACT-R's conflict resolution mechanism, the production with the highest utility is not always chosen. Rather, there is a stochastic process implemented as a Gaussian noise parameter that will sometimes cause a production with a lower utility to be selected. The probability of selecting a given production with expected gain g_i from n possibilities is determined by the Likelihood Ratio equation (Anderson & Lebiere 1998, p. 65):

Probability of production i firing =

$$p(firing(\mathrm{Pr}odi)) = \frac{e^{g_i/t}}{\Sigma^{eg_n/t}}$$

where g is equal to PG-C, t is:

$$\sqrt{6}\sigma/\pi$$

where g is equal to PG-C, t is sqrt(6)* σ/π, with σ being the standard deviation of the noise, and the summation is across the n productions in the conflict set. Produc-

[1] ACT-R does not propose that people explicitly analyze their problem-solving choices using decision theory, but rather that their choices conform with this analysis.

tion rule utilities, combined with the Gaussian noise parameter, provide a useful approach to modeling student choice behavior.

Using ACT-R's utility mechanism just described, we tuned our EAPS cognitive model to behave in accordance with a detailed coding of students' strategies and errors (Koedinger & MacLaren, 2002). For setting initial values for R_{alpha} and R_{beta} we used an estimate of what the value of P should be and also the number of relevant problems the student is likely to have seen in prior experience (i.e., R_{alpha} plus R_{beta}).

A key subset of the 11 parameters used to fit EAPS can be seen in Table 1. The first two (VC, SC) deal with comprehension of a verbal versus a symbolic representation. OA, Operate-on-Adjacent-Numbers, is an over-general production that ignores operator precedence in an equation and, thus, sometimes produces incorrect results. CRH is for comprehending relations with hi-priority operators (e.g., "*" and "/"). The next three (AR, SL, BG) capture behavior on arithmetic.

Table 1. Subset of Parameters for the Math Model

VC	(92%)	– Comprehend-Verbal-Relation
SC, CN, CO	(78%)	– Comprehend-Symbolic-Relation,Comp-Number,Comp-Operator
OA	(4%)	– Operate-on-Adjacent-Numbers
CRH	(69%)	– Comprehend-Relation-Hi (relation with high-precedence operator)
AP	(97%)	– Arithmetic-Procedure
SL	(3%)	– Arithmetic-Slip
BG	(0%)	– Arithmetic-Bug

The math model equation representing the probability of a correct solution on a result-unknown story problem is shown in Table 2a. EAPS needs to comprehend two verbal relations and perform two arithmetic operations. Using the likelihood that these productions will fire for the average student, shown in parentheses in Table 1, we get $VC_1(.92) * AP(.97) * VC_1(.92) * AP(.97) = 80\%$ correct story result unknowns.

The model of a correct solution to a result-unknown equation like "25 * 4 + 10 = X" is a bit more complex as we explicitly represent the comprehension of the elements in the equation.[2] As can be seen in Table 2b, EAPS must comprehend two numbers (CN) and two operators (CO) before it can comprehend the symbolic relations (SC) and perform the arithmetic. A path resulting in an Order-of-Operations error is shown in Table 2b. After comprehending "800 – 30" (CN * CO * CN), EAPS decides to operate on these adjacent numbers and subtract them (OA * AP) instead of continuing to comprehend the rest of the equation. This results in an Order-of-Operations error, 100 ((80 - 30) * 2) instead 20 (80 - (30 * 2)). For more details on how the cognitive model was used to guide the creation of the math model equations and how we tuned the resulting math model see Koedinger & MacLaren (2002).

---

[2] We chose not to model verbal comprehension in contrast to equation comprehension because beginning algebra students had little difficulty comprehending the word problems used.

Table 2. Subset of Math Model Equations for Paths through Model

a. Paths leading to a correct answer for Result-unknown problems (with integer arithmetic):
Story Problem: VC * AP * VC * AP
Equation: CN * CO * CN * CO * CN * SC * AP * SC * AP
b. A path for an Order-of-Operations error on problems like "80–30*2=X"
Equations: CN * CO * CN * OA * AP * CO * CN * SC * AP

3 Developmental Models for Early Algebra Problem Solving

EAPS did a good job of capturing the main effects of the student error and strategy selection behavior for the "average" student, but in order to begin to simulate instruction using EAPS, we needed models of students at different performance levels. We needed versions of EAPS that model weaker students, that could be "taught" to become more like the better performers in our empirical data. We separated students into six developmental "zones," shown in Table 3, and then fit a version of EAPS to model each one of these zones.

Table 3. Performance Groups of Students Based on Categories of Problems Correctly Solved

Zone 0	– None
Zone 1	– Verbal arithmetic problems (top 2 on the left in Figure 1)
Zone 2	– Verbal problems (verbal algebra as well as verbal arithmetic) (top 4 problems)
Zone 3	– Arithmetic problems (symbolic as well as verbal arithmetic) (3 problems on left)
Zone 4	– All problems but symbolic algebra (all but the problem on the bottom right)
Zone 5	– All problems.

Most traditional textbooks (Nathan, Long & Alibali, 2002) present arithmetic symbol problems first, followed by arithmetic word problems, symbolic algebra problems, and finally algebra word problems. This "Textbook curriculum" is shown in Table 4a. However, students in Koedinger and Nathan (2000) got 67% of the arithmetic word problems correct, but only 49% of symbolic arithmetic problems correct. Given this data, it was natural to ask why the more difficult problems should be given earlier in the curriculum? Students might learn better following a different curriculum, which we call Verbal Precedence, shown in Table 4b, where informal verbal reasoning skills precede the equivalent reasoning skills in the symbolic representation.

Table 4. Curricula for SimStudent Experiment 1

a. Control Condition: Textbook curriculum:	
1. Arithmetic equations	49% correct in DFA data
2. Arithmetic word problems	67% "
3. Algebra equations	36% "
4. Algebra word problems	56% "
b. Experimental Condition: Verbal Precedence curriculum:	
1. Arithmetic word problems	67% correct in DFA data
2. Arithmetic equations	49% "
3. Algebra word problems	56% "
4. Algebra equations	36% "

4 SimStudent Experiment 1: Textbook vs. Verbal Precedence

The EAPS theory and simulation provide a means to evaluate whether the Verbal Precedence curriculum can lead to more effective learning than the Textbook curriculum and to provide an explanation for why it might.

We used eight hundred SimStudents in the experiment, one hundred in each of the eight conditions resulting from crossing the three dimensions shown in Table 5.

Table 5. Variables for the Eight Instructional Conditions in SimStudent Experiment 1

1. **Ordering** – Half of each of these groups followed the Verbal Precedence Curriculum where word problems were practiced before symbolic problems. The other half followed the Textbook Curriculum where symbolic problems were given first.
2. **Mastery** – Half the SimStudents solved 1-operator problems before 2-operator problems, while SimStudents in the non-mastery control condition only solved 2-operator problems.
3. **Initial competence** – Half of each group was made up of SimStudents parameterized to be in the lowest performance group (Zone 0), while the other half started with an ability to solve Verbal-Arithmetic problems (Zone 1). We used these two performance groups simply because they represent students most in need of instruction.

Procedure. Each SimStudent was presented with ten problems at each level in the chosen curriculum. If the simulated student solved eight of the ten problems successfully, it was moved on to the next problem in the curriculum sequence. If it failed to reach this level of success, it was given the same problem type again. If it failed to get eight out of ten problems correct after four attempts at any level, the SimStudent failed. SimStudents are given feedback after each problem and are performing utility learning during the presentation of each of these problems, so its performance can improve with practice. This enables EAPS to strengthen the productions that transfer from simpler problems to the harder ones that follow.

The feedback given is only whether the problem is correct or incorrect, not what the exact error was. If EAPS makes it all the way through all the problem sets in the curriculum sequence, it is considered to have passed the curriculum. If the student gives up before actually writing anything down on a problem, the productions fired up to that point were not penalized. This is because if EAPS gives up, it does not know that the productions used to that point are wrong.

The results of this experiment can be seen in Table 6. 59% of Zone 1 SimStudents make it through all levels of the Verbal Mastery curriculum, and 44% make it through all levels of the Textbook Mastery. By breaking down the problems presented into parts, the probability of getting a problem correct increases. Initial performance on a problem like "X*2=10", for example, is much greater than performance on a problem like "X*4+2=10". This performance difference dramatically improves the learning process as positive feedback is much more likely and thus good productions do not get penalized along with bad ones.

Table 6. Results from SimStudent Experiment 1

	Zone 0	Zone 1
Verbal Mastery	28%	59%
Textbook Mastery	24%	44%
Verbal Control	14%	32%
Textbook Control	6%	26%

Two conclusions can be drawn. First, the Verbal Precedence curriculum was consistently more effective than the traditional Textbook curriculum. Second, Mastery Learning proved to be a much more effective pedagogical strategy than the control. For the two curricula compared, the Mastery condition made it more likely that a SimStudent would successfully complete the curriculum. When the SimStudent did fail on a problem, feedback was more likely to be focused on those skills responsible, making blame assignment more effective.

5 Experiment 2: Discouraging Shallow Knowledge

Another set of predictions from our model that we wanted to explore followed from EAPS' representation of bugs as overly general knowledge.[3] Like other learning mechanisms, ACT-R's utility learning mechanism yields learning outcomes that are sensitive to the order in which problems are presented. In particular, some presentation orders can lead to a greater likelihood of over-general productions than others can. In poorly designed instruction, the sequences of examples and practice problems may be such that students can initially learn and be successful with an overly general production. In this case, continued practice on such problems would yield increasing utility estimates for the overly general productions. It would be then necessary to distinguish the overly general production from the correct one. This potential confusion suggests that curricula should have problems early on that will cause overly general rules to fail. In such cases, more specific correct productions increase in utility and further practice will replace the over-general production.

An example problem that will cause the over-general Operate-on-Adjacent-Numbers production to fail is "800 – 40 x 4". This production can apply to the first part of this expression and incorrectly subtract 40 from 800. It could also apply to the second relation, "40x4", and produce a correct answer. It seems a plausible instructional hypothesis that problems like "800–40x4" should be introduced early in a curriculum so that overly general productions can receive negative feedback before they have a chance to accumulate a significant utility value.

The first step we took was to modify the most successful curriculum in SimStudent Experiment 1 (using parameter setting three and Zone 1) by replacing the fourth problem with a 2-operator symbolic problem "10 + 4 * 25 = X" to the curriculum. The first four problems of the resulting curriculum are shown in Table 7.

---

[3] In ACT-R new productions are created by analogy from example problem solving traces. ACT-R hypothesizes that bugs are introduced from incorrect generalizations during analogy.

Table 7. Control Curriculum1 for SimStudent Experiment 2

1. 1-operator Story Arithmetic problems (equivalent to "4 * 25 = X")
2. 2- operator Story Arithmetic problems (equivalent to "4 * 25 + 10 = X")
3. 1- operator Symbolic Arithmetic – indifferent to Order-of-Operations: 10 * 4 = X
4. 2- operator Symbolic Arithmetic – discouraging Order-of-Operations: 10 + 4 * 25 = X
5.

As before, we then ran one hundred SimStudents on this curriculum. The SimStudents did not succeed at all with this curriculum. This failure was due to the over-general Operate-on-Adjacent-Numbers, which succeeded on single operator problems (problem category 3 in Table 7) and had its utility dramatically increased. Then, once the model started attempting two operator problems like "10 + 4 * 25" (problem category 4 in Table 7) it would usually fail because Operate-on-Adjacent-Numbers had such high utility. Thus, not only would Operate-on-Adjacent-Numbers be penalized but many other good productions that fired with it would as well.

As a potential solution to the problem of prematurely strengthening Operate-on-Adjacent-Numbers, we decided to introduce problems earlier in the curriculum where Operate-on-Adjacent-Numbers would fail, like "10 + 4 * 25". We also *raised* the R_{alpha} and R_{beta} values of the lower level symbolic comprehension productions to make them more resistant to change, and *lowered* R_{alpha} and R_{beta} values of Operate-on-Adjacent-Numbers production, to make it more sensitive to change. Using two versions of two operator result-unknown equations as the third and fourth problems in the sequence, we created two new curricula, shown in Table 8:

Table 8. Curricula for SimStudent Experiment 2

a. Control Curriculum2:

1. 1- operator Story Arithmetic problems (equivalent to "4 * 25 = X")
2. 2- operator story arithmetic problems (equivalent to "4 * 25 + 10 = X")
3. 2- operator Symbolic Arithmetic – indifferent to Order-of-Operations: 10 * 4 * 25 = X
4. 2- operator Symbolic Arithmetic – discouraging Order-of-Operations: 10 + 4 * 25 = X

b. Experimental Curriculum to challenge shallow (overly-general) knowledge:

1. 1- operator Story Arithmetic (equivalent to "4 * 25 = X")
2. 2- operator Story Arithmetic (equivalent to "4 * 25 + 10 = X")
3. 2- operator Symbolic Arithmetic – discouraging Order-of-Operations: 10 + 4 * 25 = X
4. 2- operator Symbolic Arithmetic – indifferent to Order-of-Operations: 10 * 4 * 25 = X

On the third problem category in Control Curriculum2, Operate-on-Adjacent-Numbers produces a correct answer. Operate-on-Adjacent-Numbers is then strengthened, so that when the fourth problem category in the Control Curriculum is reached, Operate-on-Adjacent-Numbers will be more likely to operate on "10+4." However, it is incorrect in this case, and all the other correct productions that fired to reach the resulting incorrect answer will also be penalized.

On the third problem in the Experimental Curriculum the situation is different. When Operate-on-Adjacent-Numbers fires on “10+4” it results in an incorrect solution. The resulting errors make it more likely that Operate-on-Adjacent-Numbers will be penalized and weakened before it has become strengthened too much.

The results of SimStudent Experiment 2 are shown in Table 9. Thirty-six percent of students made it through the experimental curriculum, compared to six percent for the control. In the control, Operate-on-Adjacent-Numbers was strengthened as it produced correct answers on the problem “10*4*25” in problem set 3. When EAPS then attempted problem set 4, “10+4*25” it often failed (because Operate-on-Adjacent-Numbers would not yield a correct answer when it applied to “10+4”), driving not only its R value down, but more importantly weakening the non-buggy productions that fired along with it to produce an answer (as can be seen in Table 2b) and reducing the likelihood of success.

Table 9. Results from SimStudent Experiment 2

	Zone 1
Challenge Shallow Knowledge	36%
Control Condition	6%

In the experimental condition where Operate-on-Adjacent-Numbers was discouraged early, its R-value was lowered by the problems in set 3, “10+4*25”, before it had a chance to get strengthened. By the time EAPS got to problem set 4 Operate-on-Adjacent-Numbers was no longer firing – it had been successfully weeded out.

It was not obvious that our hypothesis about how to weed out shallow knowledge was correct. It is in conflict with the mastery learning prescription to have students master easier problems before moving to more difficult ones. Experiment 2 was designed to test this hypothesis, namely to see whether early introduction of more difficult problems that challenge over-general knowledge hurts or helps learning.

6 Discussion and Future Work

Our SimStudent experiments targeted two instructional principles that appear to be in conflict with each other. The "Mastery Learning" principle suggests that presentation and mastery of simpler component problems should precede presentation of more complex problems. A second "Challenge Shallow Knowledge" principle suggests that certain harder problems should be introduced earlier in a curriculum to prevent students from over-practicing shallow strategies that only work on simpler problems. Together these principles result in an instructional dilemma. The Mastery Learning principle encourages the use of simple problems early in the curriculum while the Challenge Shallow Knowledge principle encourages the use of harder problems early in the curriculum.

The acquisition of shallow, over-general productions appears more likely for certain classes of problems than others. For those problem categories where shallow knowledge acquisition is less likely, the Mastery Learning principle should be heeded, that is, curricula should start with simpler problems. For problem categories where shallow production acquisition is likely, we suspect the Challenge Shallow Knowledge principle is more important and harder problems should be introduced early. Results of experiment 2 are consistent with this claim, but we need to do further experiments and sensitivity analysis.

A broader issue is how faithfully our simulated student experiments have captured the notion of learning. For example, we do not model the knowledge-level learning process (i.e., the acquisition of productions via ACT-R's analogy mechanism). We could add this functionality, but analogy produces a number of productions at different levels of generality, which then need to be pruned by experience. We feel this pruning process is more central than the knowledge level learning, and hypothesize the results presented here would not be affected if we enabled EAPS to acquire productions via analogy.

There are also three main issues that need to be considered with respect to Mastery Learning. The first issue that needs to be considered is the nature of the feedback students receive. In our experiments the SimStudents were simply told they were right or wrong. Clearly giving the model more directed feedback, or giving the model the ability to provide some of this corrective feedback, would be a more accurate model of a good instructional environment. This kind of more directed feedback is not possible with hidden skills, so it is possible to imagine a combination of directed and blanket feedback being used in future work with simulated students.

A second issue that needs to be addressed is the corrective instruction students receive if they have not mastered the lesson they are currently working on. According to Bloom (1987) if students are having trouble with a given lesson the teacher should provide corrective instruction using a different approach. However, we are currently only modeling learning as an increase in the utility estimate of the knowledge, and the student just applies the same approach again, so different ways of assisting EAPS needs to be looked into.

We have considered some ways in which the credit and blame attribution might be improved in our SimStudents learning processes including: 1) a better blame attribution algorithm, which assigns blame in proportion to current strength of productions involved, and 2) more deliberate reasoning in blame attribution whereby the student reasons about which subgoal is at fault. Beyond the challenge of implementing such improvements, it is an interesting question whether human learner behavior is consistent with such improvements or is more like ACT-R's current model of blame attribution.

It is interesting to note in retrospect that the two curricula in SimStudent1 did not contain problems that frequently resulted in an error when the Operate-on-Adjacent-Numbers production fired, which means that the SimStudents that pass those curricula may still make Order-of-Operations errors. However it is also probably the case that curricula being used in schools today do the same thing, leaving large amounts of shallow knowledge present in *real* students!

References

Anderson, J. R., & Lebiere, C. (1998). The Atomic Components of Thought. Laurence Erlbaum.

Bloom, B. (1987). A Response to Slavin's Mastery Learning Reconsidered, 57, 507-508.

Corbett, A.T. and Anderson, J. R. (1995). Knowledge decomposition and subgoal reification in the ACT programming tutor. Artificial Intelligence and Education, 1995: The Proceedings of AI-ED 95. Charlottesville, VA: AACE.

Koedinger, K.R., & MacLaren, B. A. (2002). Developing a Pedagogical Domain Theory of Early Algebra Problem Solving. CMU-HCII Tech Report 02-100. Accessible via http://reports-archive.adm.cs.cmu.edu/hcii.html

Koedinger, K. R. & Nathan, M. J. (Under review). The real story behind story problems: Effects of representations on quantitative reasoning.

Kulik, C. C., Kulik, J. A., & Bangert-Drowns, R. L. (1990). Effectiveness of mastery learning programs: A meta-analysis. Review of Education Research, 60, 265-299.

Lovett, M. (1998). Choice. In The Atomic Components of Thought. Anderson J.R. & Lebiere, C. (pp255-296) Laurence Erlbaum Associates .

Nathan, M. J. , Long, S. D. & Alibali, M. W. (in press). An analysis of the presentation of arithmetic and algebraic topics in common mathematics textbooks. In preparation.

Slavin, R. (1987). Mastery learning reconsidered. Review of educational research, 57, 175-213.

VanLehn, K., Ohlsson, S., & Nason, R. (1994). Applications of simulated students: An exploration. Journal of Artificial Intelligence and Education, 5(2), 135-175.